



VRIJE UNIVERSITEIT BRUSSEL

FACULTEIT INGENIEURSWETENSCHAPPEN
VAKGROEP TOEGEPASTE MECHANICA

TOWARDS SAFE CONTROL OF A COMPLIANT
MANIPULATOR POWERED BY PNEUMATIC
MUSCLES

Michaël Van Damme

May 2009

Jury:

Prof. dr. ir. F. Daerden (VUB-MECH), advisor
Prof. dr. ir. D. Lefeber (VUB-MECH), co-advisor
Prof. dr. ir. J. Tiberghien (VUB-ETRO), chairman
Prof. dr. ir. R. Pintelon (VUB-ELEC), vice-chairman
Prof. dr. ir. B. Vanderborght (VUB-MECH), secretary of the chairman
Prof. dr. ir. Ph. Lataire (VUB-ETEC)
Prof. dr. ing. A. Bicchi (University of Pisa)
Dr.-Ing. A. Albu-Schäffer (German Aerospace Center, München)

Print: DCL Print & Sign, Zelzate

© 2009 Michaël Van Damme

Uitgeverij VUBPRESS Brussels University Press
VUBPRESS is an imprint of ASP nv (Academic and Scientific Publishers nv)
Ravensteingalerij 28
B-1000 Brussels
Tel. ++32 (0)2 289 26 50
Fax ++32 (0)2 289 26 59
E-mail: info@vubpress.be
www.vubpress.be

ISBN 978 90 5487 585 7
NUR 959
Legal Deposit D/2009/11.161/044

All rights reserved. No parts of this book may be reproduced or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the author.

To Loyda

Acknowledgements

The work that resulted in this dissertation has only been possible thanks to the direct and indirect support of many people.

I am particularly grateful to my advisor, Frank Daerden, for his never-ending support and his advice, for all the freedom he gave me in my work, and for always being there when I needed it. I would also like to warmly thank my co-advisor, Dirk Lefeber, for his support, for many interesting suggestions and discussions, and for the flexibility he allowed me in conducting my work. Without the guidance and confidence of my advisors, the realization of this thesis would not have been possible.

I would also like to thank all members of the thesis committee for their time, effort and their interest in my work.

I'm especially indebted to Bram Vanderborcht, for his help in countless ways, for his many suggestions, his relentless effort and for keeping me "on track" from time to time. Without him, the thesis might never have been finished in time.

Thank you also to Ronald Van Ham for all his practical advice and support (both research and non-research related), thanks to Joris Naudet for many interesting discussions and to Pieter Beyl for his suggestions and for proofreading a part of the manuscript. Thanks to Innes Vanderniepen, Rino Versluys, Björn Verrelst and the other members of the Robotics & Multibody Mechanics research group. I also want to thank Patrick Guillaume and Svend Bram for their help and suggestions, Gert De Sitter for his support, Thierry Lenoir for his help with the computer network, Jean-Paul Schepens, André Plasschaert and the other guys from the workshop for their technical assistance, Jenny D'haes for the administrative support and all master's students who have contributed to this work.

Of course, I'm very grateful to both of my parents. A warm thank you to my father, for convincing me to study engineering, and for always believing in me, and to my mother, for her openness and for always supporting and stimulating me. And of course, thanks to Eliane and Karel, too.

A big thank you to my brother Rafael, who always encouraged me.

I'd also like to thank my friends for being who they are, and for all their direct and indirect support during the last years. I won't list all your names here, but you know who you are.

And last but not least, a very warm thank you to Loyda, for always being there for me, for taking very good care of me and for making me happy.

Michaël Van Damme
Brussels, May 2009

Contents

Acknowledgements	v
1 Introduction	1
1.1 Motivation	1
1.2 Compliance	2
1.3 Compliant actuators	2
1.4 Goal	3
1.5 Overview	4
I Design and Modeling	7
2 Design and Instrumentation	9
2.1 Introduction	9
2.2 Pleated pneumatic artificial muscle	9
2.2.1 Introduction	9
2.2.2 Force characteristic	10
2.2.3 Volume characteristic	14
2.2.4 Antagonistic setup	14
2.3 Design	16
2.3.1 Introduction	16
2.3.1.1 Degrees of freedom	16
2.3.1.2 Configuration	16
2.3.1.3 Muscle placement	17

2.3.1.4	Muscle slenderness	17
2.3.1.5	Link length	18
2.3.1.6	Workspace	18
2.3.2	Torque calculation	18
2.3.2.1	Torque functions	20
2.3.2.2	Global contraction	20
2.3.2.3	Series arrangement of PPAMs	21
2.3.2.4	Connection length	22
2.3.2.5	Torques	23
2.3.2.6	Compliance	24
2.3.3	Design	24
2.3.3.1	Muscle parameters	25
2.3.3.2	Attachment point locations	25
2.3.3.3	Torque characteristics	26
2.3.3.4	Mechanical design	27
2.3.3.5	Maximum load	27
2.4	Instrumentation	31
2.4.1	Control system	31
2.4.2	Pressure sensors	31
2.4.3	Valves	32
2.4.4	Force sensors	32
2.4.5	Encoders	32
2.4.5.1	Angle measurement	32
2.4.5.2	Angular velocity measurement	33
2.4.5.3	Inter-pulse time measurement board	36
2.4.6	Glue electronics	38
2.5	Summary	38
3	Modeling	39
3.1	Introduction	39
3.2	Mechanical model	39
3.3	Pressure dynamics	40
3.3.1	Δp - approach	41
3.4	Muscle dynamics	42
3.5	Parameter estimation	42

3.5.1	Static model	42
3.5.1.1	Pneumatic muscles	43
3.5.1.2	Valves	44
3.5.1.3	Mechanical model	45
3.5.1.4	Experiment	47
3.5.1.5	Gravity compensation	47
3.5.1.6	Validation	48
3.5.2	Dynamical case	49
3.5.2.1	Introduction	50
3.5.2.2	Accelerations	51
3.5.2.3	Friction	53
3.5.2.4	Excitation	54
3.5.2.5	Noise	56
3.5.2.6	Parameter estimates	57
3.5.2.7	Validation	58
3.6	Hysteresis in PPAMs	62
3.6.1	Experiments	62
3.6.2	Modeling	65
3.6.3	The Preisach model	66
3.6.4	Application to the PPAM	69
3.6.5	Results	71
3.6.6	Conclusion	71
3.7	Summary	73
II Control		75
Introduction to Part II		77
	Previous Work	77
	Challenges	79
4	Sliding Mode Control	81
4.1	Introduction	81
4.2	Sliding mode control	82
4.2.1	Ideal sliding mode control	82
4.2.2	Safety	84

4.2.3	Robustness	85
4.2.4	Chattering	87
4.2.5	Actuator dynamics	89
4.2.6	Feedback linearization	89
4.2.7	Internal dynamics	93
4.3	Sliding mode control of the manipulator	95
4.3.1	Introduction	95
4.3.2	Complete system model	96
4.3.3	Feedback linearization	97
4.3.4	Controller	99
4.4	Results	100
4.4.1	Simulation	100
4.4.2	Experiments	101
4.5	Conclusion	102
5	Proxy-Based Sliding Mode Control	105
5.1	Introduction	105
5.2	Proxy-based sliding mode control	106
5.2.1	Introduction	106
5.2.2	Replacing the signum function	106
5.2.3	Proxy	107
5.2.4	Discrete-time controller	109
5.2.5	Continuous-time controller	111
5.2.6	Discussion	113
5.2.6.1	Sliding surface	113
5.2.6.2	Compliance	114
5.2.6.3	Tracking	116
5.2.6.4	Relation with traditional controllers	117
5.2.6.5	Summary	119
5.3	Controllers	119
5.3.1	PID control	119
5.3.2	Proxy-based sliding mode control – Task space implementation	120
5.3.3	Proxy-based sliding mode control – Joint space implementation	121
5.3.4	Experimental results	125
5.3.5	Discussion	133

5.4	Interaction	134
5.4.1	Admittance control	135
5.4.2	Force estimation	136
5.4.2.1	Recursive least-squares	136
5.4.2.2	Generalized momentum based observer	138
5.4.2.3	Comparison	141
5.4.3	Interaction experiment	142
5.5	Conclusion	144
6	Safety	145
6.1	Introduction	145
6.2	Robot safety	146
6.2.1	Quantifying safety	146
6.2.2	Design and control for robot safety	147
6.3	Safety of the pneumatic manipulator	149
6.3.1	Contact model	149
6.3.2	Simulation	151
6.3.3	Results and discussion	152
6.3.4	Influence of joint stiffness	154
6.3.5	Limitations	156
6.4	Conclusion	157
7	Conclusion	159
7.1	Overview	159
7.2	General conclusions	161
7.3	Future work	161
A	Torque Calculations	163
A.1	Muscle torques	164
A.2	Torque functions	164
A.3	Contractions	166
A.4	Maximum load	169
B	Dynamic Model	171
B.1	Equations of motion	171
B.2	Adapted torque functions	173

B.3	Equations of motion linear in the parameters	174
B.4	Filtered equations of motion linear in the parameters	175
B.4.1	Friction	176
B.5	Filtered observation matrix	177
B.6	Results of the static estimation	178
C	Feedback Linearization	179
C.1	Matrices	179
C.2	Coordinate transformation	181
D	Proxy-Based Sliding Mode Control	189
D.1	Signum function and unit saturation function	189
D.2	Massless proxy	190
D.3	Discrete-time controller equations	192
D.4	The proxy in state-space	195
E	Publication list	199
	Bibliography	205

Chapter 1

Introduction

1.1 Motivation

Exactly fifty years ago, in 1959, a prototype of Unimation Inc.'s Unimate robot was installed in a factory of General Motors. It started the era of industrial robots. Today, over one million industrial robots are operational worldwide, in many different industries.

Historically, one of the prime reasons for the introduction of robots in industrial applications was to remove human operators from potentially hazardous work environments. Paradoxically, the robots themselves also pose a threat to workers. The design and control of industrial robots are optimized for performance, which provides them with a high speed of execution, high accuracy and high repeatability. Their high weight, high speeds, stiff characteristics and high gain control make them dangerous if a collision with a human should occur. For this reason, people are not allowed in the vicinity of a robot while it is working.

Over the last few years, the evolution in new and envisaged robotic applications requires increasingly closer contact between humans and robots. The growing interest in robots that operate in the close vicinity of people, or even physically interact with them, is witnessed by the number of European projects active in the field (e.g. PHRIENDS and URUS in FP6, and VIATORS, ROBOT@CWE, CHRIS and DEXMART in FP7). This trend is even stronger in Japan. The spectrum of possible applications includes rehabilitation robots (to help people re-learn motor skills they've lost in an accident or due to stroke, for instance), robotic prostheses, robot assistants for helping the elderly, manufacturing (with human-robot collaboration), entertainment robots, wearable robots, etc.

The requirements for the new generation of robots are fundamentally different. Unlike industrial robots, they will function in unstructured environments, and have only partial knowledge of their surroundings. Since contact between the robot and

objects or people surrounding it is possible, safety is the most important requirement.

Making robots sufficiently safe and dependable to be suitable for physical human-robot interaction is a challenge. It requires the combination of lightweight materials, new actuators, soft-robotics features and adapted control strategies.

1.2 Compliance

Most robots available today use non-backdriveable actuators such as geared electric motors (with high gear ratio) and hydraulic actuators. This causes the robot to appear stiff if it comes into contact with the environment or a person. High stiffness can lead to high contact forces, especially in combination with high gain control. This is undesirable and possibly dangerous for robots that operate in close contact with humans. Human friendly robots have to be “soft”, they need to be compliant.

Compliance can be obtained by means of design, which is called passive compliance, or by means of control, which is called active compliance.

In active compliance, an inherently stiff robot is made to appear compliant in case of contact with the environment. This can be achieved with a form of impedance control (Hogan, 1985), for instance. The drawback of active compliance is that its response is limited by the actuator, sensor and control bandwidth. Thus, it is not guaranteed that it will be able to respond fast enough in case of a collision, which is not optimal from the viewpoint of safety.

By introducing compliant (or flexible) elements in the robot’s structure, we obtain passive compliance. Passive compliance, or “hardware” compliance, is inherent to the mechanism, it doesn’t depend on control. This allows the robot to naturally handle impacts and disturbances, without the need for any control action. The disadvantage of passive compliance is that it makes the robot more difficult to control, which generally translates into lower speed and accuracy.

We see that safety and performance are competing features. When designing for performance, it is difficult to be safe (due to the limitations of active compliance), and when designing for safety (i.e. lightweight and with passive compliance) it is difficult to achieve high control performance. Finding the optimal balance between safety and performance remains an open challenge in robotics.

1.3 Compliant actuators

A common way to make a robot passively compliant is to use compliant actuators. The most well known compliant actuator is probably the series elastic actuators (SEA, Pratt and Williamson (1995)), which consists of a spring in series with a (geared) electric motor. Since the spring constant is fixed, the compliance of SEAs

cannot be changed during operation¹. For reasons of safety (Bicchi and Tonietti, 2004), energy efficiency (Vanderborght et al., 2006b, 2008b) or functionality (Van Ham et al., 2007), one often wishes the compliance to be adaptable. An overview of the different types of actuators with adaptable compliance is given in Van Ham et al. (2009)

Pneumatic artificial muscles (PAMs), which will be introduced more formally in section 2.2, are contractile devices operated by pressurized air. When inflated, they bulge, shorten and thereby generate a contraction force.

A PAM can be considered equivalent to a pneumatic cylinder with a varying piston area. The compressibility of air, combined with the change in effective piston area, make the PAM a compliant actuator. When used in an antagonistic setup, a pair of PAMs allows both position and compliance to be controlled. Since the PAM is also a very lightweight device, it is a good candidate to be used as a “soft” actuator in a human friendly robot. The pleated pneumatic artificial muscle (PPAM), a specific type of PAM developed by Daerden (1999) is the compliant actuator used in this work.

1.4 Goal

It is commonly accepted that “making a heavy, rigid robot behave gently and safely is an almost hopeless task, if realistic conditions are taken into account” (Bicchi et al., 2008). In other words, adding active compliance to a standard robot is not enough to obtain a safe system.

The situation is not as clear for robots that have been designed with safety in mind, however. Low weight and passive compliance are the most cited design factors that contribute to the safety of a robot, but it is not clear if they are sufficient. Lightweight, passively compliant robots could be safe given that their compliance is high enough, but it isn’t sure.

The goal of this dissertation is to make a contribution to closing the gap between safety and performance by developing a controller that combines good tracking performance with a high degree of safety for a passively compliant system. Safety is considered in the control design itself. Thus, it is not left to the hardware to (hopefully) take care of safety by itself, but the controller actively contributes to it.

Since control of highly compliant systems is challenging, it was important to verify that the approach actually works. For this reason, a lightweight two degree-of-freedom planar manipulator actuated by pleated pneumatic artificial muscles was built (see fig. 2.13 on page 28) to serve as a test platform. The main results of

¹Compliance of a series elastic actuator can be modified by means of control (see for instance Vallery et al. (2008)), but due to limited control bandwidth the real (i.e. passive) compliance will be felt in case of a sudden impact.

this work are not specific to pneumatic muscles, however, and can be generalized to other compliant actuators.

Having a safety-oriented controller gives us the possibility to compare performance with a standard controller, and to evaluate if the “safe” controller actually improves safety, and how much. In the process, we can also determine whether the hardware safety features are sufficient by themselves to guarantee an acceptable level of safety, or not.

1.5 Overview

The text is organised in two parts. Part I describes the design and modeling of the pneumatic manipulator:

Design and Instrumentation Chapter 2 gives an introduction to the pleated pneumatic artificial muscle (PPAM), describes its properties, and details the design of the 2-DOF manipulator actuated by PPAMs.

Modeling Chapter 3 describes how the mechanical system, the pneumatic servo valves and the muscles were modeled, and how the parameters in the models were determined. It also describes a hysteresis model for the PPAM.

Part II starts with an overview of previous work on control of pneumatic muscle systems. It continues with the description of the different controllers that were implemented, and includes a discussion about safety.

Sliding Mode Control Chapter 4 explains why sliding mode control was investigated, how it was implemented and how it performed.

Proxy-Based Sliding Mode Control In spite of its name proxy-based sliding mode control (PSMC) is quite different from ordinary sliding mode control. Chapter 5 explains PSMC and investigates its properties. Two versions of PSMC are implemented on the manipulator, and their performance is compared to that of PID (proportional-integral-derivative) control.

An interactive mode that uses admittance control and PSMC is also described. It is based on the estimation of external force acting on the end-effector. Two different methods to do this are explained.

Safety After a brief overview of safety in robotics, chapter 6 describes the impact simulations that were done to assess the safety aspects of the PSMC and PID controllers. It also investigates the influence of compliance on safety.

Conclusion Chapter 7 contains the conclusions of this work.

To avoid overloading the main text, some calculations (and in case of chapter 5 also discussions) were put in appendices. Chapters 2, 3, 4 and 5 all have an associated appendix, appendices A, B, C and D respectively.

An overview of the publications that have resulted from this work and from collaborating with colleagues on related subjects is given in appendix E.

Part I

Design and Modeling

Chapter 2

Design and Instrumentation

2.1 Introduction

Control algorithms and safety strategies for physical human-robot interaction cannot be developed by computer simulation alone, they have to be verified experimentally. In order to be able to do this, a small scale 2-DOF manipulator actuated by Pleated Pneumatic Artificial Muscles (PPAMs) was designed and built.

This chapter starts with an overview of the most important properties of the PPAM actuator, and continues to describe how the manipulator was designed. The design process will be seen to be heavily influenced by the fact that the muscle's output force changes as it contracts. Along the way, concepts that will be used in later chapters, such as torque functions, are introduced.

The last part of the chapter discusses the valves, sensors and electronics that were used, with an emphasis on how angular velocity is measured.

2.2 Pleated pneumatic artificial muscle

2.2.1 Introduction

A pneumatic artificial muscle (PAM), also called a fluidic muscle, an air muscle or pneumatic muscle actuator, is a contractile linear actuator operated by gas pressure. Its core element is a reinforced closed membrane that expands radially and contracts axially when inflated with pressurized air. Hereby the muscle generates a uni-directional pulling force along the longitudinal axis.

Over the years, different types have been developed. Daerden and Lefebber (2002) classified the pneumatic muscles under Braided muscles (this category contains the McKibben muscle and the Sleeved Bladder Muscle), Pleated PAMs, Netted Muscles (Yarlott Muscle, RObotic Muscle Actuator, Kukolj Muscle) and Embedded Muscles

(Morin Muscle, Baldwin Muscle, UnderPressure Artificial Muscle, Paynter Knitted Muscle, Paynter Hyperboloid Muscle, Kleinwachter torsion device).

The McKibben muscle (Schulte, 1961; Caldwell et al., 1995; Chou and Hannaford, 1996) is the most well known type, and it is commercially available from different companies (Shadow Robot Company, Merlin Systems Corporation, Hitachi Medical Corporation and Festo). It consists of a rubber tube, which expands when inflated, surrounded by a netting that transfers tension.

Although relatively easy to make, the McKibben muscle has some important drawbacks: moderate capacity of contraction (limited to 20% to 30% of its initial length), hysteresis as a result of friction between the outer sleeve and the rubber tube, and the presence of a threshold pressure, under which no contraction occurs.

Daerden (1999) developed a new PAM, the Pleated Pneumatic Artificial Muscle (PPAM, Daerden and Lefeber (2001, 2002)), to remedy some of these disadvantages. The PPAM has a folded membrane that unfolds as it expands. Because of the unfolding, there is virtually no threshold pressure and there is a strong reduction in energy losses in comparison to other muscle types. It can develop higher forces and it can reach higher levels of contraction (up to 40% of the muscle's maximum length). Verrelst et al. (2006a) have developed a second generation of the PPAM to extend the muscle lifespan and to simplify the construction process of the muscles.

Since it can contract more along its longitudinal axis than a McKibben muscle, the PPAM expands more radially as well, which is a disadvantage. This also makes it more difficult to group several PPAMs in a bundle, which is sometimes done with McKibben muscles (see for instance Ritter et al. (2005)), or to twist the PPAM around something. The main disadvantage of the PPAM is that it is not commercially available. It has to be produced manually and it is rather difficult to produce, which leads to variations in quality and durability. The PPAM's membrane also has to be shielded from sharp objects, as it is easily pierced.

Figure 2.1 shows three different states of contraction of a (second generation) PPAM. The membrane, shown in black on the picture, is made of a flexible, woven polyester fabric, made airtight by a polymer liner. Each pleat contains a strand of high modulus Kevlar® fibre (para-aramid synthetic fibre), shown in yellow in the photo. As the pressure inside the device is increased, the pleats in the membrane unfold and it expands radially. The Kevlar® fibre strands in the pleats, which are being pushed away from the muscle's longitudinal axis by the expanding membrane, translate this radial expansion into a longitudinal contraction.

2.2.2 Force characteristic

When neglecting the work needed to deform the membrane's material (very low for a PPAM due to its pleated structure) and the force needed to overcome the inertia of the muscle's moving parts (generally very low as well), the force generated by a



Figure 2.1: A second generation Pleated Pneumatic Artificial Muscle (PPAM) shown in three different values of contraction.

pneumatic artificial muscle can be written as (Daerden, 1999)

$$F = -p \frac{dV}{dl}. \quad (2.1)$$

In this expression, p is the gauge pressure inside the muscle, dV the infinitesimal change of the muscle volume (the volume enclosed by the membrane), and dl the change in actuator length (generally negative (indicating contraction) for a PPAM).

The volume of the actuator increases with decreasing length until a maximum volume is reached. At this point, which corresponds to maximum contraction, the force becomes zero. At low contraction the forces can be very high.

The fact that the force changes as a function of contraction is an essential difference between the pneumatic muscle and the pneumatic cylinder. The force generated by a pneumatic cylinder at gauge pressure p is proportional to the piston area inside the device. Since the piston area is constant, the force doesn't change with piston position. A pneumatic muscle can be considered as a pneumatic cylinder with varying piston area (equal to $-\frac{dV}{dl}$).

Based on the work of Daerden and Lefeber (2001), Verrelst et al. (2006a) have developed a detailed mathematical model for the PPAM. The model describes the shape of the muscle as a function of contraction, and it provides essential characteristics such as muscle force and enclosed volume. Static load tests have confirmed its validity.

Under the assumption of negligible elastic deformation the static force exerted by the PPAM muscle (i.e. eq. (2.1) written specifically for the PPAM) is given by the

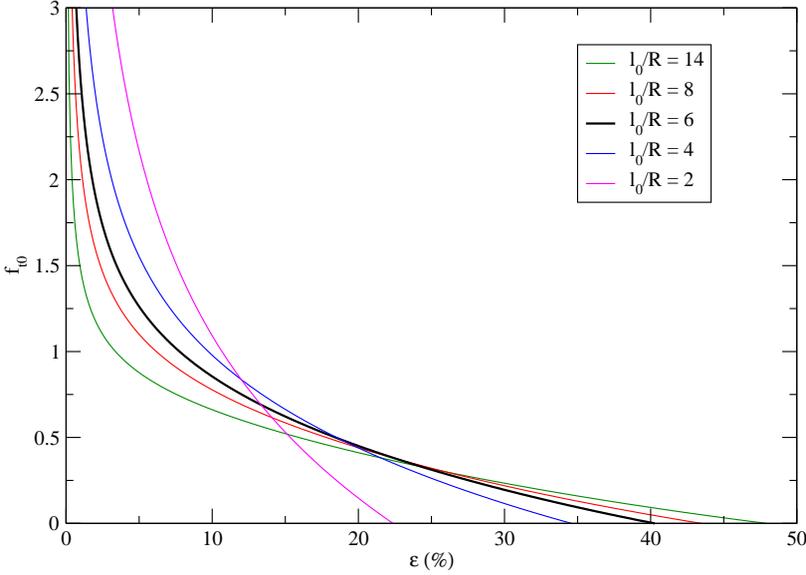


Figure 2.2: f_{t0} (dimensionless force function) as a function of contraction for different values of the slenderness l_0/R , and with $N = 25$ fibres.

model as

$$F = pl_0^2 f_{t0} \left(\epsilon, \frac{l_0}{R}, N \right), \quad (2.2)$$

In this expression, p is the applied gauge pressure, l_0 is the muscle's uncontracted length (or maximum length), R is its radius in uncontracted state (or minimum radius), N is the number of Kevlar[®] fibre strands in the membrane and ϵ is the muscle contraction. If we call l the muscle length, we have

$$\epsilon = \frac{l_0 - l}{l_0} = 1 - \frac{l}{l_0}.$$

f_{t0} is a nonlinear, dimensionless function that depends on contraction, the number of fibre strands and on the muscle's geometry as represented by the design-time parameter l_0/R (called the slenderness). f_{t0} is shown in figure 2.2 for different values of l_0/R . As expected, figure 2.2 and equation (2.2) show there is a varying force-displacement relation at constant gauge pressure, with high forces being generated at low contractions and very low forces at high contractions. This can be seen in figure 2.3 for a muscle with slenderness $l_0/R = 6$, $l_0 = 6$ cm and $N = 25$.

In the original model by Daerden and Lefebber (2001), the membrane was assumed to consist entirely of Kevlar[®], which corresponds to taking the number of fibres

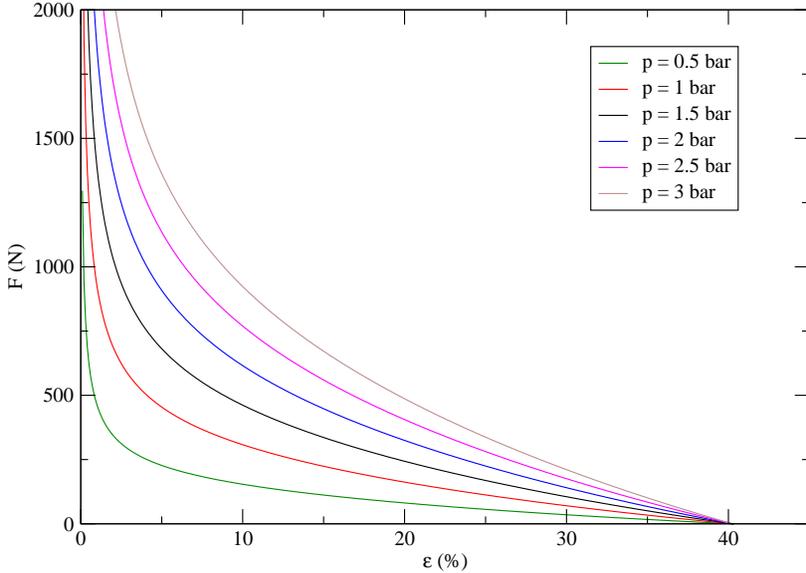


Figure 2.3: Force exerted by a PPAM with $l_0/R = 6$, $l_0 = 6$ cm and $N = 25$ for different gauge pressures (1 bar equals 100 kPa).

N equal to infinity in (2.2). The modification by Verrelst et al. (2006a) to include the effect of a limited number of fibres is only important if the number of fibres is very low. If N is higher than 15, the difference between both models is less than 3%, and even smaller for higher N . Because of this, the dependence of f_{t0} on N is usually not explicitly written, i.e. we write $f_{t0}(\epsilon, l_0/R)$.

In practice, the dimensionless function f_{t0} is difficult to work with since it is not available in analytical form. In order to evaluate it for given ϵ , l_0/R and N , a system of equations involving elliptic integrals has to be solved numerically (see Daerden and Lefebvre (2001); Verrelst et al. (2006a)). Another problem is that the radius R is usually not accurately known, and varies slightly between different muscles. For these reasons, the full mathematical model is usually only used during the design phase of a system that involves PPAMs. Once the slenderness is known, f_{t0} is approximated by a function of the following form (Verrelst et al., 2006a):

$$f_{t0}(\epsilon) \approx f_0\epsilon^{-1} + f_1 + f_2\epsilon + f_3\epsilon^2 + f_4\epsilon^3. \quad (2.3)$$

The coefficients $f_0 \dots f_4$ are determined by fitting (2.3) to the theoretical f_{t0} or to measured data.

Since low contractions correspond to very high forces (see fig. 2.3), contraction should be kept above a certain minimum (chosen to be 5% in this work) in order to avoid excessive material loading. For the same reason, and in order to ensure

sufficient lifespan for the actuators, gauge pressures have to be limited. Although in this work a limit of 3 bar (or 300 kPa) is chosen, the PPAM can generally withstand pressures of up to 4 bar (400 kPa), which is lower than the industrially used 6 to 10 bar.

2.2.3 Volume characteristic

Since the enclosed volume of a pneumatic muscle changes with contraction, the pressure regulating servo valves continuously have to adjust the airflow in or out of the muscle in order to keep the gauge pressure at the desired value. For modeling this interaction as well as to calculate compliance, it is necessary to know the volume of the muscle.

Assuming negligible membrane and fiber elasticity, the volume of a PPAM is given by

$$V = l_0^3 v_0 \left(\epsilon, \frac{l_0}{R} \right) \quad (2.4)$$

according to the theoretical model (Daerden and Lefeber, 2001; Verrelst et al., 2006a). $v_0(\epsilon, l_0/R)$ is a dimensionless function that isn't available in analytical form. Just as in the case of f_{t0} , it is computationally very expensive to evaluate, so once the slenderness l_0/R is known, it is approximated by a polynomial (Verrelst et al., 2006a):

$$v_0(\epsilon) \approx v_0 + v_1\epsilon + v_2\epsilon^2 + v_3\epsilon^3 + v_4\epsilon^4 + v_5\epsilon^5. \quad (2.5)$$

The coefficients $v_0 \dots v_5$ are determined by fitting (2.5) to the theoretical v_0 . The enclosed volume V as given by eq. (2.4) is shown in figure 2.4.

2.2.4 Antagonistic setup

Since PAMs are contractile devices, they can only exert force in one direction (they can only pull, not push). In order to have a bidirectionally actuated revolute joint, two PAMs have to be used in an antagonistic setup.

There are two simple ways to antagonistically couple pneumatic muscles to achieve a rotating joint: using a pulley mechanism (as shown in fig. 2.5a), or using a lever (fig. 2.5b).

In the pulley mechanism, the radius of the lever arm is constant, equal to the radius r of the pulley, so the torque generated by a muscle is proportional to its pulling force (see eq. (2.2)). As can be seen in figure 2.3, the force drops very rapidly with contraction. This means that the joint will only be able to exert useful torque in a limited angle range.

If the muscle is coupled to the link using a lever arm, the point where the muscle is attached to the link can be chosen in such a way that the effective lever arm becomes longer with increasing contraction. This causes the nonlinear force-contraction

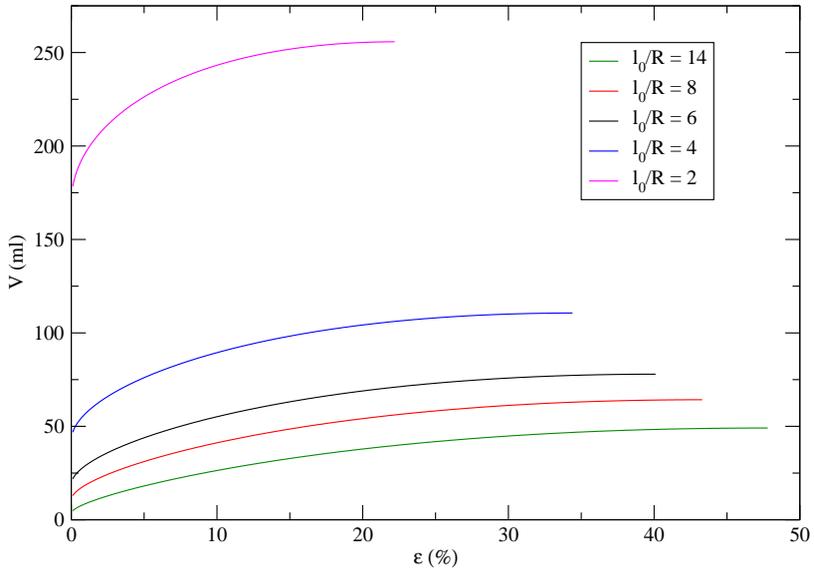


Figure 2.4: Volume enclosed by the PPAM membrane as a function of contraction for different values of the slenderness l_0/R .

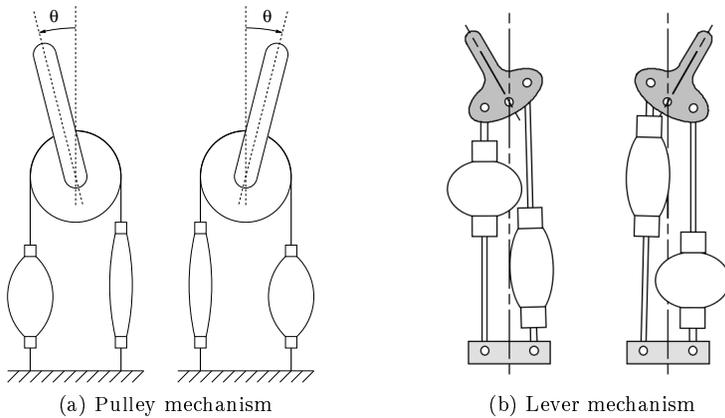


Figure 2.5: Antagonistic coupling of pneumatic muscles to achieve a rotating joint.

characteristic of the PPAM to be translated in a flatter torque-angle relation of the joint.

Since the nonlinear characteristic of the muscle can be somewhat smoothed when using lever arms, this option was selected.

2.3 Design

2.3.1 Introduction

Since it was chosen to use levers to realize the antagonistically actuated joint, we have to decide on where exactly to attach the PPAMs to the links. We also have to choose the geometry of the PPAMs we will use.

Before starting the actual design of the manipulator, initial decisions have to be made about the number of degrees of freedom, which configuration to use and how large the manipulator and its workspace are going to be.

2.3.1.1 Degrees of freedom

The difficulty of designing (and controlling) systems actuated by pleated pneumatic artificial muscles rises very rapidly as the number of degrees of freedom increase. It was decided to work with two degrees of freedom, which is enough to investigate control principles and physical human-robot interaction without adding unnecessary complexity.

2.3.1.2 Configuration

Two possible link configurations were considered for the manipulator, elbow-up and elbow-down. Representative shapes for their workspaces are shown in fig. 2.6 (the specifics depend of course on the length of the links and the angle range of the joints).

For testing safe interaction with people, both configurations would be ok. However, since the manipulator will not be equipped with a gripper or a tool, a load cannot be rigidly attached to it. Any load we attach will thus dangle below the end effector. In that case, it is unpractical if the load can collide with the second link, which may be possible in the elbow-down configuration. If the load is also being manipulated by a human operator, we don't want the system to obstruct his movements, which seems more likely in the elbow-down configuration as well. For these reasons, the elbow-up configuration was chosen.

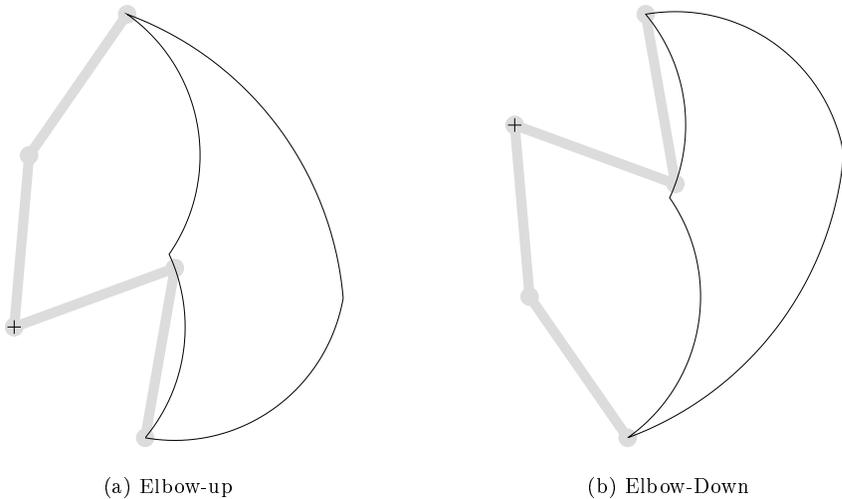


Figure 2.6: Considered configurations for the 2-DOF manipulator.

2.3.1.3 Muscle placement

In most systems actuated by muscles, both artificial and biological, the muscles are placed on the link before the one they actuate. This is the case in the human arm, for instance, where the muscles in the upper arm generate the torque that drives the lower arm. This is also the case in the antagonistic setups of fig. 2.5.

For the manipulator, it was decided not to follow this example, and to place the muscles that actuate a link on that link itself. This way, no muscles have to be placed on the base link, which results in a more compact system. A schematic representation of how the muscles are placed (including the muscle numbers as they are used in the rest of this text) can be seen in fig. 2.7.

A consequence is of course that the actuators themselves contribute to the inertia of the link they actuate. In this case, this effect is very moderate because of the low mass of the PPAMs (around 150 g).

2.3.1.4 Muscle slenderness

In order not to complicate the production of the muscles, it was decided that all muscles used in the manipulator would have the same slenderness (l_0/R) and minimum radius R .

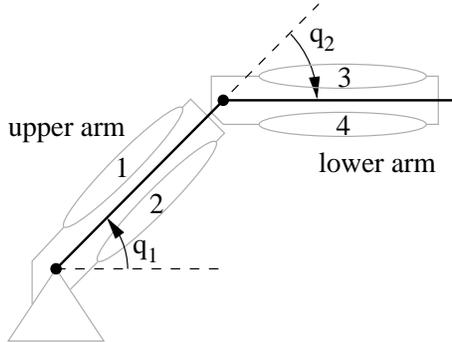


Figure 2.7: Muscle placement in the manipulator.

2.3.1.5 Link length

Since the manipulator was meant to be a lab-based research setup, we didn't want it to be too bulky. Thus the length of the links was chosen to be 30 cm, somewhat similar to the dimensions of the human arm. Both links were chosen to be of equal size in order to maximize the workspace.

The limited size also means that shorter, and thus weaker (see eq. (2.2)) muscles will be used, which increases safety during experimentation (in case of controller instability, for instance).

2.3.1.6 Workspace

The desired operating range of the joint angles was chosen to be

$$20^\circ \leq q_1 \leq 105^\circ \quad (2.6)$$

$$-130^\circ \leq q_2 \leq -30^\circ, \quad (2.7)$$

which avoids singularities in the workspace (the arm would be in a singular configuration for $q_2 = 0^\circ$ and $q_2 = 180^\circ$). Figure 2.7 shows how the joint angles q_1 and q_2 are defined. The resulting workspace is shown in figure 2.8.

2.3.2 Torque calculation

Before we are able to decide on the muscle attachment point locations, we have to calculate the effect they have on the torque output of each antagonistic pair of muscles. In order to do that, consider figure 2.9 and assume the attachment point locations to be known.

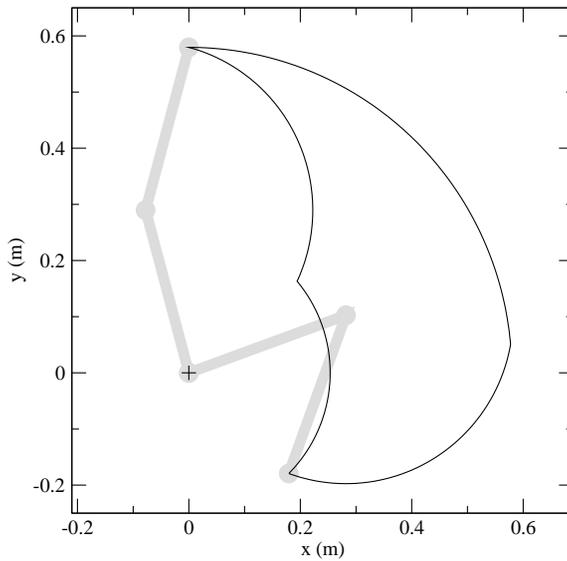


Figure 2.8: Workspace for the chosen link lengths and angle range.

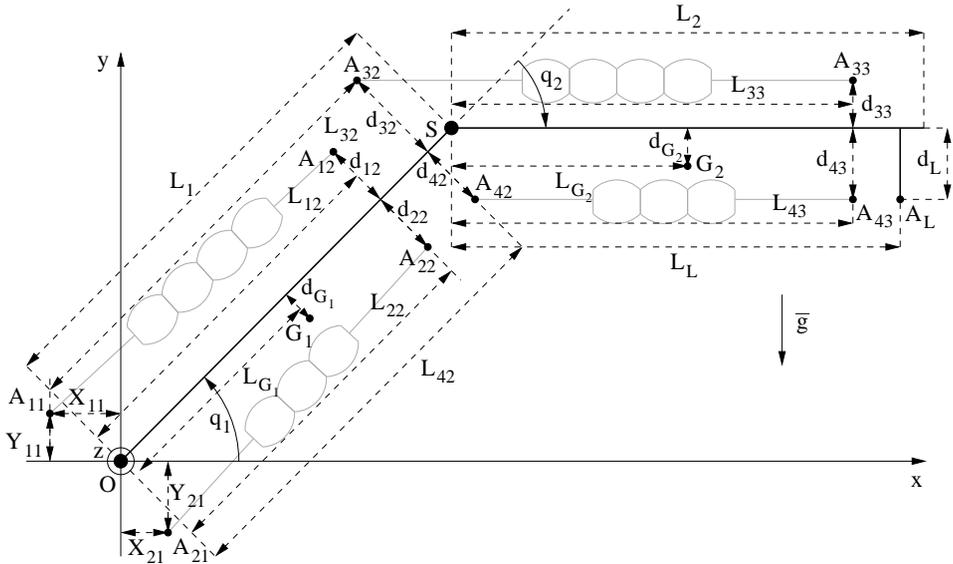


Figure 2.9: Schematic representation of the manipulator with all distances, angles and attachment points indicated. By convention q_1 is positive in this figure (anticlockwise), and q_2 is negative (clockwise).

2.3.2.1 Torque functions

Fig. 2.9 shows that muscle 1 (see fig. 2.7 for the muscle numbers) applies its force F_1 between points A_{11} and A_{12} (using the convention that A_{xy} indicates the point where muscle x is attached to link y , with the immobile base link having number 1, the upper arm having number 2 and the lower arm number 3). The moment of force this creates in the link's hinge point O is given by

$$\begin{aligned} \mathbf{M}_{O1} &= \mathbf{OA}_{12} \times (-F_1 \mathbf{e}_1) \\ &= F_1 (\mathbf{e}_1 \times \mathbf{OA}_{12}) \end{aligned}$$

with

$$\mathbf{e}_1 = \frac{\mathbf{OA}_{12} - \mathbf{OA}_{11}}{\|\mathbf{OA}_{12} - \mathbf{OA}_{11}\|},$$

where $\|\cdot\|$ represents the euclidean norm. \mathbf{M}_{O1} only has a z -component (since all forces and location vectors are in the $x - y$ plane), which we write as $\tau_{m,1}$:

$$\begin{aligned} \tau_{m,1} &= M_{O1,z} \\ &= \mathbf{M}_{O1} \cdot \mathbf{1}_z \\ &= F_1 (\mathbf{e}_1 \times \mathbf{OA}_{12}) \cdot \mathbf{1}_z. \end{aligned} \tag{2.8}$$

Using eq. (2.2) to replace F_1 this expression becomes

$$\begin{aligned} \tau_{m,1} &= p_1 l_0^2 f_{t0} \left(\epsilon_1(q_1), \frac{l_0}{R} \right) \cdot (\mathbf{e}_1 \times \mathbf{OA}_{12}) \cdot \mathbf{1}_z \\ &= p_1 m_{\tau 1} \left(\epsilon_1(q_1), \frac{l_0}{R} \right) \end{aligned} \tag{2.9}$$

with

$$m_{\tau 1} \left(\epsilon_1(q_1), \frac{l_0}{R} \right) = l_0^2 f_{t0} \left(\epsilon_1(q_1), \frac{l_0}{R} \right) \cdot (\mathbf{e}_1 \times \mathbf{OA}_{12}) \cdot \mathbf{1}_z \tag{2.10}$$

the so-called torque function of muscle 1. The torque functions of the other muscles can be calculated similarly (where the torque exerted by muscles 3 and 4 is calculated in point S (hinge point of the second link) instead of O).

2.3.2.2 Global contraction

Before we can evaluate (2.10) for given muscle parameters, we have to know $\epsilon_1(q_1)$, the contraction of muscle 1 as a function of the joint angle q_1 .

If we set

$$d_1(q_1) = \|\mathbf{OA}_{12} - \mathbf{OA}_{11}\| \tag{2.11}$$

the distance between muscle 1's attachment points, and

$$d_{1,max} = \arg \max_{q_{1,min} \leq q_1 \leq q_{1,max}} d_1(q_1)$$

$$d_{1,min} = \arg \min_{q_{1,min} \leq q_1 \leq q_{1,max}} d_1(q_1)$$

the maximum and minimum of that distance in the working area (i.e. with $q_{1,min} = 20^\circ$ and $q_{1,max} = 105^\circ$, see (2.6)) we see that muscle 1 has to be able to shorten by $d_{1,max} - d_{1,min}$.

Although at this stage we don't exactly know $d_{1,max} - d_{1,min}$ (since we haven't actually chosen the attachment point locations yet), we can assume (from the working area and link length) that it will be somewhere between 5 and 10 cm. Since the usable contraction of a PPAM is around 30% of its maximum membrane length l_0 (the force drops too low for higher contractions), the minimum muscle length needed to achieve 5 cm of contraction is $5/0.3 \approx 16.7$ cm. This is too long to be practical in this application since a muscle this size blows up to too large a diameter (same order of magnitude as its maximum length l_0) as it contracts.

2.3.2.3 Series arrangement of PPAMs

This can be solved by using a series arrangement of several identical short PPAMs (all used at the same gauge pressure). A series of n identical PPAMs exerts the same force as a single muscle, but the total shortening is n times larger. This allows for large contractions and relatively small diameters (when inflated) at the same time.

The disadvantage of this arrangement is that the maximum force that can be produced is reduced by a factor n^2 when compared to a single muscle n times as long. This follows from equation 2.2. However, since the forces developed by a PPAM can be very high, this needn't be a problem.

In order to calculate the contraction of an individual muscle in a series arrangement of n_m muscles, consider figure 2.10. If we group the length of the connecting rods and of the connections between the muscles under the name L_{conn} , if we call the total length d and the length of a single muscle membrane L_{membr} we have

$$d = L_{conn} + n_m L_{membr}$$

$$= L_{conn} + n_m (1 - \epsilon) l_0$$

or

$$\epsilon = \frac{L_{conn} + n_m l_0 - d}{n_m l_0} \quad (2.12)$$

Eq. (2.12) gives us the contraction of a single muscle in the series arrangement.

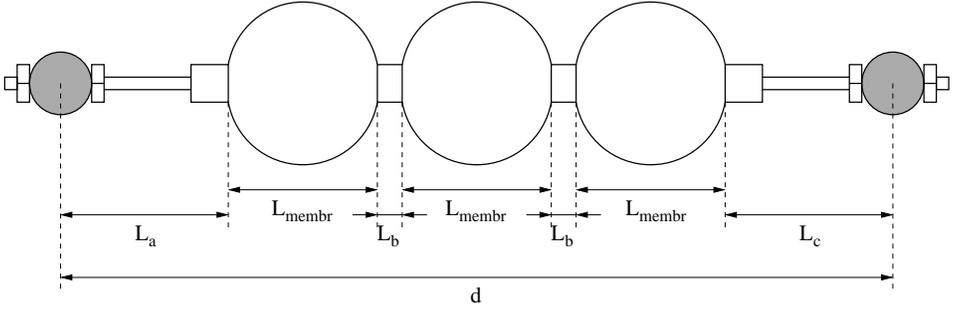


Figure 2.10: Series arrangement of PPAMs. The length of the connecting elements $L_{conn} = L_a + 2L_b + L_c$.

For muscle 1 of the manipulator, this becomes

$$\epsilon_1(q_1) = \frac{L_{conn,1} + n_{m,1}l_0 - d_1(q_1)}{n_{m,1}l_0}, \quad (2.13)$$

with $n_{m,1}$ the number of muscles in muscle group 1 (the series arrangement that we usually refer to as muscle 1), $L_{conn,1}$ the total connection length in the group, and $d_1(q_1)$ its total length as defined in 2.11.

2.3.2.4 Connection length

In order to determine $L_{conn,1}$, we return to a condition mentioned in section 2.2.2: the fact that muscle contraction has to remain above a certain minimum, $\epsilon_{1,min}$, to avoid excessive material loading. Of course, the minimum contraction occurs at maximum length, so (2.13) becomes

$$\epsilon_{1,min} = \frac{L_{conn,1} + n_{m,1}l_0 - d_{1,max}}{n_{m,1}l_0},$$

so we have

$$\begin{aligned} L_{conn,1} &= d_{1,max} + \epsilon_{1,min}n_{m,1}l_0 - n_{m,1}l_0 \\ &= d_{1,max} - n_{m,1}l_0(1 - \epsilon_{1,min}). \end{aligned} \quad (2.14)$$

The useful contraction range of a PPAM also has an upper bound: if the contraction becomes too high, the force drops so low that it is no longer useful (see fig. 2.3). We therefore also want the contraction to remain under a maximum value. This is used to determine the connection length of the antagonistic muscle group 2. We have

$$\epsilon_{2,max} = \frac{L_{conn,2} + n_{m,2}l_0 - d_{2,min}}{n_{m,2}l_0},$$

and thus

$$L_{conn,2} = d_{2,min} - n_{m,2}l_0(1 - \epsilon_{2,max}),$$

with

$$d_{2,min} = \arg \min_{q_{1,min} \leq q_1 \leq q_{1,max}} d_2(q_1)$$

and

$$d_2(q_1) = \|\mathbf{OA}_{22} - \mathbf{OA}_{21}\|.$$

The connection lengths for muscle groups 3 and 4 can be calculated similarly. For both muscle pairs, the same minimum and maximum values of contraction were used: $\epsilon_{min} = 5\%$ and $\epsilon_{max} = 30\%$.

2.3.2.5 Torques

By substituting (2.14) in (2.13), and the result in (2.10), we can now calculate the torque exerted by muscle 1 using eq. (2.9), for a given gauge pressure and a given set of design parameters.

Using similar calculations, we can do the same for the other muscles. We thus find the functions

$$\tau_{m,i} = p_i \cdot m_{\tau i}(\gamma) \quad (2.15)$$

with $\gamma = q_1$ for muscles 1 and 2 and $\gamma = q_2$ for muscles 3 and 4, i being the muscle's index (see fig. 2.7) and $m_{\tau i}$ the torque function associated with that muscle (the torque functions are calculated in more detail in appendix A.) p_i ($i = 1 \dots 4$) is the gauge pressure in muscle i . Equation (2.15) provides a clear separation between the two factors that determine torque: gauge pressure and a torque function $m_{\tau i}(\gamma)$, that depends on the design parameters (i.e. attachment point locations and muscle-related parameters) and on the joint angles.

The torque exerted by the actuators of the first link (muscles 1 and 2) is given by

$$\tau_1 = \tau_{m,1}(p_1, q_1) + \tau_{m,2}(p_2, q_1),$$

the torque exerted by the actuators of the second link (muscles 3 and 4) by

$$\tau_2 = \tau_{m,3}(p_3, q_2) + \tau_{m,4}(p_4, q_2).$$

The total actuator torque (in both joints) can thus be represented by

$$\boldsymbol{\tau} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} \tau_{m,1} + \tau_{m,2} \\ \tau_{m,3} + \tau_{m,4} \end{bmatrix} \quad (2.16)$$

$$= \begin{bmatrix} p_1 \cdot m_{\tau 1}(q_1) + p_2 \cdot m_{\tau 2}(q_1) \\ p_3 \cdot m_{\tau 3}(q_2) + p_4 \cdot m_{\tau 4}(q_2) \end{bmatrix}. \quad (2.17)$$

For a given set of gauge pressures and design parameters, this equation allows us to calculate the actuator torque in both joints.

Eq. (2.17) illustrates a very important property of the manipulator (and in general of devices powered by pneumatic muscles): the torque available through the actuators doesn't just depend on the input (gauge pressures), but also on the configuration (angles q_1 and q_2). Consequently, the maximum torque the actuators can deliver is different in each point of the workspace.

2.3.2.6 Compliance

Eq. (2.17) allows us to calculate the stiffness (and thus also its inverse, the compliance) of both joints. When considering the first joint, for example, we can write its stiffness as

$$\begin{aligned} K_1 &= -\frac{d\tau_1}{dq_1} \\ &= -\frac{dp_1}{dq_1}m_{\tau_1} - p_1\frac{dm_{\tau_1}}{dq_1} - \frac{dp_2}{dq_1}m_{\tau_2} - p_2\frac{dm_{\tau_2}}{dq_1}. \end{aligned} \quad (2.18)$$

The terms $\frac{dp_i}{dq_1}$ have to be considered, since a change in joint angle changes a muscle's contraction, and thus its volume, which affects muscle pressure. An expression for $\frac{dp_i}{dq_1}$ will be calculated in section 3.3.

2.3.3 Design

Since we have four PPAM groups, there are eight attachment points. The location of each of these points can be described by two coordinates.

Each muscle group has three parameters: slenderness l_0/R and maximum length l_0 of the individual muscles, and the number of muscles n_m in the series arrangement. This means there are a total of 28 parameters to be determined. The chosen parameter set has to meet two important conditions:

- producibility: not all imaginable muscles are producible (in general, the higher the slenderness, the more difficult to produce). In addition, attachment point locations cannot be chosen too close to each other, nor too far away from the link axis.
- absence of "space conflicts", which is the most difficult condition to verify. As PPAMs are inflated, they expand. At maximum contraction, a PPAM's diameter is close to its maximum length. Obviously, the muscle needs space to be able to expand.

The transmission rods that transfer the exerted muscle force to the structure can also cause problems. It must be made sure that the rods stay clear of all other structural elements throughout the entire operating area.

Additionally, we would like the muscles to have a high slenderness. As fig.2.2 shows, having a high slenderness expands the useful contraction range of the PPAM, since it increases the maximum contraction and lowers the forces at low contraction, so the muscle can be used at lower contractions without generating excessive force. As mentioned above, however, muscles with high slenderness are more difficult to produce.

Determining the best design means finding a global optimum in a 28-dimensional parameter space, subject to the above described conditions (some of which have to be verified throughout the entire working area). This has proven to be computationally intractable. Therefore, the different parameters were chosen manually after extensive computer experiments. The two main criteria taken into account when selecting the final parameter set were:

- ease of production (both for the muscles and the mechanical structure).
- avoiding areas in the workspace where the load carrying capability becomes too low.

2.3.3.1 Muscle parameters

In order to keep the muscles used in the manipulator easy to produce, a slenderness value of 6 was chosen for all muscles (i.e. $l_0/R = 6$). To keep the muscles compact when fully inflated, it was necessary to use relatively short muscles: a maximum membrane length l_0 of 6 cm was chosen. This has an important influence on the force output, since muscle force increase quadratically with l_0 (see eq. (2.2)).

Since the top muscle groups (1 and 3) carry both the manipulator and the load, they need the highest force output. For this reason, the number of individual muscles n_m in these muscle groups was chosen to be as high as the space available permits: $n_m = 4$. The higher n_m the more the contraction levels of the individual muscles are reduced. This keeps their force output high (see fig. 2.3).

In most of the workspace, the bottom muscle groups (2 and 4) don't contribute to supporting the weight of the load or the manipulator, only to the manipulator's stiffness. They thus don't have to be able to exert as much force as the top muscles, which means we can tolerate higher contractions. Therefore, these muscles were realized using three PPAMs in series.

2.3.3.2 Attachment point locations

The chosen attachment point coordinates are summarized in table 2.1 . Please refer to fig. 2.9 to see how the different distances are defined.

Coordinate	Value (mm)
X_{11}	39.0
Y_{11}	19.3
X_{21}	26.7
Y_{21}	24.1
L_{12}	315.0
d_{12}	30.0
L_{22}	217.5
d_{22}	30.0
L_{32}	334.8
d_{32}	19.0
L_{42}	270.0
d_{42}	0.0
L_{33}	285.0
d_{33}	30.0
L_{43}	285.0
d_{43}	30.0

Table 2.1: Attachment point locations, as indicated on figure 2.9.

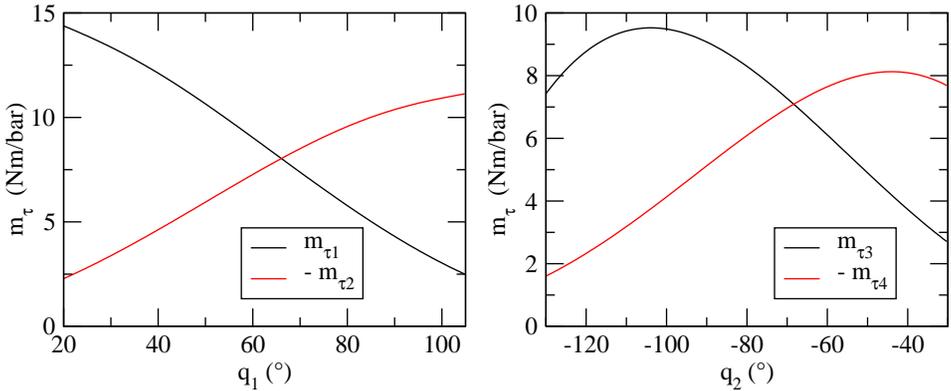


Figure 2.11: Torque functions $m_{\tau i}$, expressed in Nm/bar or $Nm/100kPa$.

2.3.3.3 Torque characteristics

With all design parameters known, we can now evaluate the torque functions $m_{\tau i}$ (with i the muscle number, see (2.10) for the definition of $m_{\tau 1}$). They are shown in figure 2.11.

These torque functions, multiplied by the applied gauge pressures, fully determine the torque delivered by the muscles, as seen in eq. (2.17).

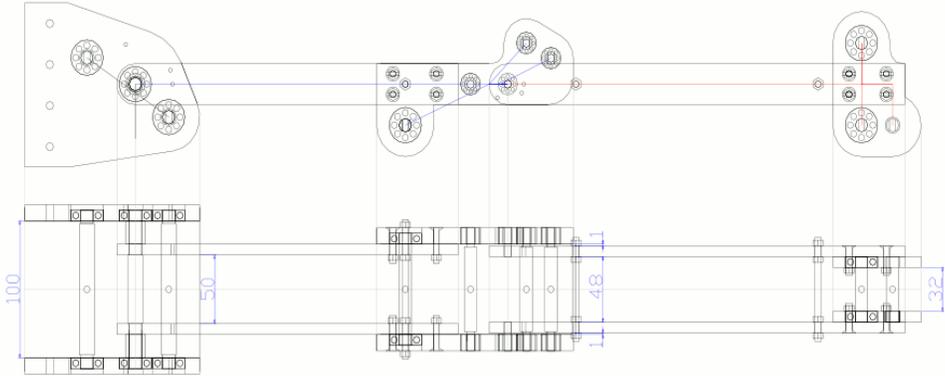


Figure 2.12: Actual design of the manipulator (dimensions in millimeter).

Coordinate	Value (mm)
L_L	300.0
d_L	30.0
L_{G_1}	≈ 192
d_{G_1}	≈ 0
L_{G_2}	≈ 196
d_{G_2}	≈ 2
L_1	300
L_2	310

Table 2.2: Dimensions as indicated on figure 2.9.

2.3.3.4 Mechanical design

With all attachment point locations known, the “skeleton” shown in fig. 2.9 still has to be converted to a real design. The result is shown in fig. 2.12, and a picture can be seen in fig. 2.13. In order to be able to make changes in the future the design was not optimized for weight.

The value for the coordinates of point A_L , where a load can be attached to the manipulator, and for the other dimensions shown on figure 2.9 are listed in table 2.2. Note that the coordinates of the centers of gravity of both links as listed in the table are based only on the CAD models, so they are not very accurate.

2.3.3.5 Maximum load

We will now calculate the maximum load the manipulator can carry throughout its workspace.

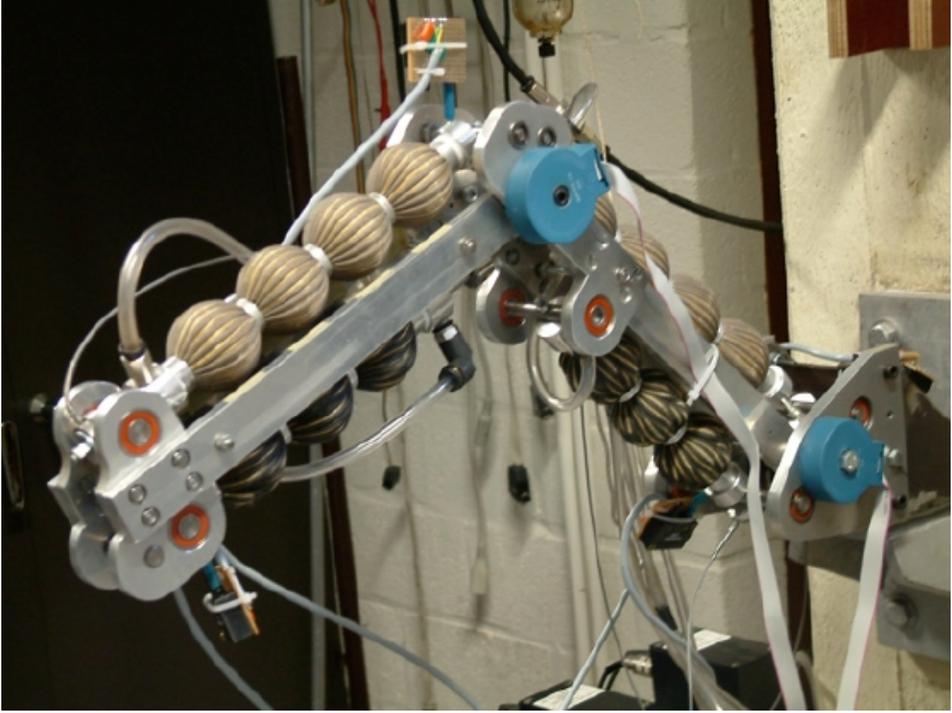


Figure 2.13: The manipulator.

The moment of force in O due to gravity is given by

$$\mathbf{M}_{Og} = \mathbf{OG}_1 \times (-m_1g\mathbf{1}_y) + \mathbf{OG}_2 \times (-m_2g\mathbf{1}_y) + \mathbf{OA}_L \times (-Mg\mathbf{1}_y),$$

and the one in S by

$$\mathbf{M}_{Sg} = \mathbf{SG}_2 \times (-m_2g\mathbf{1}_y) + \mathbf{SA}_L \times (-Mg\mathbf{1}_y).$$

In these equations, m_1 represents the mass of the first link, m_2 the mass of the second link and M the mass of a load that is assumed to be attached to the manipulator in point A_L . Estimated values for m_1 and m_2 are available from the CAD model.

Both moments only have a z -component, which we call τ_{Og} and τ_{Sg} respectively. We can then write:

$$\begin{aligned} \boldsymbol{\tau}_G &= \begin{bmatrix} \tau_{Og} \\ \tau_{Sg} \end{bmatrix} & (2.19) \\ &= \begin{bmatrix} (\mathbf{OG}_1 \times (-m_1g\mathbf{1}_y) + \mathbf{OG}_2 \times (-m_2g\mathbf{1}_y) + \mathbf{OA}_L \times (-Mg\mathbf{1}_y)) \cdot \mathbf{1}_z \\ (\mathbf{SG}_2 \times (-m_2g\mathbf{1}_y) + \mathbf{SA}_L \times (-Mg\mathbf{1}_y)) \cdot \mathbf{1}_z \end{bmatrix} \end{aligned}$$

If we assume the system to be in static equilibrium in a certain configuration, we must have (using eq. (2.16))

$$\tau + \tau_G = 0$$

or

$$p_1 \cdot m_{\tau_1}(q_1) + p_2 \cdot m_{\tau_2}(q_1) = -\tau_{Og} \quad (2.20)$$

$$p_3 \cdot m_{\tau_3}(q_2) + p_4 \cdot m_{\tau_4}(q_2) = -\tau_{Sg} \quad (2.21)$$

By setting the gauge pressure in the carrying muscles to the maximum pressure p_{max} (which we take to be 3 bar or 300 kPA), and in the antagonists to zero, we can calculate the maximum load M_{max} the manipulator can carry from both eq. (2.20) and eq. (2.21). Assuming muscle 3 is a carrying muscle, for instance, eq. (2.21) and (2.19) give

$$M_{max,2} = \frac{p_{max} \cdot m_{\tau_3}(q_2) + (\mathbf{SG}_2 \times (-m_2 g \mathbf{1}_y)) \cdot \mathbf{1}_z}{(\mathbf{SA}_L \times g \mathbf{1}_y) \cdot \mathbf{1}_z} \quad (2.22)$$

(see also appendix A).

Before we can calculate the actual maximum load, however, we have to determine which muscles are carrying the weight and which muscles are not. For the first joint, there is no problem: the global center of gravity is always to the right of the hinge point O , so the top muscle (muscle 1) will always be the carrying muscle. For the second joint, the situation is different: in the lower left region of the working area, it is possible for the joint center of gravity of the second link and the load to be to the left of hinge point S . In that case, muscle 4 becomes the carrying muscle. The easiest way to determine which muscle is the carrying muscle, is to assume it is muscle 3 and calculate the maximum load M from eq. (2.22). If it is negative, muscle 4 is the carrying muscle (a negative result indicates that muscle 3's torque in S and the gravitational torque of link 2 and the load in S work in the same direction, so static equilibrium can only be achieved by reversing the direction of gravity). The region of the workspace where muscle 4 carries is shown in grey in figure 2.14.

Once we know the carrying muscles for a certain configuration (a point in the workspace), we can isolate M from eqs. (2.20) and (2.21). Both equations will yield a certain value for the maximum load (which we call $M_{max,1}$ and $M_{max,2}$), the minimum of which is of course the true maximum.

The maximum load was calculated for a closely spaced grid of points in the working area, the results are shown in fig. 2.15. Some contour lines in fig. 2.15 have sharp angles. The reason for this is the fact that the figure shows $M_{max} = \min(M_{max,1}, M_{max,2})$. The sharp angles are caused by a switch between the maximum load according to (2.20) and the one according to (2.21). The combination of high gravitational torque and high muscle contraction (which means low force output) is the cause for the regions with low maximum load.

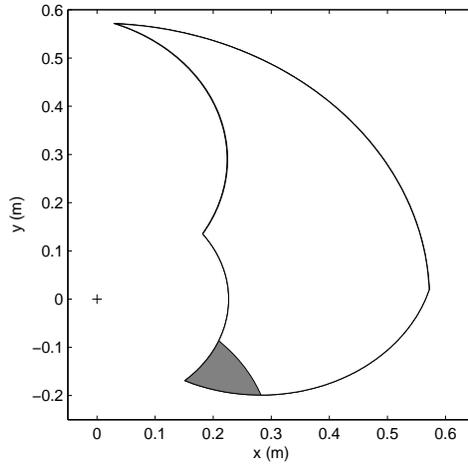


Figure 2.14: Region of the workspace where muscle 4 is the carrying muscle (shown in grey).

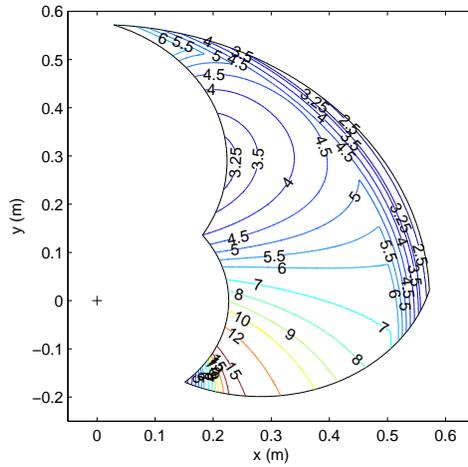


Figure 2.15: Contour plot showing the maximum load (in kg) that the manipulator can carry throughout its workspace.

The lowest value in the graph is just over 2 kg. This is the true maximum load of the system, since it can be supported anywhere in the working area. Of course, if one stays clear of the edges of the workspace the maximum load becomes much higher.

The maximum load value of 2 kg should be compared to the manipulator's own mass, however, which stands at around 2.5 kg, so the the system has a load to mass ratio of around $2/2.5$ or about 80%.

2.4 Instrumentation

Over the course of the project, the instrumentation of the manipulator has evolved significantly: valves, angle sensors, pressure sensors and control hardware have all been changed. We will only describe the hardware components that are currently in use on the manipulator.

2.4.1 Control system

All controllers except for the low-level pressure control (which is done by pressure servo valves) are implemented using a dSpace ACE1103PX4CLP rapid control prototyping system. The system's main component is the DS1103 PPC, a PowerPC based controller board with real time operating system that can be programmed using Matlab's Simulink. It has 36 16-bit ADC channels, 8 12-bit DAC channels, 6 incremental encoder interfaces and 50 digital I/O channels.

Since very fast sampling times are useless because of the slow dynamics of the pneumatic valves, the sampling time is set to 1 ms.

2.4.2 Pressure sensors

The gauge pressures in each muscle groups is measured by a Sensortechincs HCX005D6V differential pressure sensor. The pressure difference between the two sensing ports of the HCX005D6V is converted to a voltage and amplified to a range of 0.5 – 4.5 V by a built-in signal amplifier.

In order to measure a muscle's gauge pressure, one sensing port is connected to the muscle while the other is left open and thus subjected to atmospheric pressure.

Differential pressure sensors were used because they automatically eliminate the influence of atmospheric pressure variations.

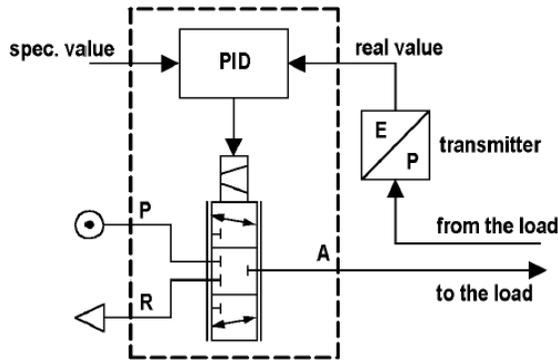


Figure 2.16: Overview of the Kolvenbach KPS 3/4-00 (figure taken from the valve’s datasheet).

2.4.3 Valves

The gauge pressure in the muscle groups is controlled by four Kolvenbach KPS 3/4-00 3/3-way pressure regulating servo valves. An overview of the valve is shown in figure 2.16.

A PID controller inside the valve controls the airflow into or out of the muscle by comparing the desired pressure (which is read from a voltage input) to the actual pressure in the muscle. Since the valves use external pressure sensors, the actual pressure is read from a voltage input as well. The pressure signal is based on the output of the Sensortechincs HCX005D6V pressure sensors described above.

2.4.4 Force sensors

Although not strictly necessary for control, the system is equipped with force sensors that measure the actuator forces. Muscle groups 1, 2 and 3 are all equipped with a Futek L2760 thru-hole force sensor. Because of space restrictions, it wasn’t possible to use the same sensor for muscle 4. Instead, it was equipped with a Futek L1650 (static) force sensor.

All force sensor signals are amplified using Futek JM-2AD loadcell amplifiers.

2.4.5 Encoders

2.4.5.1 Angle measurement

Joint angles are measured using two Agilent HEDM-6540#T13 incremental rotary optical encoders. This type of encoder contains a circular disk, rigidly attached to

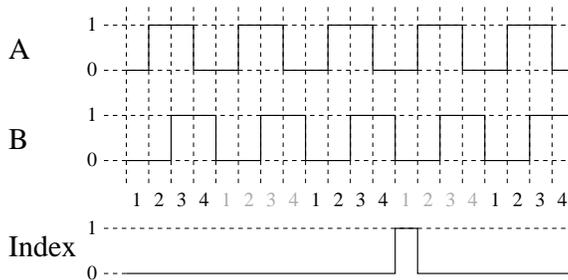


Figure 2.17: Example of the output signals of an incremental optical encoder.

the link, centered on its rotational axis and perpendicular to this axis. The disk thus rotates with the link, and can be used to measure its angular position.

The encoder disk contains a set of regularly spaced slots that pass in front of a light source as the disk rotates. On the other side of the disk two optical detectors measure the light that passes through the disk. As the disk rotates, the movement of the slots causes the light to alternate between passing through the disk or being blocked by it. The detectors thus measure light pulses, which are converted into digital voltage pulses. By counting the pulses one can keep track of the angular displacement of the link.

The two optical detectors are placed in such a way that their output pulse signals are in quadrature (i.e. have a 90° phase difference). This makes it possible to see if the angle is increasing or decreasing, by checking which signal leads the other one. An example of the encoder signals is shown in fig. 2.17.

The HEDM-6540#T13 encoder has 2000 slots, so for each revolution there are 2000 light-dark periods. Fig. 2.17 shows that by using both signals (usually called *A* and *B*) one can distinguish 4 different phases in each period. The encoder thus allows us to detect angle displacements of $360^\circ / (4 \cdot 2000) = 0.045^\circ$.

By counting pulses in the *A* and *B* signals, only changes in angle can be detected, there is no absolute reference. For this reason the encoder contains a third output, called index: if a certain (fixed) point on the disk passes before a detector, a single pulse is sent through the index line, which can be used as an absolute angle reference. Since after start-up absolute angles are unknown as long as no index pulse has been detected, an initialization procedure that makes sure the encoder passes its index position is necessary.

2.4.5.2 Angular velocity measurement

A good angular velocity measurement is important for control. Determining angular velocity based on encoder signals isn't generally easy, however, since we have to differentiate the angle signal numerically in real time (i.e. no "future" measurements

are available). While numerical differentiation is well known to magnify measurement errors, the fact that the angle signal is discretized in amplitude presents an extra difficulty: theoretically, the derivative of such a function is zero, except on points where a jump occurs (i.e. when a new pulse has been received), where the derivative is infinite.

Pulse counting To illustrate the problem, let's look at one of the most straightforward methods, pulse counting: we count the number of pulses N_p during the sampling interval T_s . If the smallest angle difference we can detect is called Δq , then our velocity estimate becomes

$$\omega_e \approx N_p \cdot \frac{\Delta q}{T_s}. \quad (2.23)$$

We see that the estimate is quantized: it is always an integer multiple of $\Delta q/T_s$, and $\Delta q/T_s$ is the smallest angular velocity we can measure (apart from zero). In our case, we have $\Delta q = 0.045^\circ$ and $T_s = 1$ ms, which results in $\Delta q/T_s \approx 45^\circ/s$. The angular velocity will thus be quantized in multiples of $45^\circ/s$, which is clearly unacceptable (for robotic applications $45^\circ/s$ is a considerable angular velocity).

From (2.23) we see that using encoders with a higher resolution (i.e. encoders with a smaller Δq) will improve the situation, as can be expected. On the other hand, increasing the sampling frequency $f_s = 1/T_s$ will make it worse. When f_s goes to infinity, or equivalently, when T_s goes to zero, we approach the theoretical situation: $\omega_e = 0$, except when $N_p \neq 0$, when ω_e becomes infinite. In practice, f_s can not be taken too small (or T_s too large), however, without compromising controller performance.

If we consider the “true” angular velocity to be given by $\omega = a \cdot \Delta q/T_s$ (with $a \in \mathbb{R}^+$), the absolute error on the estimation is given by

$$\begin{aligned} \delta_\omega &= |\omega - \omega_e| \\ &= \left| \frac{a \cdot \Delta q}{T_s} - N_p \cdot \frac{\Delta q}{T_s} \right| \\ &= \frac{\Delta q}{T_s} |a - N_p|. \end{aligned}$$

The relative error then becomes

$$\begin{aligned} \mu_\omega &= \frac{|\omega - \omega_e|}{\omega} \\ &= \frac{1}{a} |a - N_p|. \end{aligned}$$

Since $|a - N_p| < 1$ we have

$$\mu_\omega < \frac{1}{a}.$$

The higher a , or equivalently, the higher the angular velocity (since $\omega = a \cdot \Delta q / T_s$), the lower the relative error. The expected angular velocities in the system are low in terms of multiples of $\Delta q / T_s$, which is another argument against using pulse counting.

Pulse timing The other basic approach we can take is measuring the time Δt between two successive pulses. In digital implementations, however, time is always discrete, so we can only measure t relative to a clock signal with period T_c (with $T_c \lll T_s$ if we want to use the velocity signal in a control system with sampling time T_s):

$$\Delta t \approx N_c T_c,$$

with N_c the number of clock cycles counted in the time interval between the arrival of two successive encoder pulses. This results in the following angular velocity estimate:

$$\omega_e \approx \frac{1}{N_c} \cdot \frac{\Delta q}{T_c}. \quad (2.24)$$

This time, the velocity also involves a fixed quantum, $\Delta q / T_c$, but instead of being multiplied it is divided by an integer quantity N_c . This makes $\Delta q / T_c$ the largest angular velocity we can measure. If we write the true ω as $\Delta q / \Delta t$, the velocity error becomes

$$\begin{aligned} \delta_\omega &= |\omega - \omega_e| \\ &= \left| \frac{\Delta q}{\Delta t} - \frac{1}{N_c} \cdot \frac{\Delta q}{T_c} \right| \\ &= \frac{\Delta q}{\Delta t \cdot N_c T_c} |N_c T_c - \Delta t|, \end{aligned}$$

which results in the following relative error:

$$\begin{aligned} \mu_\omega &= \frac{|\omega - \omega_e|}{\omega} \\ &= \frac{1}{N_c T_c} |N_c T_c - \Delta t| \end{aligned}$$

Since $|N_c T_c - \Delta t| < T_c$ we get

$$\mu_\omega < \frac{1}{N_c}. \quad (2.25)$$

The higher N_c , the lower the relative error. Measuring the inter-pulse time is thus more precise at low velocities (the lower the speed, the higher N_c). This can be

seen more clearly by combining (2.25) and (2.24) to calculate the maximum relative error for a given angular velocity estimate ω_e :

$$\mu_\omega < \omega_e \cdot \frac{T_c}{\Delta q}. \quad (2.26)$$

As $T_c/\Delta q$ is fixed, we see that the relative error increases linearly with angular velocity.

Since the pulse counting method is not applicable in our case, and since the angular velocities in the 2-DOF manipulator will be relatively low, the pulse timing method seems to be the best method of the two to use. The dSpace control board, however, can't measure the time between encoder pulses. For this reason, a number of velocity measurement algorithms that only use encoder outputs (no timing) were investigated. Quite some papers exist on the subject, see for instance Brown et al. (1992); Su et al. (2006); Liu et al. (2004); Lee and Song (2001); Li et al. (2005); Janabi-Sharifi et al. (2000); Bélanger (1992); Tilli and Montanari (2001) and the references therein. None of the investigated algorithms were found to give satisfactory results, however, mostly due to phase delays (which can adversely affect control stability Brown et al. (1992)) or because they didn't cope well with sign reversals of the velocity. Since in simulation, the pulse timing method performed best, it was decided to add extra hardware that measures the time between encoder pulses.

2.4.5.3 Inter-pulse time measurement board

A PIC16F876A microcontroller based board was designed to measure the time between two successive encoder pulses. The principle of operation is the following: rising and falling edges in the A and B signals from the encoder are detected and trigger an interrupt. The interrupt routine records the number of clock cycles N_c measured since the last edge, restarts the timer (taking into account the time it has taken to reach that point in the code), presents N_c to a separate DAC-chip (DAC712P) and has it convert it to an analog voltage. This voltage is then periodically sampled by the control system, which uses eq. (2.24) to calculate the corresponding angular velocity estimate.

The two available input capture pins (the pins that trigger an interrupt when an edge occurs) posed an implementation problem, since they can only be set to trigger an interrupt on rising or on falling edges, but not on both. As figure 2.17 shows, it is necessary to detect the rising and falling edges in both the A and B signals in order to distinguish the four different "states" of the encoder. This would require four input capture pins on a PIC16F876A.

Fortunately, this problem can be solved by using a digital XOR circuit, as is shown in fig. 2.18. Any rising or falling edge in either A or B will cause an edge (rising or falling) in $A \oplus B$ (with \oplus meaning XOR). By presenting $A \oplus B$ to both input

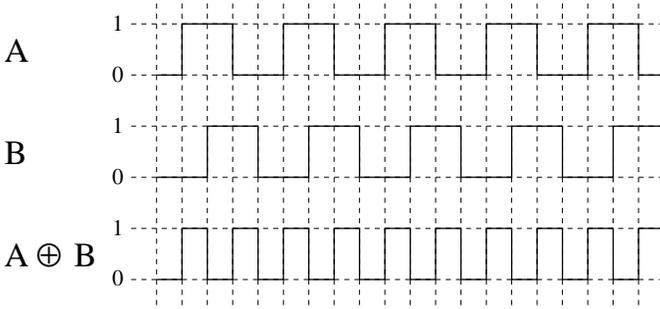


Figure 2.18: Encoder signals A and B , and the result of applying the XOR operator to both, $A \oplus B$.

capture pins, one configured for rising edges and the other one for falling edges, we can detect all edges. Of course, it is impossible to reconstruct A and B from $A \oplus B$, so the current state of A and B is read by means of other digital input pins.

The knowledge of N_c alone only allows the control board to calculate the absolute value of the angular velocity. For this reason, the inter-pulse timing board also has a digital output that is used to communicate the sign of the velocity.

The internal clock frequency in the microcontroller is 2 MHz, so $T_c = \frac{1}{2} \cdot 10^{-6}$. The number of clock cycles since the previous encoder pulse is stored as a 16-bit number, which will overflow after 2^{16} cycles. In that case, the board will output the maximum value of N_c , equal to $2^{16} - 1$, which implies that there is not only an upper limit to the velocity that can be measured, but also a lower limit:

$$\omega_{min} = \frac{1}{N_{c,max}} \cdot \frac{\Delta q}{T_c} \approx 1.37^\circ/s.$$

In order to have a margin, all measured angular velocities with absolute value below $0.025 \text{ rad/s} \approx 1.43^\circ/s$ are considered to be zero by the control board.

As mentioned above, the relative error increases with angular velocity. For a given ω determined using the inter-pulse timing board, the maximum relative error can be calculated from (2.26) using the previously given values of T_c and Δq . For an angular velocity of 5 rad/s (or around $286^\circ/s$, which is much higher than what we expect to see during operation), this gives $\mu_\omega < 0.32\%$, and for $\omega = 1 \text{ rad/s}$ (or almost $60^\circ/s$) we have $\mu_\omega < 0.06\%$. These levels of accuracy are surely high enough for robot control.

If angular velocity and acceleration are low, it can take a relatively long time for the next encoder pulse to arrive (if $\omega = \omega_{min}$, that time is equal to $N_{c,max}T_c \approx 32.8$ ms, or almost 33 sample periods of the control system). This means that it also takes a long time before the velocity that is communicated to the control system gets updated. The significant “lag” when low velocities and low accelerations are

present does not present practical problems, because of the low acceleration: it may take a while for the new velocity value to become available, but the new value doesn't differ much from the previous one.

2.4.6 Glue electronics

In order to combine all the sensors and the control electronics into one system, some extra “glue” electronics was necessary to solve a few practical problems:

- The range of the pressure sensors is 5 bar (or 500 kPa), with an output voltage between 0.5 V and 4.5 V. The Kolvenbach servo valves, however, expect a pressure signal that starts at 0 V for 0 bar and that rises with 1 V per bar of pressure. The signals from the pressure sensors first have to be transformed before they are presented to the valves.
- The sensors and the valves need electrical power in order to function, but only the encoders are powered by the dSpace control board. The multiwire cables connected to the devices contain both signal and power lines, so a component was needed that puts the correct voltage on the different devices' supply lines, while “routing” the signal lines to and from the dSpace control board. In case of the pressure sensors the transformed signal had to be presented to both the control board and the pressure regulating valves.

2.5 Summary

After introducing the Pleated Pneumatic Artificial Muscle (PPAM), the PPAM-specific parts of how the manipulator was designed were described in detail. The fact that muscle force varies with contraction means that the maximum available torque of the actuators is different in every point of the workspace, which complicates the design process.

The instrumentation of the manipulator was discussed as well, with emphasis on how a useful angular velocity signal was obtained from the encoder signals.

Chapter 3

Modeling

3.1 Introduction

The manipulator design presented in the previous chapter did not include any dynamic effects. The maximum load calculations, for instance, assume both static equilibrium and accurate knowledge of the muscle parameters and (some) inertial parameters. Although useful during the design phase, these are not a realistic assumptions. The different controllers described in part II depend on either static or dynamic models of the manipulator, the muscles and the servovalves. This chapter introduces the various models, all of them based on approximations, that are used in part II.

It can be expected that a better knowledge of the parameters present in these models will lead to better control performance. Instead of relying on theoretical values or values that were obtained using CAD models, experiments were performed to estimate the model parameters. The estimation procedures used for the different models are also discussed in this chapter.

Since the estimation experiments showed that hysteresis is not an unimportant effect in the system, a Preisach-based hysteresis model for the PPAM is introduced as well.

3.2 Mechanical model

The equations of motion of the arm's rigid body structure can be derived using for instance Lagrange's equations. It is well known (see for example Spong et al. (2006)) that they can be written as

$$H(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \boldsymbol{\tau}, \quad (3.1)$$

where $\mathbf{q} = [q_1 \ q_2]^T$ is the vector of joint angles, $H(\mathbf{q})$ is the inertia matrix, $C(\mathbf{q}, \dot{\mathbf{q}})$ is the centrifugal matrix (centrifugal and Coriolis forces), $\mathbf{G}(\mathbf{q})$ contains the gravitational torques in the joints and $\boldsymbol{\tau}$ is a vector representing the actuator torques, as given by eq. (2.17). Expressions for $H(\mathbf{q})$, $C(\mathbf{q}, \dot{\mathbf{q}})$ and $\mathbf{G}(\mathbf{q})$ are given in appendix B.

Note that the influence of the varying shape and position of the actuators is marginal and is thus not taken into account in this model. Due to the low mass of the PPAM, this is a reasonable approximation.

Friction is not included in (3.1) either, which will be discussed in section 3.5.2.3.

3.3 Pressure dynamics

There are two main factors that influence the actual gauge pressure in the muscles: the pressure regulating valves and the coupling between actuator gauge pressures and link angles and angular velocities.

Although elaborate models of pneumatic valves exist (see for instance Sorli et al. (2004); Falcão Carneiro and Gomes de Almeida (2006)), we will not attempt to develop a full model of the electro-pneumatic pressure regulating valves. Apart from the considerable difficulty involved in creating a detailed model (and estimating its parameters), such a model would greatly increase the complexity of the full system model. Its practical use would also require a lot of extra sensors (to measure for instance inlet pressure, outlet pressure, plunger position, etc.).

Instead, we will simply approximate the valve response by a first order system. If p is the pressure inside a muscle and p_d is the desired pressure then the approximated first order valve dynamics can be written as

$$\dot{p} = -p/T + p_d/T, \quad (3.2)$$

with T a time constant. Similar approximate models were used in for instance Inoue (1987); Tondu et al. (1994); Sira-Ramírez et al. (1996); Šitum and Herceg (2008).

A change in joint angle q also changes the contractions of the muscles driving that joint, which means their volume changes as well. The muscle volume V is thus a function of q , i.e. $V(q)$ (this can be seen in eq. (2.4) as well, since ϵ depends on q).

If we assume closed valves and polytropic compression/expansion, the pressure in the muscle only depends on its volume:

$$PV^n = P_0V_0^n \quad (3.3)$$

In this equation, P is the absolute pressure (equal to the sum of the atmospheric pressure P_{atm} and the gauge pressure p , i.e. $P = P_{atm} + p$), P_0 is the initial absolute

pressure, V_0 the initial volume and n the polytropic coefficient (in the isentropic limit we have $n = 1.4$ for dry air at room temperature).

From (3.3) we have

$$\begin{aligned} \frac{dP}{dq} &= \frac{d}{dq} \left(\frac{P_0 V_0^n}{V^n} \right) \\ &= P_0 V_0^n \cdot \frac{dV^{-n}}{dq} \\ &= -n \cdot P_0 V_0^n \cdot \frac{1}{V^{n+1}} \frac{dV}{dq} \\ &= -n \cdot PV^n \cdot \frac{1}{V^{n+1}} \frac{dV}{dq} \end{aligned}$$

or (using $P = P_{atm} + p$)

$$\frac{dp}{dq} = -n(P_{atm} + p) \cdot \frac{1}{V(q)} \cdot \frac{dV(q)}{dq}, \quad (3.4)$$

This model has been experimentally validated in (Vanderborght et al., 2008b) and Vanderborght et al. (2006b).

Equation (3.4) leads to

$$\dot{p} = \frac{dp}{dq} \frac{dq}{dt} = -n(P_{atm} + p) \cdot \frac{1}{V(q)} \cdot \frac{dV(q)}{dq} \dot{q}, \quad (3.5)$$

which describes the coupling between p and q and \dot{q} .

By combining (3.2) and (3.5) we obtain the (approximated) gauge pressure dynamics

$$\dot{p} = -\frac{p}{T} + \frac{pd}{T} - n(P_{atm} + p) \cdot \frac{1}{V(q)} \cdot \frac{dV(q)}{dq} \dot{q}. \quad (3.6)$$

3.3.1 Δp - approach

A link actuated by an antagonistic pair of pneumatic artificial muscles has two inputs, the gauge pressures of both muscles. Eqs. (2.17) and (2.18) show that both torque and compliance depend on these inputs. Within certain limits, having two inputs thus makes it possible to control torque and compliance at the same time. In many cases, however, we are only interested in controlling torque or position. In those cases, it is easier to have just a single input for each link (since there is only a single output that matters).

In pneumatic muscle systems, this reduction of the number of inputs is usually achieved by choosing an average pressure p_m for both muscles of the antagonistic

pair (Inoue, 1987). An amount of pressure Δp is then added to p_m in one muscle ($p_1 = p_m + \Delta p$), and subtracted from it in the other ($p_2 = p_m - \Delta p$), which leaves the link with only one input, Δp . Although both p_m and Δp determine torque and compliance, the choice of p_m mainly influences compliance while Δp mainly influences joint torque.

If we use the Δp -approach to calculate the desired pressure p_d , we can write (3.6) for all muscles,

$$\begin{aligned}
 \dot{p}_1 &= -\frac{p_1}{T_1} + \frac{p_{m_1} + \Delta p_1}{T_1} - n(P_{atm} + p_1) \cdot \frac{1}{V_1} \cdot \frac{dV_1}{dq_1} \dot{q}_1 \\
 \dot{p}_2 &= -\frac{p_2}{T_2} + \frac{p_{m_1} - \Delta p_1}{T_2} - n(P_{atm} + p_2) \cdot \frac{1}{V_2} \cdot \frac{dV_2}{dq_1} \dot{q}_1 \\
 \dot{p}_3 &= -\frac{p_3}{T_3} + \frac{p_{m_2} + \Delta p_2}{T_3} - n(P_{atm} + p_3) \cdot \frac{1}{V_3} \cdot \frac{dV_3}{dq_2} \dot{q}_2 \\
 \dot{p}_4 &= -\frac{p_4}{T_4} + \frac{p_{m_2} - \Delta p_2}{T_4} - n(P_{atm} + p_4) \cdot \frac{1}{V_4} \cdot \frac{dV_4}{dq_2} \dot{q}_2.
 \end{aligned} \tag{3.7}$$

with Δp_1 and Δp_2 (the desired values of Δp) the control inputs for the upper and lower arm joints, respectively, and p_{m_1} and p_{m_2} are the average pressures used in both joints.

3.4 Muscle dynamics

Reynolds et al. (2003) have proposed a dynamic model for McKibben muscles, that represents the McKibben muscle as a parallel connection of a nonlinear spring, dashpot and contractile element. All three components have coefficients that depend on the muscle pressure, which is considered to be the model's input.

No dynamic model exists for the PPAM, but experience suggests that it reacts very quickly to pressure changes. In practice, the pressure dynamics will almost certainly be much slower than any dynamics of the muscle itself. For this reason, no muscle dynamics (other than the influence of changes in volume on gauge pressure, as discussed above) will be considered.

3.5 Parameter estimation

This section explains how the parameters in the above described models have been estimated.

3.5.1 Static model

Most of the controllers described in part II only use gravity compensation in the feedforward part, i.e. they don't use the "dynamical" terms $H(\mathbf{q})\ddot{\mathbf{q}}$ and $C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}}$ in eq. (3.1). For this reason, it was initially decided to simplify the estimation problem by considering only the statical case. This section describes how parameters of the

muscles, the mechanical model and the valves were estimated, all assuming the system is in static equilibrium.

3.5.1.1 Pneumatic muscles

As noted in section 2.2.2, the dimensionless function f_{t0} present in the muscle force F (see (2.2)) can be approximated using a function of the following form:

$$f_{t0}(\epsilon) \approx f_0\epsilon^{-1} + f_1 + f_2\epsilon + f_3\epsilon^2 + f_4\epsilon^3.$$

We will use this approximation in a model for F as a function of gauge pressure and contraction,

$$F(p, \epsilon) = pl_0^2 (f_0\epsilon^{-1} + f_1 + f_2\epsilon + f_3\epsilon^2 + f_4\epsilon^3). \quad (3.8)$$

Model equation (3.8) can easily be rewritten in a form that is linear in the parameters:

$$F = p(a_0\epsilon^{-1} + a_1 + a_2\epsilon + a_3\epsilon^2 + a_4\epsilon^3) \quad (3.9)$$

$$= \begin{bmatrix} p\epsilon^{-1} & p & p\epsilon & p\epsilon^2 & p\epsilon^3 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \end{bmatrix} \quad (3.10)$$

$$= D(p, \epsilon) \mathbf{a} \quad (3.11)$$

with $a_j = l_0^2 f_j$ for $j = 0 \dots 4$, $D(p, \epsilon) = [p\epsilon^{-1} \quad p \quad p\epsilon \quad p\epsilon^2 \quad p\epsilon^3]$ and $\mathbf{a} = [a_0 \quad a_1 \quad a_2 \quad a_3 \quad a_4]^T$.

Given N measurements F_i , p_i and ϵ_i of force, gauge pressure and contraction, respectively, and by setting $D_i = D(p_i, \epsilon_i)$, we can combine all data into vector

$$\tilde{\mathbf{F}} = \begin{bmatrix} F_1 \\ F_2 \\ \vdots \\ F_N \end{bmatrix}$$

and matrix

$$\tilde{\mathbf{D}} = \begin{bmatrix} D_1 \\ D_2 \\ \vdots \\ D_N \end{bmatrix}$$

and write

$$\tilde{\mathbf{F}} = \tilde{D}\mathbf{a}.$$

The standard least-squares estimate of \mathbf{a} is then given by (Eykhoff, 1979)

$$\hat{\mathbf{a}} = \left(\tilde{D}^T \tilde{D} \right)^{-1} \tilde{D}^T \tilde{\mathbf{F}}.$$

Due to noise in \tilde{D} , the estimate will not be unbiased.

Since neither eq. (2.2) nor model equation (3.8) account for the hysteresis phenomenon observed in the muscle output (see Verrelst et al. (2006a) as well as section 3.6), a very good match between observation and prediction cannot be expected.

3.5.1.2 Valves

The Kolvenbach servo valves (see section 2.4.3) are supposed to set the pressure according to the rule 1 volt = 1 bar, i.e. given an input signal voltage of x V it should set the pressure to x bar (with 1 bar equal to 100 kPa). In reality, however, it is not uncommon to see deviations as high as 0.1 bar in steady state.

In order to accommodate this, a very simple model was proposed,

$$\begin{aligned} p &= v_1 + v_2 p_s & (3.12) \\ &= \begin{bmatrix} 1 & p_s \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \end{bmatrix} \\ &= E(p_s) \mathbf{v}, \end{aligned}$$

with p the measured pressure, p_s the set-value of the pressure (or the desired pressure), v_1 and v_2 the two model parameters to be estimated and $E(p_s)$ and \mathbf{v} defined by the above equation.

As above, given N measurements p_i of the gauge pressure and knowledge of the corresponding set-value $p_{s,i}$, and by setting $E_i = E(p_{s,i})$ we can form the vector

$$\tilde{\mathbf{P}} = \begin{bmatrix} p_1 \\ p_2 \\ \vdots \\ p_N \end{bmatrix}$$

and matrix

$$\tilde{E} = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_N \end{bmatrix}$$

to write

$$\tilde{\mathbf{P}} = \tilde{\mathbf{E}}\mathbf{v},$$

and estimate \mathbf{v} using (Eykhoff, 1979)

$$\hat{\mathbf{v}} = \left(\tilde{\mathbf{E}}^T \tilde{\mathbf{E}} \right)^{-1} \tilde{\mathbf{E}}^T \tilde{\mathbf{P}}.$$

3.5.1.3 Mechanical model

Since we're considering the statical case, we only have to estimate the parameters in $\mathbf{G}(\mathbf{q})$ of eq. (3.1). The influence of $H(\mathbf{q})$ and $C(\mathbf{q}, \dot{\mathbf{q}})$ can easily be excluded by measuring only when the system is in static equilibrium, i.e. when $\dot{\mathbf{q}} = \ddot{\mathbf{q}} = 0$. In that case, (3.1) becomes

$$\mathbf{G}(\mathbf{q}) = \boldsymbol{\tau}. \quad (3.13)$$

Since $\mathbf{G}(\mathbf{q})$ is given by

$$\mathbf{G}(\mathbf{q}) = \begin{bmatrix} g_1(\mathbf{q}) \\ g_2(\mathbf{q}) \end{bmatrix} \quad (3.14)$$

with (see appendix B)¹

$$\begin{aligned} g_1(\mathbf{q}) &= g (\sin(q_1) d_{G_1} m_1 + \sin(q_1 + q_2) d_{G_2} m_2 + \cos(q_1 + q_2) L_{G_2} m_2 \\ &\quad + \cos(q_1) (L_{G_1} m_1 + L_1 m_2)) \\ g_2(\mathbf{q}) &= g (\sin(q_1 + q_2) d_{G_2} m_2 + \cos(q_1 + q_2) L_{G_2} m_2), \end{aligned}$$

we see that is it possible to write it as

$$\begin{aligned} \mathbf{G}(\mathbf{q}) &= \begin{bmatrix} g (\sin(q_1) \theta_4 + \sin(q_1 + q_2) \theta_3 + \cos(q_1 + q_2) \theta_2 + \cos(q_1) \theta_1) \\ g (\sin(q_1 + q_2) \theta_3 + \cos(q_1 + q_2) \theta_2) \end{bmatrix} \\ &= \begin{bmatrix} g \cos(q_1) & g \cos(q_1 + q_2) & g \sin(q_1 + q_2) & g \sin(q_1) \\ 0 & g \cos(q_1 + q_2) & g \sin(q_1 + q_2) & 0 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \end{bmatrix} \\ &= K(\mathbf{q}) \boldsymbol{\theta} \end{aligned} \quad (3.15)$$

with K and $\boldsymbol{\theta}$ defined by the above equation and

$$\begin{aligned} \theta_1 &= m_1 L_{G_1} + m_2 L_1 \\ \theta_2 &= m_2 L_{G_2} \\ \theta_3 &= m_2 d_{G_2} \end{aligned}$$

¹See fig. 2.9 on page 19 to see how the various distances are defined.

$$\theta_4 = m_1 d_{G_1}.$$

$\mathbf{G}(\mathbf{q})$ is thus linear in the parameters θ_i , and eq. (3.13) becomes

$$K(\mathbf{q})\boldsymbol{\theta} = \boldsymbol{\tau}. \quad (3.16)$$

Knowledge of \mathbf{q} (which is known from the encoders) is enough to calculate $K(\mathbf{q})$. In principle, $\boldsymbol{\tau}$ could be calculated from eq. (2.17), using measured gauge pressures. Eq. (2.17), however, is based on the muscle model (2.2), which isn't very accurate (it doesn't account for the hysteresis seen in the muscles, for instance, and it also depends on the muscle slenderness l_0/R , which is only approximatively known). For this reason $\boldsymbol{\tau}$ is determined based on force sensor readings, by using

$$\boldsymbol{\tau} = \begin{bmatrix} F_1 (\mathbf{e}_1 \times \mathbf{OA}_{12}) \cdot \mathbf{1}_z + F_2 (\mathbf{e}_2 \times \mathbf{OA}_{22}) \cdot \mathbf{1}_z \\ F_3 (\mathbf{e}_3 \times \mathbf{SA}_{33}) \cdot \mathbf{1}_z + F_4 (\mathbf{e}_4 \times \mathbf{SA}_{43}) \cdot \mathbf{1}_z \end{bmatrix}, \quad (3.17)$$

which follows from (2.16) and (A.1)-(A.4).

By measuring joint angles and muscle forces N times in many different configurations (while making sure the arm is always in static equilibrium), and calculating the matrix $K_i = K(\mathbf{q}_i)$ and torque $\boldsymbol{\tau}_i$ for each measurement i , we can form the observation matrix \tilde{K} by stacking all K_i in a single $2N \times 4$ matrix:

$$\tilde{K} = \begin{bmatrix} K_1 \\ K_2 \\ \vdots \\ K_N \end{bmatrix}.$$

Similarly, we have

$$\tilde{\boldsymbol{\tau}} = \begin{bmatrix} \boldsymbol{\tau}_1 \\ \boldsymbol{\tau}_2 \\ \vdots \\ \boldsymbol{\tau}_N \end{bmatrix},$$

a $2N \times 1$ vector. Eq. (3.16) thus becomes

$$\tilde{K}\boldsymbol{\theta} = \tilde{\boldsymbol{\tau}}.$$

Again, the least-squares estimate of $\boldsymbol{\theta}$ is given by (Eykhoff, 1979)

$$\hat{\boldsymbol{\theta}} = \left(\tilde{K}^T \tilde{K} \right)^{-1} \tilde{K}^T \tilde{\boldsymbol{\tau}}.$$

Parameter	Estimated value	Units
$m_1 L_{G_1} + m_2 L_1$	0.5267	$kg \cdot m$
$m_2 L_{G_2}$	0.2943	$kg \cdot m$
$m_2 d_{G_2}$	0.0570	$kg \cdot m$
$m_1 d_{G_1}$	-0.0806	$kg \cdot m$

Table 3.1: Estimated parameters of the static mechanical model (cf. eq. (3.15)).

3.5.1.4 Experiment

The parameter vectors \mathbf{a} , \mathbf{v} and $\boldsymbol{\theta}$ (of the muscle model, the valve model and the (static) mechanical model, respectively) were all estimated using data from the same experiment, which started with the desired pressure for all muscles set to 1.5 *bar* (or 150 *kPa*) and the system in a state of static equilibrium ($\dot{\mathbf{q}} = \ddot{\mathbf{q}} = 0$). Each step of the experiment involved slightly and gradually changing the desired gauge pressure in each muscle over a period of 1 second, waiting for 3 seconds to ensure static equilibrium and then taking data for a period of 4 seconds with a sampling frequency of 1 *kHz*. Data was recorded for 31 of these steps throughout the workspace.

Values of the components of $\boldsymbol{\theta}$ as estimated in the static identification procedure are shown in table 3.1. The other estimated parameter values can be found in appendix B, section B.6.

3.5.1.5 Gravity compensation

Since the goal of the static identification procedure was to obtain a usable gravity compensation, the quality of the gravity compensation obtained will be used to judge the quality of the estimated parameters.

For gravity compensation, we have to calculate the actuator gauge pressures necessary to have static equilibrium in a certain position (represented by angle vector \mathbf{q}). By assuming static equilibrium and combining the Δp -approach (see section 3.3.1) with eq. (2.17), we have

$$\begin{aligned} \mathbf{G}(\mathbf{q}) &= \begin{bmatrix} p_1 \cdot m_{\tau_1}(q_1) + p_2 \cdot m_{\tau_2}(q_1) \\ p_3 \cdot m_{\tau_3}(q_2) + p_4 \cdot m_{\tau_4}(q_2) \end{bmatrix} \\ &= \begin{bmatrix} p_m (m_{\tau_1}(q_1) + m_{\tau_2}(q_1)) + \Delta p_1 (m_{\tau_1}(q_1) - m_{\tau_2}(q_1)) \\ p_m (m_{\tau_3}(q_2) + m_{\tau_4}(q_2)) + \Delta p_2 (m_{\tau_3}(q_2) - m_{\tau_4}(q_2)) \end{bmatrix}, \end{aligned}$$

or (using (3.14))

$$\Delta p_{1,gc} = \frac{g_1(\mathbf{q}) - p_m (m_{\tau_1}(q_1) + m_{\tau_2}(q_1))}{m_{\tau_1}(q_1) - m_{\tau_2}(q_1)} \quad (3.18)$$

$$\Delta p_{2,gc} = \frac{g_2(\mathbf{q}) - p_m(m_{\tau 3}(q_2) + m_{\tau 4}(q_2))}{m_{\tau 3}(q_2) - m_{\tau 4}(q_2)}. \quad (3.19)$$

The terms $g_1(\mathbf{q})$ and $g_2(\mathbf{q})$ can be calculated from eq. (3.15) using the estimated parameters $\hat{\theta}$:

$$\begin{bmatrix} g_1(\mathbf{q}) \\ g_2(\mathbf{q}) \end{bmatrix} = K(\mathbf{q}) \hat{\theta}.$$

The torque functions m_{τ_i} in (3.18)-(3.19) can also be expressed as a function of the estimated parameters of the muscle model, provided their definition is slightly changed to use (3.8) instead of (2.2). The modified definition is given in section B.2 of appendix B.

Given the desired gauge pressures for gravity compensation, the set-value necessary to achieve this pressure is calculated from (3.12) for all valves.

3.5.1.6 Validation

In order to test the estimated parameters using the above described gravity compensation, the desired values for q_1 and q_2 were varied in steps following a sawtooth pattern. After each (gradual) change in desired angles, a 3 second pause was taken to make sure the system was in static equilibrium, and then data was taken during 1 second.

The desired and measured joint angles for this experiment are shown in fig. 3.1. The root mean square angle errors in this validation experiment were 2.47° for q_1 , and 2.06° for q_2 .

Fig. 3.1 clearly shows that for increasing joint angles, the measured angle is generally less than the desired angle, and for decreasing angles the reverse is true. A similar effect is present in the forces, as can be seen in fig. 3.2, which shows the forces measured in the muscles that drive the second link, as well as the forces that were expected based on pressure measurements and on the estimated parameters (calculated using eq. (3.9)).

By comparing the two figures (which share the same scale on the horizontal axis), it can be seen that when q_2 is increased in steps, force F_3 is lower than what was expected, while F_4 is higher. An increase in q_2 means an increase in contraction for muscle 3 (the muscle shortens), and a decrease in contraction for muscle 4 (it gets longer). Thus, if contraction increases, the output force is lower than predicted by the model, while as contraction decreases it is higher. The same effect exists for the muscles in joint 1.

This strongly suggests that hysteresis in the force-contraction characteristic of the muscle (which is not included in the model) is the underlying cause for the effect seen in the angles, with the model providing an ‘‘average’’ expected force, too high when contraction is increased and too low when it is decreased. When an angle is increased, the top muscle exerts less force than predicted, while its antagonist

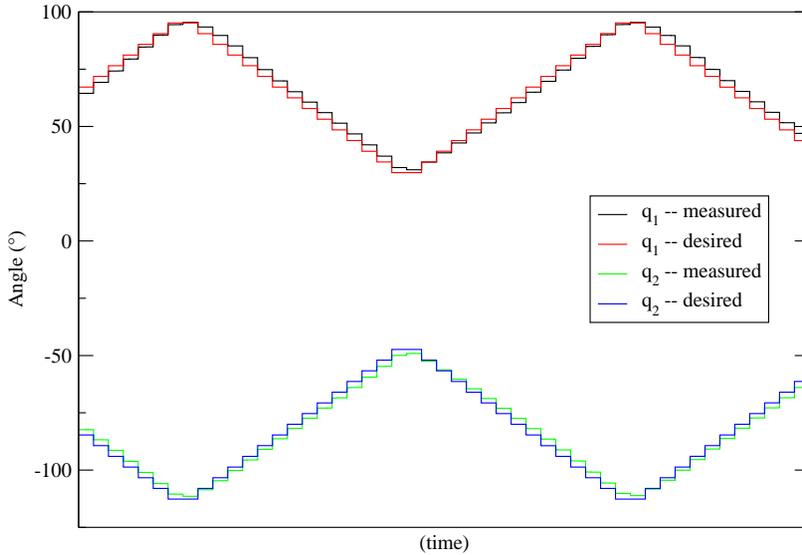


Figure 3.1: Comparison between measured and desired angles during the gravity compensation experiment. Please note that the horizontal axis doesn't represent true time, since samples were only taken at specific times during the experiment.

exerts more force. Both effects add up in the torque, which is lower than predicted, causing the joint's angle to be lower than the desired value.

Hysteresis is further discussed in section 3.6.

In order to check if bias in the muscle parameter estimation doesn't somehow aggravate the effect, the muscle identification was redone with the instrumental variables method (Eykhoff, 1979). The expected pressure in the muscles according to (3.12) (which should be unbiased and provides an excellent match between measurement and prediction) was used for the instrumental variables. The forces predicted by the model hardly changed using the new estimates, however (less than 0.01 % change).

Although not very accurate, the gravity compensation was used as a feedforward term in control with good results, as described in chapter 5.

3.5.2 Dynamical case

As said in section 3.4, no dynamic muscle model will be considered. The parameters identified statically will be considered to hold in the dynamical case as well.

The model for the pressure dynamics (3.7) introduced in section 3.3 depends on the valve time constants T_i (with $i = 1 \dots 4$) and on the muscle volume functions V_i ,

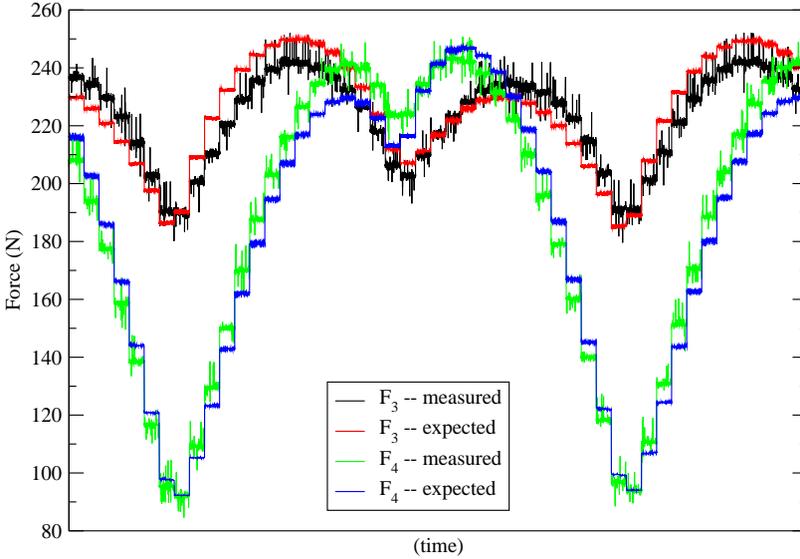


Figure 3.2: Comparison between measured and expected forces in the muscles of link 2 during the gravity compensation experiment. Please note that the horizontal axis doesn't represent true time, since samples were only taken at specific times during the experiment.

as defined in (2.4). Since muscle volume is very difficult to measure, the theoretical volume functions were used.

The valve time constants were determined by hooking up the valve to a closed volume and measuring the rise time of the response when applying an input step of 1 bar. With t_r the rise time, the time constant is then given by $T = t_r / \ln 9$.

The rest of this section is devoted to estimating the parameters of the mechanical model.

3.5.2.1 Introduction

In order to estimate the forces acting on the system (see section 5.4.2), it is also necessary to know the matrices $H(\mathbf{q})$ and $C(\mathbf{q}, \dot{\mathbf{q}})$ that appear in (3.1). As is well known, (3.1) can be written in a form that is linear in the parameters (Khosla and Kanade, 1985; An et al., 1985; Nicolò and Katende, 1983), i.e.

$$K(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \boldsymbol{\theta} = \boldsymbol{\tau}. \quad (3.20)$$

Both the parameter vector $\boldsymbol{\theta}$ and the matrix $K(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ for the dynamical case are given in appendix B.

3.5.2.2 Accelerations

The main difficulty in using (3.20) for parameter estimation is the presence of the angular acceleration $\ddot{\mathbf{q}}$, which can't be measured directly since the system isn't equipped with accelerometers. The angular velocity isn't measured directly either, but as seen in section 2.4.5 the system is equipped with pulse counting hardware to estimate $\dot{\mathbf{q}}$ from the encoder signals, which is equivalent to numerically differentiating \mathbf{q} .

The two most often used approaches to find angular velocities and accelerations for identification purposes in literature are the following:

- Using bandlimited periodic excitation trajectories (see for instance Swevers et al. (1997)), which allows the measured encoder signals to be approximated by a finite sum of sines and cosines. Angular velocity and acceleration can then be calculated analytically.

This approach cannot be used for the manipulator considered here, since it requires the excitation trajectories to be tracked with high accuracy. Systems actuated by pneumatic muscles don't currently offer the necessary tracking performance to make this approach feasible.

- Offline numerical differentiation of joint angle data, usually with bandpass filtering (Gautier, 1997; Hollerbach et al., 2008).

Since we will later also need to evaluate the equations of motion in real-time (see section 5.4.2), it was decided not to implement this technique.

Instead, we have opted for a method that avoids the explicit calculation of $\ddot{\mathbf{q}}$, as described in Li and Slotine (1987); Slotine and Li (1991). A similar technique can be found in Hsu et al. (1987).

The idea is to filter both sides of eq. (3.1) using a stable and proper filter. We have used a first order filter with transfer function

$$F(s) = \frac{1}{s/\omega + 1} \quad (3.21)$$

and impulse response

$$f(t) = \mathcal{L}^{-1}\{F(s)\} = \omega e^{-\omega t}. \quad (3.22)$$

Since multiplication with transfer function (3.21) in the frequency domain is equivalent to a convolution with the impulse response in the time domain, the filtered version of (3.1) is given by

$$\int_0^t f(t-r) \{H(\mathbf{q}) \ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{G}(\mathbf{q})\} dr = \int_0^t f(t-r) \boldsymbol{\tau}(r) dr \quad (3.23)$$

(where we temporarily consider all time dependent quantities to depend on r instead of t , and consider the dot notation to mean differentiation with respect to r). The first term on the left-hand side can be integrated using integration by parts²:

$$\begin{aligned} \int_0^t f(t-r) \{H(\mathbf{q}) \ddot{\mathbf{q}}\} dr &= f(t-r) H(\mathbf{q}) \dot{\mathbf{q}}|_0^t - \int_0^t \frac{d}{dr} \{f(t-r) H(\mathbf{q})\} \dot{\mathbf{q}} dr \\ &= f(0) H(\mathbf{q}) \dot{\mathbf{q}} - f(t) H(\mathbf{q}(0)) \dot{\mathbf{q}}(0) \\ &\quad - \int_0^t \left\{ f(t-r) \dot{H}(\mathbf{q}) \dot{\mathbf{q}} - \dot{f}(t-r) H(\mathbf{q}) \dot{\mathbf{q}} \right\} dr. \end{aligned}$$

The angular acceleration $\ddot{\mathbf{q}}$ no longer appears in this expression. If we assume the initial angular velocity $\dot{\mathbf{q}}(0)$ to be zero, the left-hand side of (3.23) becomes (using $f(0) = \omega$, which follows from (3.22))

$$\omega H(\mathbf{q}) \dot{\mathbf{q}} + \int_0^t f(t-r) \left\{ C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) - \dot{H}(\mathbf{q}) \dot{\mathbf{q}} \right\} dr + \int_0^t \dot{f}(t-r) \{H(\mathbf{q}) \dot{\mathbf{q}}\} dr. \quad (3.24)$$

All convolutions in this expression can be calculated by filtering³, using transfer function (3.21) for the first convolution and

$$\begin{aligned} F_2(s) &= \mathcal{L} \left\{ \dot{f}(t) \right\} \\ &= \mathcal{L} \left\{ -\omega^2 e^{-\omega t} \right\} \\ &= -\frac{\omega^2}{\omega + s} \end{aligned} \quad (3.25)$$

for the second convolution. Likewise, the right-hand side of (3.23) can be calculated by filtering with (3.21).

Since (3.24) contains terms that don't appear in the original equation (3.1), and since different filters are applied to different terms, it is clear that the previously calculated observation matrix $K(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ cannot be used in this context. Instead, we have to express the various terms that appear in (3.24) as a (linear) function of the parameter vector $\boldsymbol{\theta}$, i.e.

$$\begin{aligned} H(\mathbf{q}) \dot{\mathbf{q}} &= W_1(\mathbf{q}, \dot{\mathbf{q}}) \boldsymbol{\theta} \\ \dot{H}(\mathbf{q}) \dot{\mathbf{q}} &= W_2(\mathbf{q}, \dot{\mathbf{q}}) \boldsymbol{\theta} \end{aligned}$$

²In the formula for integration by parts, $\int u dv = uv - \int v du$, we set $u = f(t-r) H(\mathbf{q})$ and $dv = \dot{\mathbf{q}} dr$, so $v = \dot{\mathbf{q}}$ and $du = \frac{d}{dr} \{f(t-r) H(\mathbf{q})\} dr$.

³In Matlab[®], this can be done using the `lsim` function of the Control System Toolbox, for example.

$$C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = W_3(\mathbf{q}, \dot{\mathbf{q}}) \boldsymbol{\theta}.$$

The matrices $W_1(\mathbf{q}, \dot{\mathbf{q}})$, $W_2(\mathbf{q}, \dot{\mathbf{q}})$ and $W_3(\mathbf{q}, \dot{\mathbf{q}})$ are defined in appendix B (section B.4).

If we use the notation $\langle x(t) \rangle_{F(s)}$ to indicate the signal that results from filtering time signal $x(t)$ with the filter with transfer function $F(s)$, the filtered equations of motion (3.23) can thus be written as (using (3.24))

$$\left(\omega W_1(\mathbf{q}, \dot{\mathbf{q}}) + \langle W_3(\mathbf{q}, \dot{\mathbf{q}}) - W_2(\mathbf{q}, \dot{\mathbf{q}}) \rangle_{F(s)} + \langle W_1(\mathbf{q}, \dot{\mathbf{q}}) \rangle_{F_2(s)} \right) \boldsymbol{\theta} = \langle \boldsymbol{\tau} \rangle_{F(s)},$$

with $F(s)$ and $F_2(s)$ defined by eqs. (3.21) and (3.25), respectively. By comparing with eq. (3.20) we see that

$$\langle K(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \rangle_{F(s)} = \left(\omega W_1(\mathbf{q}, \dot{\mathbf{q}}) + \langle W_3(\mathbf{q}, \dot{\mathbf{q}}) - W_2(\mathbf{q}, \dot{\mathbf{q}}) \rangle_{F(s)} + \langle W_1(\mathbf{q}, \dot{\mathbf{q}}) \rangle_{F_2(s)} \right) \quad (3.26)$$

$$= K_f(\mathbf{q}, \dot{\mathbf{q}}), \quad (3.27)$$

the advantage of the right-hand side being that it doesn't explicitly contain the angular acceleration $\ddot{\mathbf{q}}$.

It is clear that the cutoff frequency of the first-order filter (3.21) will influence the results. Ideally, it should be chosen higher than the highest meaningful frequency component present in the measured torques, but not too high in order to smoothen the noise on position and velocity measurements. Since the manipulator will generally move slowly, the measured signals are expected to have only very low frequency components. Therefore, the cutoff frequency was chosen to be 5 Hz (which means $\omega = 10\pi \text{ rad/s}$).

Practical implementation As before, if N measurements are taken they have to be combined, but the filtering makes this process a bit more complicated. The details are explained in appendix B, section B.5 (page 177). The final result is a filtered version of eq. (3.20):

$$\tilde{K}_f \boldsymbol{\theta} = \tilde{\boldsymbol{\tau}}_f. \quad (3.28)$$

3.5.2.3 Friction

Most industrial robots don't have joint-torque sensors, the torques are estimated by measuring motor currents. Since joint friction typically consumes a large fraction of the torque that the motor produces (Hollerbach et al., 2008), a friction model is usually included in the dynamical equations (3.1), and its parameters are estimated in the identification procedure.

It is unclear if the inclusion of a friction model would be beneficial in the case considered here. There are no joint torque sensors, instead joint torque is calculated

by means of muscle force sensors. Part of the muscle force is consumed by friction in the attachment points, and since muscle force can vary greatly, so will friction. The angular velocity of the muscle with respect to the link is generally very low, which makes the friction characteristics more complex.

In addition, there are several air-supply tubes and cables running along the links, which results in extra “friction” (or unmodeled disturbance torques). The motion of the air within muscles and tubes is another possible source of friction.

It was decided to do the parameter estimation procedure both with and without friction model and proceed with the model that gives the best results. The friction model used consists only of Coulomb friction and viscous friction, an acceptable simplification for many robotics applications (Swevers et al., 2007), although it might not be in this case. Its advantages are simplicity and the fact that it is linear in the parameters.

A friction term $\boldsymbol{\tau}_f$ is thus added to the left-hand side of (3.1):

$$H(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) + \boldsymbol{\tau}_f(\dot{\mathbf{q}}) = \boldsymbol{\tau}, \quad (3.29)$$

with

$$\boldsymbol{\tau}_f(\dot{\mathbf{q}}) = \mathbf{c}^T \operatorname{sgn}(\dot{\mathbf{q}}) + \mathbf{b}^T \dot{\mathbf{q}} \quad (3.30)$$

and $\mathbf{c}^T = [c_1 \quad c_2]$ the Coulomb friction coefficients and $\mathbf{b}^T = [b_1 \quad b_2]$ the viscous friction coefficients.

Of course, this influences the definition of $\boldsymbol{\theta}$ (which must now include the friction parameters) and $\langle K(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \rangle_{F(s)}$ (as given in (3.26)). The changes are shown in appendix B, section B.4.1 (page 176).

3.5.2.4 Excitation

Swevers et al. (1996, 1997) have introduced the use of bandlimited periodic excitation trajectories to robotics. The main advantages (other than the analytical calculation of velocities and accelerations mentioned earlier) are the possibility to do time-domain averaging of the (noisy) torque data, and the fact that the characteristics of the measurement noise can be estimated. Several derived methods exist as well (Calafiore et al., 2001; Park, 2006).

Most excitation trajectory generation methods available in literature target industrial robots and assume that the robot is always operating under feedback control. In the case of the pneumatic manipulator considered here, it is perfectly possible to excite it in open loop⁴, however. Since the tracking control available for pneumatic muscle systems can't track excitation trajectories closely enough (as mentioned

⁴In this context, we mean by “open loop” that there is no position or trajectory controller. The built-in pressure controllers in the pneumatic servo valves (see section 2.4.3) cannot be bypassed, however, so the muscle gauge pressures are always feedback controlled.

above), it is logical to opt for open loop excitation. This means we don't need an excitation trajectory, but an excitation input.

Inspired by Swevers et al. (1997), finite Fourier series were used as excitation signals for both links⁵:

$$\Delta p_i = \sum_{k=1}^{N'} (a_{k,i} \sin(k\omega t) + b_{k,i} \cos(k\omega t)),$$

with i the link number. Once transients have died out the system's response to such an excitation is also periodic.

The number of frequency components was chosen to be 5 ($N' = 5$), and the signal period was set to 10 seconds ($\omega = 2\pi/10$).

The coefficients $a_{k,i}$ and $b_{k,i}$ were determined by minimizing the condition number of the normalized $\tilde{K}_f^T \tilde{K}_f$ matrix (Otani and Kakizaki, 1993), with \tilde{K}_f defined in (B.7). Since excitation inputs are used instead of trajectories, this is computationally very expensive: the full system response has to be calculated (simulated) over a time interval of at least two full periods of the excitation signal to obtain \tilde{K}_f (when given the $a_{k,i}$ and $b_{k,i}$). In more detail, the following has to be done for each evaluation of the objective function during the minimization:

- Suitable initial conditions for the system simulation have to be calculated by numerically solving the static equilibrium equations (3.13) for \mathbf{q} (when given Δp_i). This makes sure that transients disappear relatively quickly, so only the first simulated period will be affected by them.
- The system's response to two input signal periods (or 20 seconds) is calculated. Only the second period is used in further steps, since it isn't affected by the initial transients.
- Given the 10 seconds of simulated data from the second period, the filtered observation matrix \tilde{K}_f is calculated using (3.26) and (B.7), and the condition number of the normalization of $\tilde{K}_f^T \tilde{K}_f$ is calculated.

The minimization procedure is also subject to a number of constraints: for any given input signal to be acceptable, the manipulator has to remain within its workspace, and angular velocities and accelerations have to remain below certain limits (which were chosen as $|\dot{q}_i| \leq 2 \text{ rad/s}$ and $|\ddot{q}_i| \leq 5 \text{ rad/s}^2$). Since the system response has to be known to evaluate the constraints, the steps given above apply to all evaluations of the constraint function as well.

It should be noted that since the input signal optimization is simulation-based, the procedure has to assume the inertial parameters to be known. The result from

⁵Due to the use of the Δp -approach (see section 3.3.1), only one input per link is necessary.

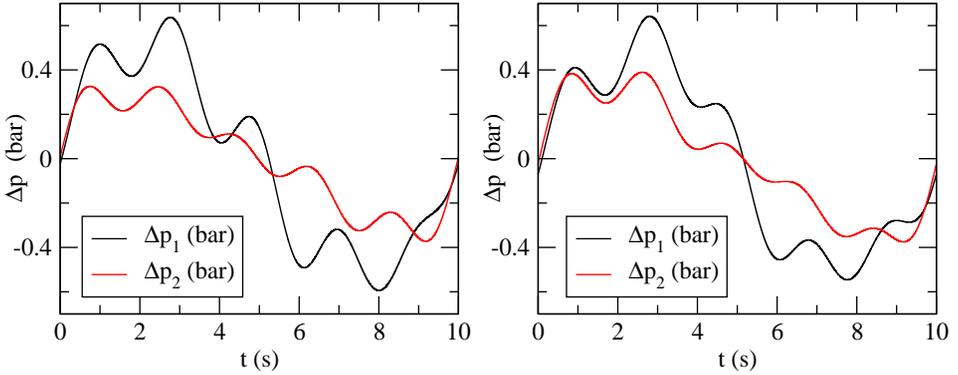


Figure 3.3: Inputs used to excite the system as calculated by the optimization procedure. Results calculated with friction are shown on the left, without friction on the right (1 bar = 100 kPa).

the optimization will be less optimal when applied to the real system, which has different inertial parameters and unmodeled dynamics.

The resulting excitation signals for the models with friction and without friction are shown in figure 3.3. In the following, the excitation signal calculated with the friction model is called excitation signal 1, while the other one is called excitation signal 2.

Both excitation signals were applied to the pneumatic arm, and 21 full periods were measured. The data from the first period was then discarded (since it contains transients), leaving us with 20 periods of data for each signal, or 200000 measurements (given the sampling period T_s of 1 millisecond and the signal period of 10 seconds).

3.5.2.5 Noise

To get an idea of the measurement noise, the average value of a signal at each timestep (calculated from the 20 samples available) was calculated. The variance with respect to that average was then determined (at each timestep), and all variances were averaged over all timesteps. If we take a signal $x(t)$ as an example, and write $x_j[k]$ as the k 'th sample taken from $x(t)$ during excitation period j , the average over the number of periods M is given by

$$\bar{x}[k] = \frac{1}{M} \sum_{j=1}^M x_j[k].$$

Signal	$\hat{\sigma}_1^2$	$\hat{\sigma}_2^2$	Units
q_1	3.46×10^{-6}	2.06×10^{-6}	rad^2
q_2	2.59×10^{-6}	6.63×10^{-7}	rad^2
\dot{q}_1	1.63×10^{-4}	1.25×10^{-4}	rad^2/s^2
\dot{q}_2	2.32×10^{-5}	2.03×10^{-5}	rad^2/s^2
F_1	0.51	0.33	N^2
F_2	1.02	1.15	N^2
F_3	0.59	0.47	N^2
F_4	0.45	0.43	N^2
τ_1	1.82×10^{-3}	1.77×10^{-3}	$N^2 m^2$
τ_2	1.01×10^{-3}	1.00×10^{-3}	$N^2 m^2$

Table 3.2: Noise variances as estimated after application of the first ($\hat{\sigma}_1^2$) and second ($\hat{\sigma}_2^2$) excitation signal. τ_1 and τ_2 weren't directly measured, but were calculated from the other signals using eq. (3.17).

The sample variance is then given by

$$s_x^2[k] = \frac{1}{M-1} \sum_{j=1}^M (x_j[k] - \bar{x}[k])^2,$$

which leads to the following overall variance estimate if there are Q samples per period:

$$\begin{aligned} \hat{\sigma}_x^2 &= \frac{1}{Q} \sum_{k=1}^Q s_x^2[k] \\ &= \frac{1}{Q} \frac{1}{M-1} \sum_{k=1}^Q \sum_{j=1}^M (x_j[k] - \bar{x}[k])^2. \end{aligned}$$

For the data being considered here, we have $Q = 10000$ and $M = 20$.

Estimated noise variances for all important signals are shown in table 3.2.

3.5.2.6 Parameter estimates

From (3.28), the estimated parameter vector $\hat{\boldsymbol{\theta}}$ was calculated in two ways:

- Using the least-squares (LS) estimator (Eykhoff, 1979):

$$\hat{\boldsymbol{\theta}}_{LS} = \left(\tilde{K}_f^T \tilde{K}_f \right)^{-1} \tilde{K}_f^T \tilde{\boldsymbol{r}}_f.$$

Signal	$\hat{\sigma}_1^2$	$\hat{\sigma}_2^2$	Units
$\tau_{1,f}$	5.57×10^{-4}	6.89×10^{-4}	N^2m^2
$\tau_{2,f}$	4.28×10^{-5}	6.24×10^{-5}	N^2m^2

Table 3.3: Noise variances of the filtered torques as estimated after application of the first ($\hat{\sigma}_1^2$) and second ($\hat{\sigma}_2^2$) excitation signal. $\tau_{1,f}$ and $\tau_{2,f}$ weren't directly measured, but were calculated using eq. (3.17) and then filtered with (3.21).

- Using the weighted least-squares (WLS) estimator (or Markov estimator (Eykhoff, 1979)):

$$\hat{\theta}_{WLS} = \left(\tilde{K}_f^T \Sigma^{-1} \tilde{K}_f \right)^{-1} \tilde{K}_f^T \Sigma^{-1} \tilde{\tau}_f,$$

with Σ the diagonal covariance matrix of the measured actuator torques. The torque variances from table 3.2 can't be used, however, since they are based on unfiltered torque measurements. The noise variances after filtering are given in table 3.3.

Both estimates will be biased due to noise in \tilde{K}_f (which stems mainly from measurement error in \dot{q}).

Regardless of how a dataset was generated (i.e. with the excitation signal optimized with friction or without), both the above estimators were used two times on each dataset, once assuming there is friction and once assuming there is not. Both datasets thus generated 4 different parameter estimates each.

3.5.2.7 Validation

Two different periodic validation signals were applied to the system, and a few periods of data were measured. For each parameter estimate, three datasets that can be used for validation are thus available: the two reference datasets, and the identification set that wasn't used in the estimation. All 8 estimates were checked against all available validation datasets, so a total of 24 validations were done.

The validation was done by comparing measured torques with torques predicted by the model. A difficulty encountered here is the fact that acceleration measurements are not available, but are nevertheless needed for validation. Angular accelerations were thus calculated using a procedure outlined in Pham et al. (2001); Gautier (1997): the angular velocities were filtered without phase shift using a non-causal zero-phase low pass filter (done using the Matlab® filtfilt function), followed by differentiation using central differences.

Results of the validation procedure are shown in table 3.4. The top row of the table contains results obtained with both estimators using the model including friction, the bottom row contains the results without friction.

Friction (LS)			Friction (WLS)		
	ES 1	ES 2		ES 1	ES 2
ES 1	(0.0273)	0.0551	ES 1	(0.0269)	0.0556
ES 2	0.0489	(0.0247)	ES 2	0.0477	(0.0247)
VS 1	0.1444	0.0842	VS 1	0.1439	0.0862
VS 2	0.0513	0.0600	VS 2	0.0505	0.0597
Average	0.0815	0.0664	Average	0.0807	0.0672

Frictionless (LS)			Frictionless (WLS)		
	ES 1	ES 2		ES 1	ES 2
ES 1	(0.0512)	0.0784	ES 1	(0.0516)	0.0792
ES 2	0.0694	(0.0465)	ES 2	0.0687	(0.0465)
VS 1	0.1569	<u>0.0979</u>	VS 1	0.1574	0.0975
VS 2	0.0639	0.0754	VS 2	0.0627	0.0750
Average	0.0967	0.0839	Average	0.0963	0.0839

Table 3.4: Results of the validation procedure. All values are expressed in N^2m^2 .

The columns of the sub-tables indicate the dataset that was used to estimate the parameters (e.g. numbers in the columns titled “ES 1” were obtained using parameters estimated by using excitation signal 1). The rows indicate the dataset that was used for validation (with “VS 1” meaning validation signal 1 for instance). The values themselves are the averages of the mean squared torque errors for both joints. Concretely, given the mean squared torque error e_i for joint i , with e_i given by

$$e_i = \frac{1}{N} \sum_{k=1}^N (\tau_i[k] - \hat{\tau}_i[k])^2,$$

with $\tau_i[k]$ the k 'th measured torque sample, $\hat{\tau}_i[k]$ the torque prediction and N the number of samples, then the number given in the table is equal to $(e_1 + e_2)/2$ (with units N^2m^2).

As an example, let's take 0.0979 N^2m^2 , which is underlined in the table. Since it's in the column titled “ES 2”, the parameter vector θ used to obtain this entry was identified using excitation signal 2, using the least-squares estimator and the frictionless model (both of which are indicated in the sub-table's title). Since it is in the “VS 1” row, validation was done with validation signal 1, which means that 0.0979 N^2m^2 is the average of both joints' mean squared error between the torque measured while applying “VS 1” and the torque predicted using θ and the frictionless model.

The values in parentheses result from validation using the same data that was used for the estimation (a bad idea), and are only included for completeness. The

Parameter	Estimated value		Units
	Including friction	Without friction	
$m_1 L_{G_1} + m_2 L_1$	0.5238	0.5275	$kg \cdot m$
$m_1 d_{G_1}$	-0.1400	-0.1415	$kg \cdot m$
$I_{zO,1} + m_2 L_1^2$	0.0181	0.0286	$kg \cdot m^2$
$m_2 L_{G_2}$	0.3028	0.3018	$kg \cdot m$
$m_2 d_{G_2}$	0.0153	0.0175	$kg \cdot m$
$I_{zS,2}$	0.0185	0.0105	$kg \cdot m^2$
c_1	0.0302		$N \cdot m$
c_2	0.0712		$N \cdot m$
b_1	0.8294		$N \cdot m \cdot s$
b_2	0.0037		$N \cdot m \cdot s$

Table 3.5: Result of the parameter estimation procedure. The parameters estimated using the model including friction were withheld, the parameters estimated using the same excitation signal but with the model without friction are shown for comparison. See appendix B and eq. (B.6) for the definition of all parameters.

averages at the bottom of the subtables were calculated without taking them into account.

The results in the table can be used to compare the torque prediction performance of the different sets of identified parameters. They show that the model that includes friction performs better. The differences between the ordinary-least squares estimator and the weighted least-squares estimator are generally small. The lowest average of all subtables is $0.0664 N^2 m^2$, which occurs when using the least squares estimator, including friction and when performing the parameter estimation using excitation signal 2 (which was actually optimized for the frictionless model). So the parameter vector identified this way was withheld as the best estimate. Values of the individual components in θ are given in table 3.5.

Figures 3.4 and 3.5 give an impression of the torque prediction quality for the first excitation signal (which had the lowest total error for the selected parameter estimate) and the first validation signal (which had the highest error according to table 3.4), respectively.

There is a reasonable correspondence between measurement and prediction. A better match might be achieved by trying to include influences that were neglected in the model, for example the motion of the muscles with respect to the links. This would increase complexity, however, and some effects (such as the influence of the air tubes (with varying pressure) running along the links) could be very difficult to model.

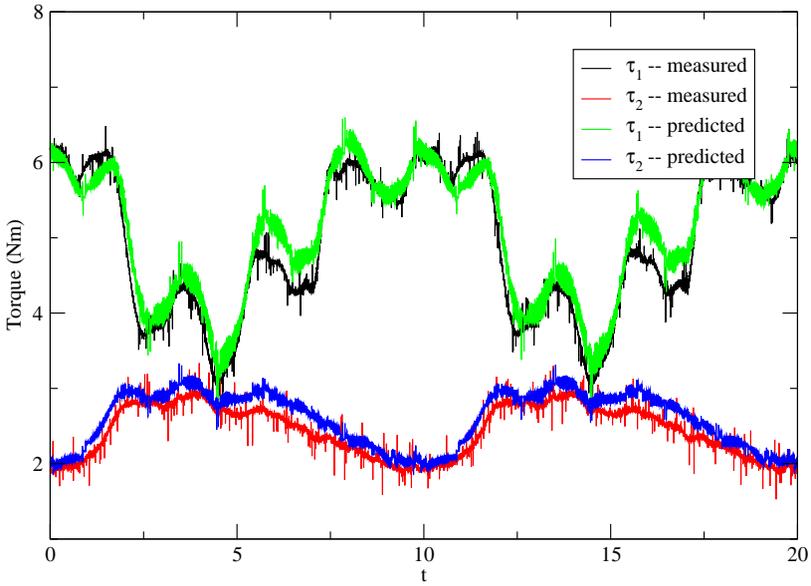


Figure 3.4: Comparison between measured and predicted torques for excitation signal 1.

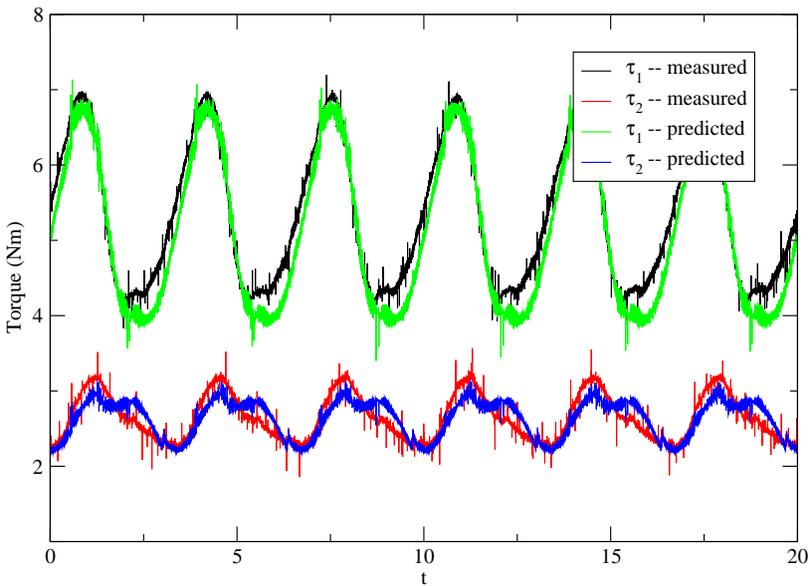


Figure 3.5: Comparison between measured and predicted torques for validation signal 1.

3.6 Hysteresis in PPAMs

Since almost all pneumatic muscle based systems use some form of feedback pressure control, these systems are always equipped with pressure sensors. Theoretically, the measured pressure could be used to estimate the force exerted by the actuator (using eq. (2.2) in the case of the PPAM). In reality, however, hysteresis is always present, which can make the estimates too inaccurate to be useful.

Chou and Hannaford (1996) report that the main cause for hysteresis in the McKibben muscle is Coulomb friction between the braided mesh shell and the internal bladder. Although the PPAM has a different working principle, it too displays hysteresis in the force-contraction characteristic (Verrelst et al., 2006a), but not as much as the McKibben muscle. This hysteresis is not modeled in (2.2).

In this section, a first step towards the inclusion of hysteresis in the PPAM's model is presented.

3.6.1 Experiments

Fig. 3.6 shows the experimentally obtained force-contraction characteristic of a PPAM with 40 fibre strands. These measurements have been performed with the muscle mounted in a tensile testing machine and the following sinusoidal force function imposed:

$$F(t) = A \sin(\omega t) + B,$$

with $A = 1400 \text{ N}$, $B = 750 \text{ N}$ and $\omega = 2\pi/500 \text{ rad/s}$. The gauge pressure was controlled to be as constant as possible by a pneumatic servo valve. One period is shown for each gauge pressure. The hysteresis is clearly visible.

Fig. 3.7 shows the dimensionless force function f_{t0} , obtained from the same data by computing the quantities $F_m(t)/p_m(t)l_0^2$ (see eq. (2.2)), and plotting them against $\epsilon_m(t)$. In these expressions, $F_m(t)$ is the measured force at time t , $p_m(t)$ the measured pressure and $\epsilon_m(t)$ the measured contraction.

For contractions above 5%, the hysteresis observed in the dimensionless force function f_{t0} does not differ significantly between the three curves shown (the difference for contractions below 5% can be explained by the elastic behavior of the muscle fibre strands, which is not taken into account in (2.2)). This indicates that the hysteresis in f_{t0} is essentially pressure independent.

Fig. 3.8 shows the measured dimensionless force function of a PPAM with 25 fibre strands. These measurements were taken while the muscle was fitted in the 2-DOF arm. While the gauge pressure was kept as constant as possible by a servo valve, the link powered by the muscle was rotated manually, and the muscle force and contraction were measured. For easy comparison, a curve from figure 3.7 (taken from a muscle with 40 fibre strands) is also shown in fig. 3.8.

Once again the different curves do not differ significantly, indicating independence

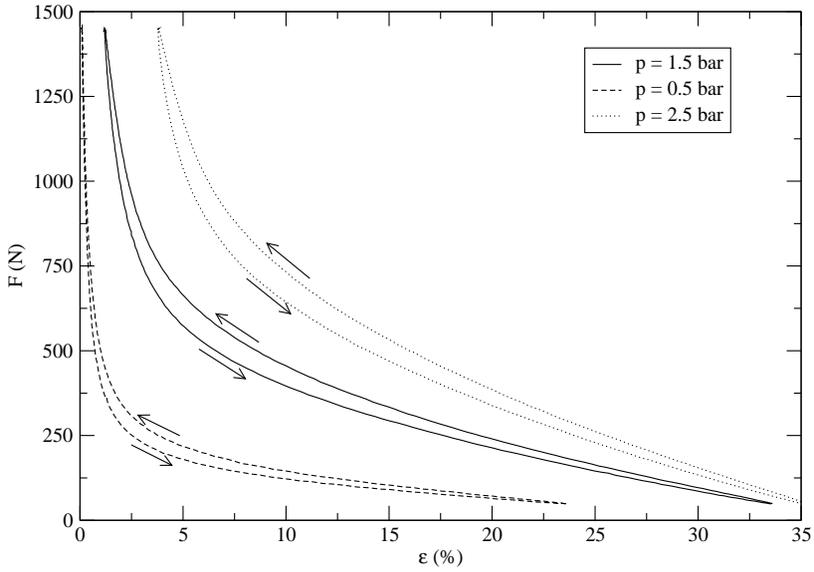


Figure 3.6: Measured force-contraction curves of a PPAM with $l_0/R \approx 6$, $l_0 = 6$ cm and $N = 40$ fibre strands for different gauge pressures.

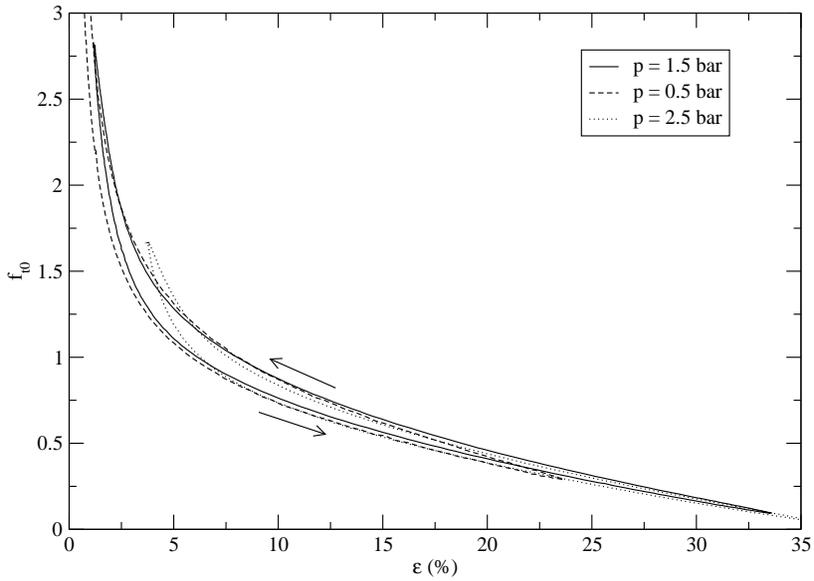


Figure 3.7: Measured dimensionless force function of a PPAM with $l_0/R \approx 6$, $l_0 = 6$ cm and $N = 40$ fibre strands for different gauge pressures.

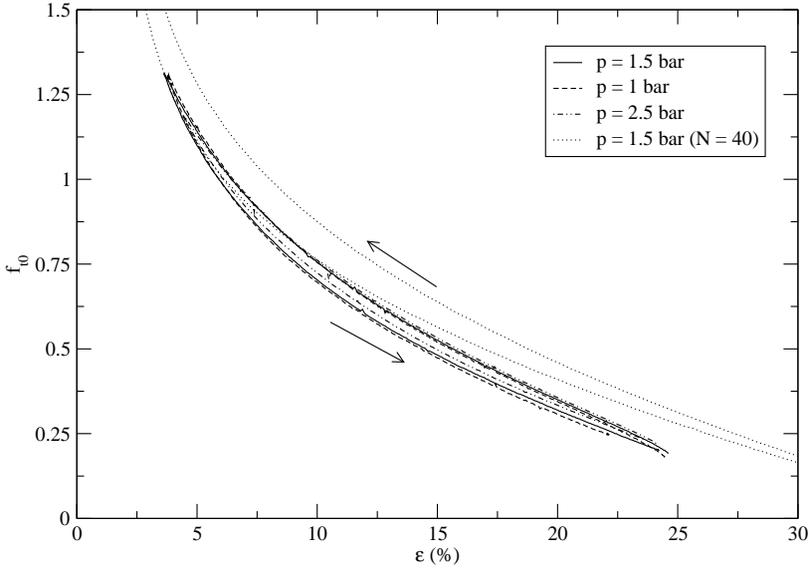


Figure 3.8: Measured dimensionless force function of a PPAM with $l_0/R \approx 6$, $l_0 = 6$ cm and $N = 25$ fibre strands for different gauge pressures. For reference, one curve measured for a similar muscle with $N = 40$ is also shown.

of gauge pressure. It is striking, however, that the hysteresis effect is much less pronounced for the case $N = 25$ than for the case $N = 40$. This indicates that friction between the Kevlar[®] fibre strands and the membrane may be an important contributing factor to the hysteresis. It could also be the unfolding of the pleats, however, or a combination of both.

Due to the lower number of fibre strands, it was expected that the dimensionless force function for $N = 25$ is lower than for $N = 40$, but the difference is too large to be consistent with the theoretical model (Verrelst et al., 2006a). Probably, there is a difference in slenderness between both muscles (the exact value of the slenderness l_0/R is hard to determine since R is difficult to measure).

Since the link rotation in the experiment was performed manually, it was impossible to achieve a constant contraction rate. Fig. 3.9 shows two measurements (with $N = 25$), one where the link was moved relatively slowly (roughly 40 seconds for the loop shown) and a second one where it was moved five times faster (around 8 seconds for the loop in the figure). The two curves hardly differ. Since the same kind of experiment but with different gauge pressures yields the same result, the hysteresis in f_{t_0} seems to be independent of the contraction rate.

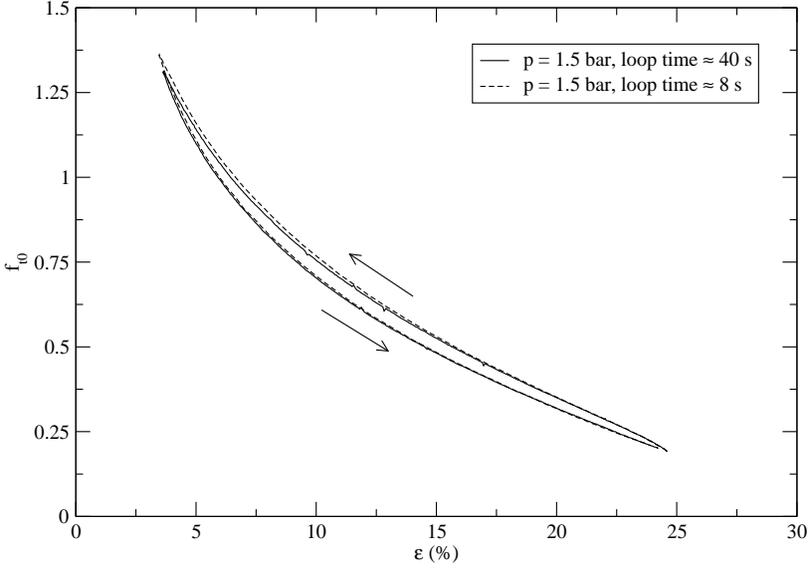


Figure 3.9: Measured dimensionless force function of a PPAM with $l_0/R \approx 6$, $l_0 = 6$ cm and $N = 25$ fibre strands for different loop speeds.

3.6.2 Modeling

The experiments suggest that the hysteresis in PPAMs can be modeled by incorporating hysteresis into the dimensionless force function f_{t0} . Since most phenomenological hysteresis models have difficulties describing hysteresis loops whose general form resembles that of f_{t0} (see fig. 2.2), we cannot model the hysteresis in f_{t0} directly.

This can be overcome by looking at the error between the observed (hysteretic) f_{t0}^{hyst} and its least-squares fit f_{t0}^{fit} of the form (2.3). Modeling the hysteretic muscle force as (cf. eq. (2.2))

$$\begin{aligned} F^{\text{hyst}} &= pl_0^2 f_{t0}^{\text{hyst}} [\epsilon] \\ &= pl_0^2 f_{t0}^{\text{fit}} (\epsilon) \cdot (1 + e [\epsilon]) \end{aligned} \quad (3.31)$$

we get

$$e [\epsilon] = \frac{f_{t0}^{\text{hyst}} [\epsilon]}{f_{t0}^{\text{fit}} (\epsilon)} - 1, \quad (3.32)$$

the relative error between the observed hysteretic dimensionless force function and its least-squares fit. In these equations, square brackets have been used to indicate

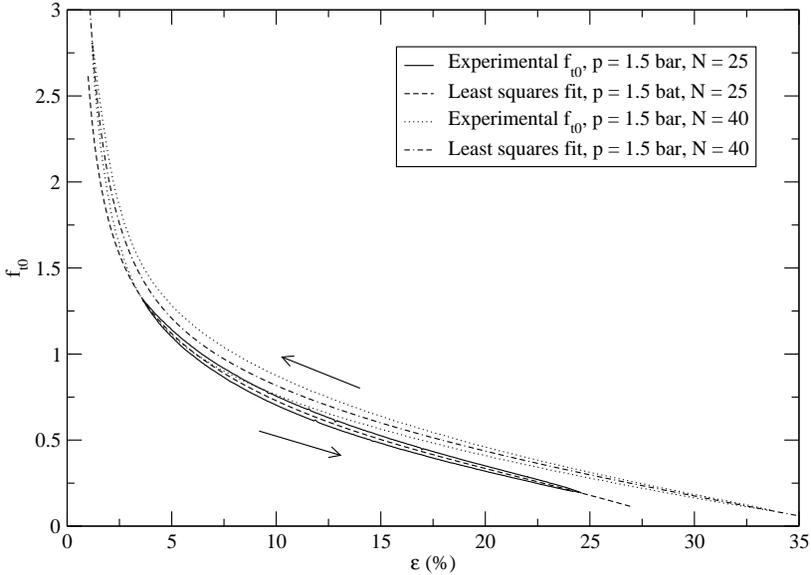


Figure 3.10: Measured and approximated dimensionless force function of PPAMs with 25 and 40 fibre strands. Both have $l_0/R \approx 6$ and $l_0 = 6$ cm.

quantities that depend hysteretically on ϵ , i.e. that depend on the current value of ϵ as well as on certain past values. Fig. 3.10 shows the measured f_{t0} for both muscles ($N = 40$ and $N = 25$, both measured at $p = 1.5$ bar), as well as their fitted approximations, and fig. 3.11 shows the relative error $e[\epsilon]$ for both cases.

The relative error $e[\epsilon]$ between the hysteretic curve and its approximation is no longer dependent on the specific shape of the dimensionless force function, which has been factored out (see $f_{t0}^{\text{fit}}(\epsilon)$ in (3.31)). This means we can model $e[\epsilon]$ by taking the scaled difference between a more conventional hysteretic loop (one that can easily be generated by a hysteresis model) and its non-hysteretic linear approximation (see fig. 3.12). It is clear that we cannot expect a perfect match, since the exact shape of $e[\epsilon]$ is not fully reproducible among experiments.

3.6.3 The Preisach model

The choice of hysteresis model is not crucial, as long as the output of the model always stays inside the major loop. We have chosen the Preisach model, since it is well studied (Mayergoyz, 1991), intuitive, invertible under mild conditions (see e.g. Brokate and Visintin (1989); Brokate (1989)), and because it has already been applied to various problems outside of its original scope (see for instance Frankowicz and Chrenowski (2006); Hu and Ben Mrad (2003); Gorbet (1997)). It

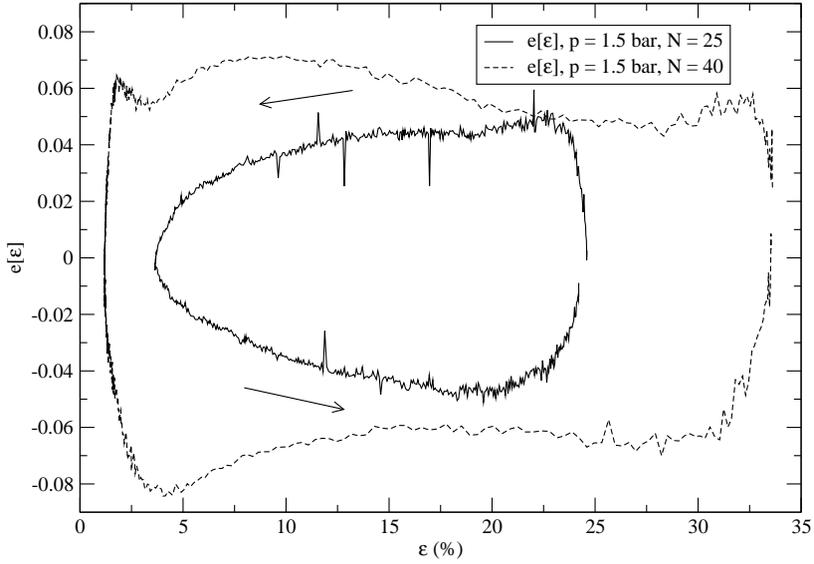


Figure 3.11: Relative error $e[\epsilon]$ between the measured and approximated dimensionless force function of PPAMs with 25 and 40 fibre strands. Both have $l_0/R \approx 6$ and $l_0 = 6$ cm.

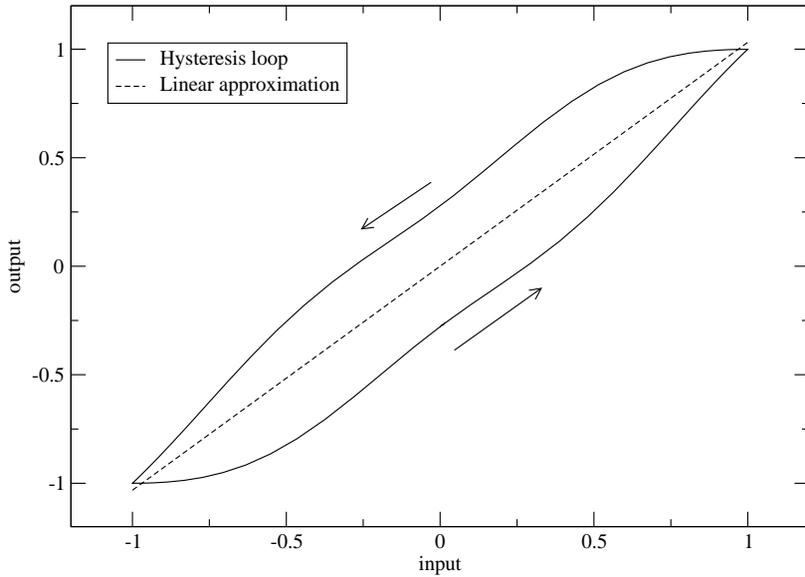
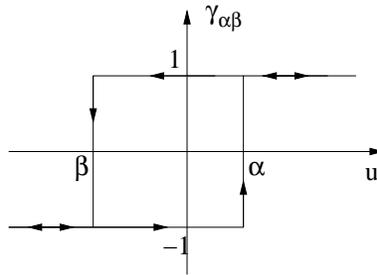
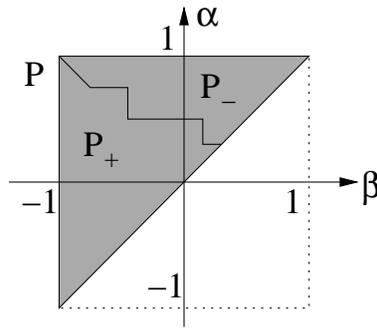


Figure 3.12: Major hysteresis loop calculated with the Preisach model, and its linear approximation.

Figure 3.13: Elementary relay $\gamma_{\alpha\beta}[u]$ Figure 3.14: Preisach plane P (in gray), divided in P_+ (relays with output $+1$) and P_- (relays with output -1) by the typical staircase line.

was introduced in the 1930's by F. Preisach to model magnetic hysteresis (Preisach, 1935), and has been called the most satisfactory mathematical model of hysteresis available (Brokate and Visintin, 1989).

The output of the Preisach model is calculated as the weighted superposition of elementary relay hysteresees $\gamma_{\alpha\beta}[u]$ (see fig. 3.13). Each relay has two switching values α and β (with $\alpha > \beta$), so the relays can be represented by points in the half-plane $\alpha > \beta$. The contribution of each relay to the output of the Preisach model is determined by a weighing function μ :

$$W[u] = \int \int_P \mu(\alpha, \beta) \cdot \gamma_{\alpha\beta}[u] d\alpha d\beta.$$

The region P of support of $\mu(\alpha, \beta)$ in the half-plane $\alpha > \beta$ is usually referred to as the Preisach plane. In this work, we will assume this region to be bound between the lines $\beta = -1$, $\beta = 1$ and $\alpha = 1$, as shown in fig. 3.14. We will also assume that $\mu(\alpha, \beta)$ is normalized in P , i.e. $\int \int_P \mu(\alpha, \beta) d\alpha d\beta = 1$.

A point with coordinates (α, β) in P represents the elementary relay $\gamma_{\alpha\beta}[u]$ with switching values α and β . This means P can be divided in two parts, P_+ and P_- ,

in which the relays $\gamma_{\alpha\beta}$ have outputs $+1$ and -1 respectively, so we have

$$W[u] = \int \int_{P_+} \mu(\alpha, \beta) d\alpha d\beta - \int \int_{P_-} \mu(\alpha, \beta) d\alpha d\beta.$$

The boundary between P_+ and P_- is a staircase line (Mayergoyz, 1991; Gorbet, 1997). To see why this is the case, consider fig. 3.15. In the top line, it is assumed that there hasn't been any input yet, and that all elementary relays are switched off. Since the output of all relays is -1 , the whole Preisach plane is called P_- . Now assume that the input is scaled, so it always remains between -1 and $+1$, and that it starts at -1 and gradually rises to a value of 0.75 . A final input value of 0.75 means that all elementary relays with $\alpha \leq 0.75$ have switched their outputs $+1$. This is visible as the grey triangle that represents P_+ on the Preisach plane. A rising input can be thought of as a horizontal line (since constant- α lines are horizontal) that moves upwards in the Preisach plane, switching on all relays it encounters. The increasing number of relays that switch on causes the output of the model to rise, as shown on the right in fig. 3.15 (second line).

As the input decreases, as seen in the third line, all relays with a value of β higher than the input switch off. Since lines of constant β in the Preisach plane are vertical, a decreasing input can be thought of as a vertical line that moves to the left in the Preisach plane, switching relays off. Since relays are switching off, the output decreases.

The increasing and decreasing of the input is thus seen to create a staircase-like separation between P_+ and P_- in the Preisach plane. The “memory” of the Preisach model is encoded in the shape of this line.

The major hysteresis loop shown in fig. 3.12 is the output of the Preisach model to one period of a sine function as input and with the normalization $\tilde{\mu}(\alpha, \beta)$ of

$$\mu(\alpha, \beta) = \begin{cases} e^{-(\beta-\alpha-c)^2 - (\beta+\alpha-d)^2} & \alpha + \beta \leq 0 \\ e^{-(\beta-\alpha-c)^2 - (\beta+\alpha+d)^2} & \alpha + \beta > 0 \end{cases} \quad (3.33)$$

as weight function, where $c = -0.1$ and $d = -1$ (Zsolt, 2002).

3.6.4 Application to the PPAM

Using the above described Preisach model, we now write $e[\epsilon]$ from eq. (3.31) as

$$e[\epsilon] = \delta \cdot \left(W[\epsilon_s] - W^{\text{fit}}(\epsilon_s) \right),$$

with

$$-1 \leq \epsilon_s = 2 \frac{\epsilon}{\epsilon_{\max}} - 1 \leq 1. \quad (3.34)$$

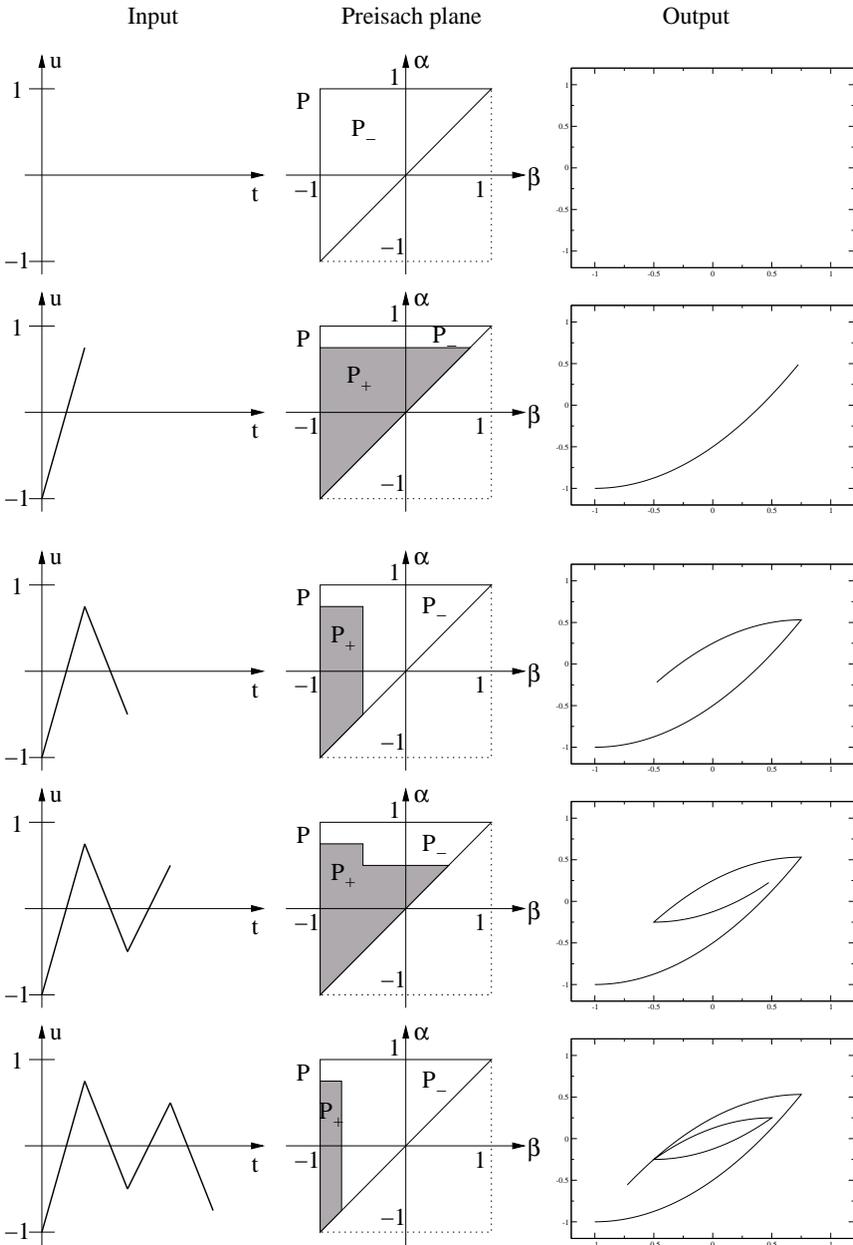


Figure 3.15: Input, Preisach plane and output of a Preisach hysteresis model with $\mu(\alpha, \beta) = 1$. It is assumed that initially all elementary relays are switched off (i.e. have output -1).

δ is a scaling factor. $W[\epsilon_s]$ is the output of the Preisach model and $W^{\text{fit}}(\epsilon_s)$ is the linear approximation obtained by fitting the two major loops generated by $W[\epsilon_s]$. Both $W[\epsilon_s]$ and $W^{\text{fit}}(\epsilon_s)$ are shown in fig. 3.12 for weight function $\tilde{\mu}$ (see (3.33)).

Definition (3.34) of ϵ_s assumes the major hysteresis loop of f_{t0} to be between $\epsilon = 0$ and $\epsilon = \epsilon_{\text{max}}$, which are mapped to -1 and 1 respectively, the minimum and maximum input values of the considered Preisach model. In what follows, the value of ϵ_{max} was chosen to be 35%.

The hysteresis model of the PPAM muscle thus becomes

$$\begin{aligned} F^{\text{hyst}} &= p l_0^2 f_{t0}^{\text{hyst}}[\epsilon] \\ &= p l_0^2 f_{t0}^{\text{fit}}(\epsilon) \cdot \left(1 + \delta \cdot \left(W[\epsilon_s] - W^{\text{fit}}(\epsilon_s) \right) \right). \end{aligned} \quad (3.35)$$

3.6.5 Results

Fig. 3.16 shows the result of the model for the muscle with 40 fibre strands using $\tilde{\mu}$ (cf. (3.33)). The fit f_{t0}^{fit} was obtained from the measurements taken at $p = 1.5$ bar (also shown in the figure), and the optimal value of δ was determined with the least-squares method ($\delta \approx 0.299$). The correspondence between model and measurement is good for $7\% \leq \epsilon \leq 20\%$, the most important region in applications. This result is slightly misleading, however, since the muscle isn't usually used at constant pressure.

In the following test (performed with data taken from the muscle with 25 fibre strands), we started out the same way: f_{t0} was fitted from the data taken at 1.5 bar, and an optimal value for δ was estimated from this data ($\delta \approx 0.139$). Next, f_{t0}^{fit} was calculated again, this time from all available data (taken at gauge pressures of 1 bar, 1.5 bar and 2.5 bar), while δ wasn't changed. Finally, the muscle's output force was calculated from the hysteresis model (3.35) for the three cases. The result is shown in fig. 3.17. Again, the most important deviations are situated in the low ϵ ranges. This is due to inaccuracies both in the model and in f_{t0}^{fit} , which isn't perfectly tuned for all gauge pressures. It is important to remark that the forces shown in fig. 3.17 were calculated using measured pressure values, the pressures were not assumed to be exactly equal to their desired values. Slight pressure deviations (which always occur because of disturbances and imperfections in the pressure regulating valves) can generate non-negligible force differences in PPAMs, so they have to be taken into account.

3.6.6 Conclusion

There is a good agreement between the hysteresis model and the experiments for the contraction range $7\% \leq \epsilon \leq 20\%$ (which is the range mostly used in applications).

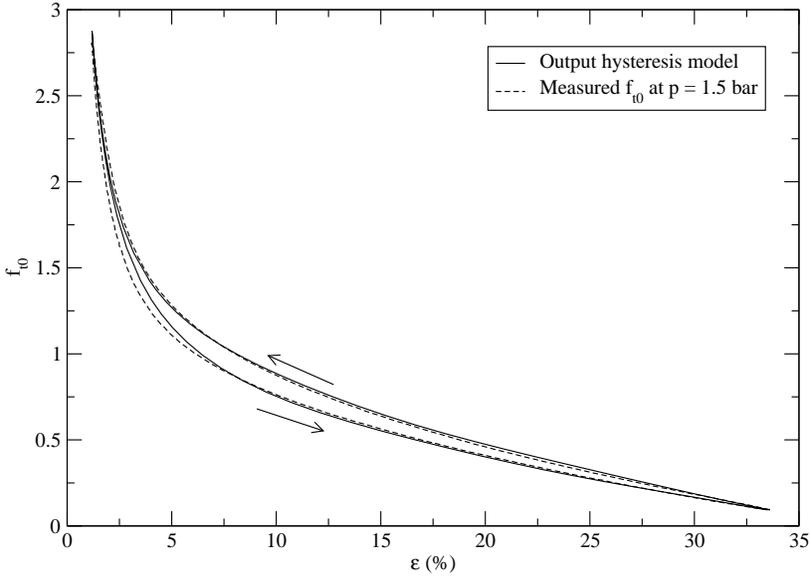


Figure 3.16: The dimensionless force function $f_{t0}^{\text{hyst}}[\epsilon]$ as calculated by the hysteresis model. The experimentally determined values for $p = 1.5$ bar are also shown. Muscle parameters were $l_0/R \approx 6$, $l_0 = 6$ cm and $N = 40$.

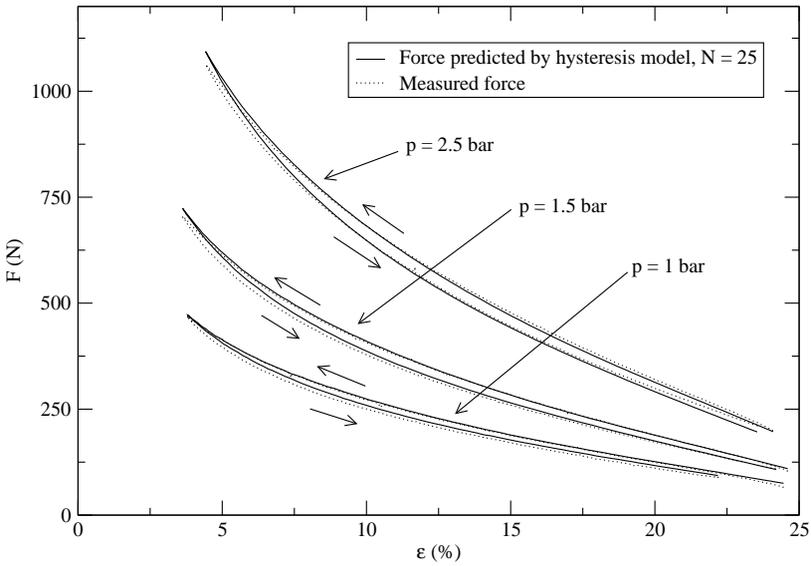


Figure 3.17: Muscle force as calculated by the hysteresis model, as well as measured values. Muscle parameters were $l_0/R \approx 6$, $l_0 = 6$ cm and $N = 25$.

This range may be expanded by numerically determining a better Preisach weight function μ from experimental data.

The model is rather cumbersome to work with in practice, however, since at startup the hysteretic state of the muscle is unknown. In order for the model to match the output of the muscle, the muscle has to be brought to one of its extremes in contraction. For this reason, the model will not be used in the rest of this work.

3.7 Summary

This chapter has introduced a rigid body model for the mechanical structure of the arm, and an approximate model for the pressure dynamics in the muscle-valve system.

Since gravity compensation will be necessary for controlling the manipulator, the parameters of the system were first estimated in static conditions. The estimates enabled a usable gravity compensation, although hysteresis in the muscle's force-contraction characteristic was seen to cause significant error.

The parameters of the mechanical model were also estimated in dynamic conditions, using a filtering technique to eliminate the need for angular accelerations. The excitation signals were calculated by numerical optimization. Reasonable results were obtained, but it is clear that parts of the dynamics are not included in the model.

In the last section, a hysteretic model of the PPAM, based on the Preisach model for hysteresis, was introduced. Good agreement between predictions and measurements was achieved in an important range of contractions.

The results from this chapter illustrate the general difficulty of obtaining accurate models for systems actuated by pneumatic muscles. For this reason experimental validation of control results remains very important.

Part II

Control

Introduction to Part II

Part II of this text is about controlling the pneumatic manipulator. Two controllers will be considered in detail, sliding mode control and proxy-based sliding mode control (which are quite different, in spite of the similar names). Before starting with sliding mode control in chapter 4, a very short overview of previous work is given, and the challenges faced when controlling a pneumatic muscle system are listed.

Previous work

After Schulte (1961) published the first thorough analysis of the properties of the McKibben muscle, the field of pneumatic muscle research lay dormant for almost thirty years, probably due to the difficulty to control the muscle. The paper by Inoue (1987) (which introduced the Δp -approach) revived interest in pneumatic muscle actuation, and since that time many publications about pneumatic muscle modeling and control have appeared, mainly focused on the McKibben muscle. In this section, a brief overview of previous work in pneumatic muscle control will be given.

Position and tracking control Most of the research in pneumatic muscle control has been focused on position and tracking control. When the systems involves antagonistically placed muscles, these controllers invariably use some form of the Δp -approach (see section 3.3.1) to make the number of controller outputs equal to the number of degrees of freedom.

Conventional PID position control has often been used for pneumatic muscle systems (Caldwell et al., 1993; Tondu et al., 1994; Caldwell et al., 2001; Caldwell and Tsagarakis, 2002; Šitum and Herceg, 2008), sometimes complemented with a feedforward term in the controller. Modified versions of PID control have been

reported in Schröder et al. (2003); Thanh and Ahn (2006b,a).

The fact that the muscle parameters are usually not very well known has led several authors to propose adaptive controllers, as in Nouri et al. (1994); Medrano-Cerda et al. (1995); Caldwell et al. (1995). Lilly (2003); Zhang et al. (2007) also propose adaptive controllers for pneumatic muscle systems, but their work is purely simulation based.

The difficulty in modeling systems actuated by pneumatic muscles has led to a lot of controllers that use soft computing methods (i.e. neural networks, fuzzy logic, evolutionary algorithms, ...), as reported in Hesselroth et al. (1994); van der Smagt et al. (1996); Eskiizmirli et al. (2001); Carbonell et al. (2001a); Balasubramanian and Rattan (2003b); Chang and Lilly (2003); Balasubramanian and Rattan (2005); Chang et al. (2006); Yamazaki and Yasunobu (2007).

Because of its robustness, sliding mode control has received a lot of attention as well, mostly in simulation based studies (Sira-Ramírez et al., 1996; Cai and Yamaura, 1996; Repperger et al., 1998; Cai and Dai, 2000; Carbonell et al., 2001b; Cai and Dai, 2003; Lilly and Quesada, 2004; Lilly and Yang, 2005; Yang, 2006), but experimental work has been reported as well (Nouri et al., 1994; Hamerlain, 1995; Tondu and Lopez, 2000; Chettouh et al., 2006, 2008a,b).

Computed torque or inverse dynamics control was used in Verrelst et al. (2005); Hildebrandt et al. (2005); Vanderborght et al. (2006a); Verrelst et al. (2006b); Vanderborght et al. (2008b,a).

Various other forms of nonlinear control have also been proposed, see Kimura et al. (1997); Carbonell et al. (2001b); Hildebrandt et al. (2002); Aschemann and Hofer (2006); Schindele and Aschemann (2008).

Force or torque control For rehabilitation purposes, PID-based torque control using joint torque sensors has been used in Tsagarakis and Caldwell (2003); Caldwell and Tsagarakis (2002); Costa and Caldwell (2006).

Sardellitti et al. (2007) proposes torque control based on muscle force measurements, while Schröder et al. (2003) presents a controller with inner torque loop that's entirely model based, i.e. it doesn't use force or torque sensors.

Noritsugu and Tanaka (1997) have presented an impedance controller using a force sensor in the tool-center point.

Compliance control Since in an antagonistic setup two muscles control a single degree of freedom, it is possible to control compliance (the inverse of stiffness) in addition to position or torque. Simultaneous position and compliance control is presented in Clapa et al. (2006); Vanderborght et al. (2008b), and adaptive simultaneous position and compliance control in Tonietti and Bicchi (2002).

Open loop control As already noted by Inoue (1987), pneumatic muscle systems can be controlled without feedback, using only model-based feedforward control. This idea has been applied in Balasubramanian and Rattan (2003a); Sugar et al. (2007).

Challenges

In general, controlling pneumatic muscle manipulators is not straightforward. Difficulties encountered when designing a controller for a system actuated by pleated pneumatic artificial muscles include the following:

- The non-linear force-contraction relation of the PPAM actuator.

The force output of a pneumatic muscle isn't only a function of gauge pressure (the control input), but also depends on the joint angle (which is part of the system state) in a nonlinear way (see eq. (2.2)).
- Compliance of the system.

Modeling and control of robots with flexible joints of finite and constant stiffness is a well established topic (see for instance Albu-Schäffer et al. (2007); De Luca (2000); Tomei (1991); Spong (1987)). Joint stiffness in a joint actuated by pneumatic muscles is a (usually nonlinear) function of gauge pressures and joint angle, however, so techniques that assume constant joint stiffness are not applicable.
- Hysteresis in the force-contraction relation of the PPAM.

Although one of the PPAM's design goals was to have less hysteresis than other types of pneumatic artificial muscles, it is still present (hysteresis was studied in section 3.6). This makes it much more difficult to accurately model the muscle, and hence to predict it's force output.
- Imprecise knowledge of PPAM parameters.

It is difficult to produce PPAM's with a well specified value for the slenderness (l_0/R), and it is difficult to measure the slenderness. Hence, the muscle parameters are usually only approximatively known.
- Inaccurately known dynamics of the pressure regulating valves.

From the viewpoint of the controller, the pressure servo valves are essentially black boxes with inaccurately known (and nonlinear) dynamics.
- Long pressure settling times.

Actuator gauge pressures can take a relatively long time to settle, over 100 ms for large pressure steps (and strongly dependent on the valve characteristics).

When the controller commands new actuator pressures, it takes a long time (compared to the sampling period) before the desired pressure is actually present. How long it takes exactly can depend on a lot of different factors, such as how large the pressure difference is, how high the supply pressure is, if the muscle's volume is changing, etc.

- The coupling between actuator gauge pressures and link angles and angular velocities.

Eq. (3.6) shows that there is a complex interaction between gauge pressure, joint angle (through the muscle volume) and angular velocity. This means that the system cannot be modeled as a cascade of a pneumatic system followed by a mechanical system.

An accurate tracking controller has to overcome these difficulties. In this work, tracking accuracy is not the primary concern, however, safety is more important.

Chapter 4

Sliding Mode Control

4.1 Introduction

Sliding mode control is known for its robustness, an appealing quality when controlling a system that is difficult to model. Encouraging results about applying sliding mode control to pneumatic muscle systems have repeatedly been reported in literature. Since it also has interesting safety features (see section 4.2.2), it was decided to test sliding mode control for the 2-DOF pneumatic manipulator.

Initial simulations and experiments revealed that implementing sliding mode control is much harder than for instance Lilly and Yang (2005); Lilly and Quesada (2004); Cai and Dai (2003) would lead us to believe. The reason is that these papers consider only computer simulation, without including pressure dynamics. The interaction between controller, servo-valve and the varying volume in the muscle is thus neglected and considered to be parasitic actuator dynamics. While this assumption is certainly justifiable in certain cases, we believe it cannot be used in the case of sliding mode control, where fast switching of the control input is inherently present. Thus, it is thought to be important that the (slow) pressure dynamics (or actuator dynamics) are considered in the control design.

Section 4.3 in this chapter describes the design of a sliding mode controller that accounts for the pressure dynamics in the PPAMs. Since the derivation of the controller is rather complicated and technical, most of the concepts used in section 4.3 are explained in section 4.2. This section introduces sliding mode control by means of an example, and details why using sliding mode control might be advantageous for safety. It also shows that accounting for actuator dynamics is necessary, and that this can be done using feedback linearization.

Section 4.4 discusses some results obtained by simulation and experimentation.

4.2 Sliding mode control

A complete introduction to sliding mode control is outside the scope of this text. In this section, we will illustrate some of the most important concepts of sliding mode control by means of an example. For more information, please refer to for instance Utkin (1977); Slotine and Li (1991); Hung et al. (1993); Khalil (2002); Perruquetti and Barbot (2002).

4.2.1 Ideal sliding mode control

As an example, suppose we want to design a sliding mode tracking controller for the one-dimensional nonlinear mass-spring model

$$m\ddot{z} = -k_1z - k_2z^3 + u, \quad (4.1)$$

where u (an applied force) is the control input. If we set $x_1 = z$ and $x_2 = \dot{z}$ we can write (4.1) in state-space form:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{k_1}{m}x_1 - \frac{k_2}{m}x_1^3 + \frac{1}{m}u, \end{aligned}$$

or equivalently

$$\dot{x}_1 = x_2 \quad (4.2)$$

$$\dot{x}_2 = b(x_1) + a \cdot u, \quad (4.3)$$

with $b(x_1) = -\frac{k_1}{m}x_1 - \frac{k_2}{m}x_1^3$ and $a = 1/m$.

If we write $x_{1d} = z_d$ for the desired position and $x_{2d} = \dot{z}_d$ for the desired velocity, we can define the position and velocity errors e_1 and e_2 as

$$e_1 = x_{1d} - x_1 \quad (4.4)$$

$$e_2 = x_{2d} - x_2. \quad (4.5)$$

The goal of a sliding mode controller is to keep the system state $\mathbf{x} = [x_1 \ x_2]^T$ on the so-called sliding surface, a (time-varying) line or (hyper)surface in state space that represents desired dynamics for the system. For a second order system such as the example considered here, it is commonly defined as $s(\mathbf{x}, t) = 0$, with

$$s(\mathbf{x}, t) = e_1 + \lambda e_2 \quad (4.6)$$

and $\lambda > 0$ a constant. If the state remains on the sliding surface (i.e. if $s(\mathbf{x}, t)$ can be kept equal to zero), it determines the dynamics of the system. With $s(\mathbf{x}, t)$

defined as above, $s(\mathbf{x}, t) = 0$ implies

$$\dot{e}_1 = -\frac{1}{\lambda}e_1$$

(since $e_2 = \dot{e}_1$), or

$$e_1 = e_{1,0}e^{-\frac{1}{\lambda}t},$$

with $e_{1,0}$ the position error e_1 at $t = 0$. We see that if the system state remains on the sliding surface, the position error exponentially decays to zero. The same holds for the velocity error, since if the value of $s(\mathbf{x}, t)$ remains constant (zero), we have $\dot{s}(\mathbf{x}, t) = 0$, or (using $\dot{e}_1 = e_2$ in eq. (4.6)) $e_2 + \lambda e_2 = 0$, so we get

$$\dot{e}_2 = -\frac{1}{\lambda}e_2.$$

The possibility to have the states exponentially converge to their desired values with a chosen time constant λ makes sliding mode control interesting in view of safe control (see section 4.2.2).

Once we have chosen the sliding surface, we still have to design a controller that makes it invariant (i.e. once on the sliding surface, the state will stay on it) and attractive (if the state is not on the sliding surface it will move towards it). We'll do this in two steps. First, we assume the state is on the sliding surface (i.e. $s(\mathbf{x}, t) = 0$), and ask ourselves what control action would be necessary to keep it there. If $s(\mathbf{x}, t)$ has to remain zero, we have $\dot{s}(\mathbf{x}, t) = 0$. Using (4.4) and (4.5), $\dot{s}(\mathbf{x}, t)$ becomes

$$\begin{aligned}\dot{s}(\mathbf{x}, t) &= \dot{e}_1 + \lambda \dot{e}_2 \\ &= e_2 + \lambda(\dot{x}_{2d} - \dot{x}_2),\end{aligned}$$

or using eq. (4.3)

$$\dot{s}(\mathbf{x}, t) = e_2 + \lambda(\dot{x}_{2d} - b(x_1) - a \cdot u) \quad (4.7)$$

(the desired acceleration $\dot{x}_{2d} = \ddot{z}_d$ is supposed to be known). Setting $\dot{s}(\mathbf{x}, t) = 0$ in the above equation gives us the control input u_{eq} necessary to keep the state on the sliding surface once it is on it:

$$u_{eq} = \frac{1}{a} \left(\dot{x}_{2d} - b(x_1) + \frac{e_2}{\lambda} \right).$$

This is called the equivalent control.

In order to make the sliding surface attractive, we add a discontinuous term to the above control law, which gives us the sliding mode controller:

$$u = \frac{1}{a} \left(\dot{x}_{2d} - b(x_1) + \frac{e_2}{\lambda} + K \operatorname{sgn} s \right), \quad (4.8)$$

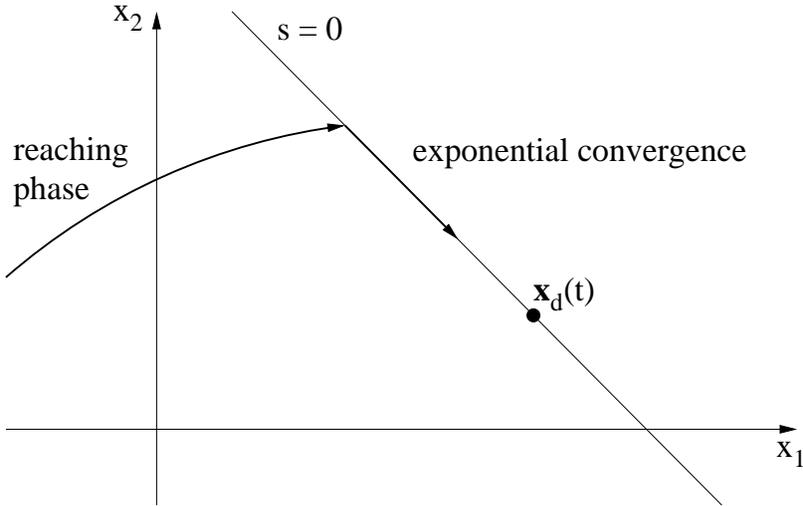


Figure 4.1: State-space trajectory when using sliding mode control (shown for a second order system).

where $K > 0$ is a constant. If we substitute (4.8) in (4.7), we get

$$\dot{s} = -\lambda K \operatorname{sgn} s.$$

Given that $\lambda > 0$ and $K > 0$, this means that if $s > 0$, we have $\dot{s} < 0$, and if $s < 0$ we have $\dot{s} > 0$. This can be summarized as

$$\dot{s} \cdot s < 0.$$

If $s \neq 0$, the controller thus ensures that s moves towards 0. Once s is equal to zero, it remains zero.

The system behavior when under sliding mode control is illustrated in fig. 4.1. The sliding surface is a line of slope $-1/\lambda$ in state space that contains the (time varying) desired state of the system, the point $\mathbf{x}_d(t) = [x_{1d} \ x_{2d}]^T = [z_d \ \dot{z}_d]^T$. Starting from any initial condition, the state trajectory reaches the sliding surface in finite time during the so-called reaching phase, and then slides along the surface while it converges exponentially to the desired state \mathbf{x}_d .

4.2.2 Safety

Fig. 4.2 shows simulations of three different step responses of the nonlinear mass-spring system when using the above designed sliding mode controller¹. It is clear

¹For the simulation, the following parameter values were chosen: $m = 2 \text{ kg}$, $k_1 = 1 \text{ N/m}$, $k_2 = 0.1 \text{ N/m}^3$ and $K = 5 \text{ m/s}^2$.

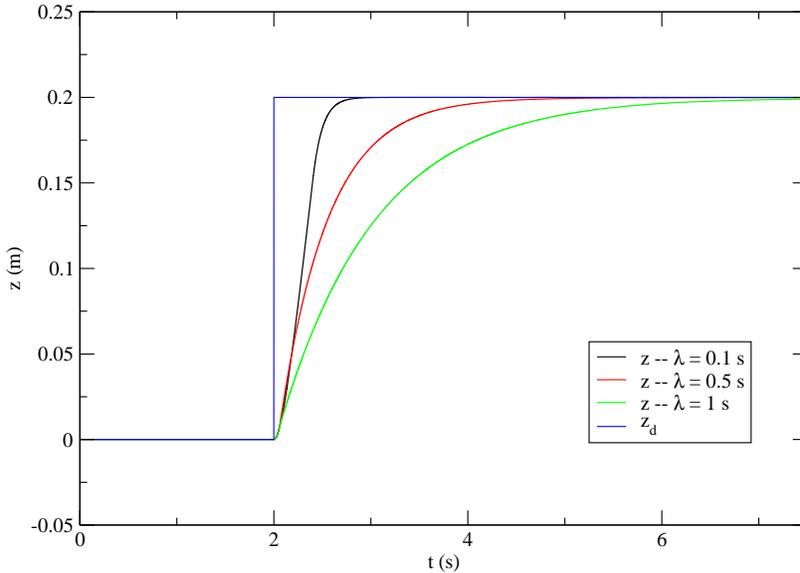


Figure 4.2: Simulated step response of the nonlinear mass-spring system when using controller (4.8) for different values of the time constant λ .

that by increasing the time constant λ , the response can easily be made slower, and thus safer.

An increase in λ has two effects: it slows down the exponential convergence once the sliding surface is reached, but it also causes the sliding surface to be reached at lower velocities (in absolute value). This can be seen in fig. 4.3. An increase in λ makes the sliding surface in state space less steep (since its slope is $-1/\lambda$), so for the same initial conditions the state will reach it at a lower velocity.

Since increasing λ doesn't deteriorate tracking performance in steady state (see fig. 4.4), sliding mode control could be a good candidate for a safe (or safer) controller. Safety will be discussed in more detail in chapter 6.

4.2.3 Robustness

Sliding mode control law (4.8) assumes a perfect knowledge of the system to be controlled. In reality, a and $b(x_1)$ are never known exactly, only their estimates $\hat{a} > 0$ and $\hat{b}(x_1)$ are available. In implementations, eq. (4.8) is calculated using these estimates,

$$u = \frac{1}{\hat{a}} \left(\dot{x}_{2d} - \hat{b}(x_1) + \frac{e_2}{\lambda} + K \operatorname{sgn} s \right). \quad (4.9)$$

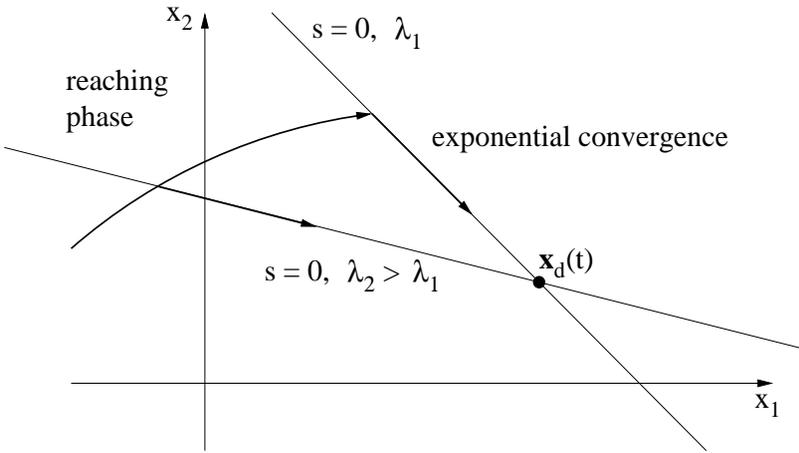


Figure 4.3: Higher values of λ imply that the sliding surface will be reached at lower velocities.

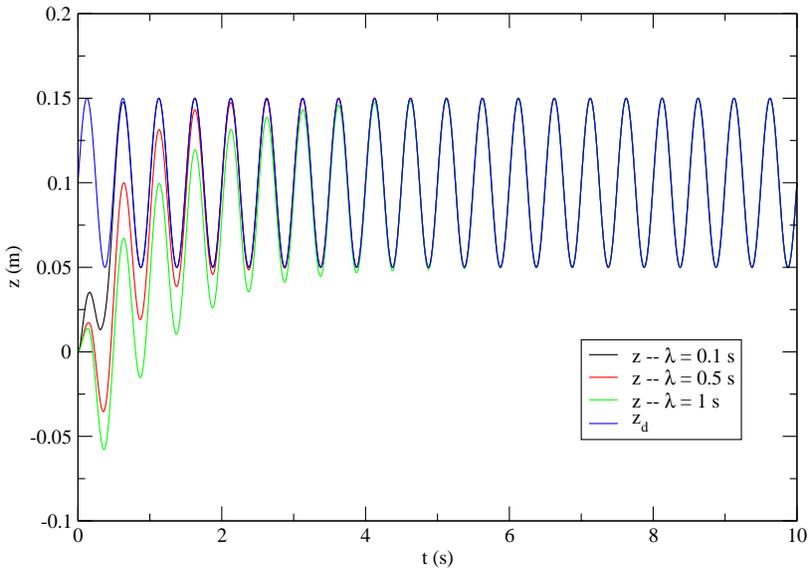


Figure 4.4: Response of the sliding mode controlled nonlinear mass-spring system when tracking a sinusoidal trajectory for different values of the time constant λ . Initial conditions were $z = \dot{z} = 0$.

By substitution of (4.9) in (4.7), we get

$$\begin{aligned}\dot{s}(\mathbf{x}, t) &= e_2 + \lambda \left(\dot{x}_{2d} - b(x_1) - \frac{a}{\hat{a}} \left(\dot{x}_{2d} - \hat{b}(x_1) + \frac{e_2}{\lambda} + K \operatorname{sgn} s \right) \right) \\ &= e_2 + \lambda \left(\dot{x}_{2d} - b(x_1) - \frac{a}{\hat{a}} \dot{x}_{2d} + \frac{a}{\hat{a}} \hat{b}(x_1) - \frac{a}{\hat{a}} \frac{e_2}{\lambda} - \frac{a}{\hat{a}} K \operatorname{sgn} s \right) \\ &= \Delta e_2 + \lambda \Delta \dot{x}_{2d} + \lambda \Delta b(x_1) - \lambda \frac{a}{\hat{a}} K \operatorname{sgn} s,\end{aligned}$$

with $\Delta e_2 = e_2 - \frac{a}{\hat{a}} e_2 = e_2 \left(1 - \frac{a}{\hat{a}}\right)$, $\Delta \dot{x}_{2d} = \dot{x}_{2d} \left(1 - \frac{a}{\hat{a}}\right)$ and $\Delta b(x_1) = \frac{a}{\hat{a}} \hat{b}(x_1) - b(x_1)$. Since a , \hat{a} , λ and K are all positive, we see that if

$$\lambda \frac{a}{\hat{a}} K > \Delta e_2 + \lambda \Delta \dot{x}_{2d} + \lambda \Delta b(x_1)$$

or

$$K > \frac{\hat{a}}{a} \left(\frac{\Delta e_2}{\lambda} + \Delta \dot{x}_{2d} + \Delta b(x_1) \right)$$

\dot{s} will always have the opposite sign of s , and the sliding surface will be attractive. If the gain K is chosen high enough to “overpower” the uncertainties in the system’s parameters, the controller will still behave as desired. Sliding mode control is thus said to be robust with respect to parametric uncertainty in the system model, which is traditionally considered to be its main advantage (Slotine and Li, 1991).

4.2.4 Chattering

Sliding mode control is discontinuous across the sliding surface. The implementation of the control switching is always imperfect: the actuator switching is never instantaneous, there are sensor and calculation delays, the value of s is not known with infinite precision, etc. This causes chattering (Slotine and Li, 1991), as illustrated in fig. 4.5. When the system’s state trajectory reaches the sliding surface (where ideally it should stay and start sliding), the delay in control switching causes it to cross over to the other side. When the control switches, it reverses direction, heads for the sliding surface but overshoots again, etc.

Chattering is highly undesirable, resulting in low control accuracy, high heat losses in electrical circuits and high wear in mechanical parts. The fast switching can also excite high-frequency unmodeled dynamics.

The most common way to eliminate chattering (see for instance Slotine and Li (1991)) is to introduce a boundary layer by replacing the sign function $\operatorname{sgn} s$ in (4.9) by a high slope saturation function $\operatorname{sat}(s/\Gamma)$, with

$$\operatorname{sat}(y) = \begin{cases} y & \text{if } |y| \leq 1 \\ \operatorname{sgn}(y) & \text{if } |y| > 1 \end{cases} \quad (4.10)$$

and Γ a positive constant. The function $\operatorname{sat}(s/\Gamma)$ is shown in fig. 4.6.

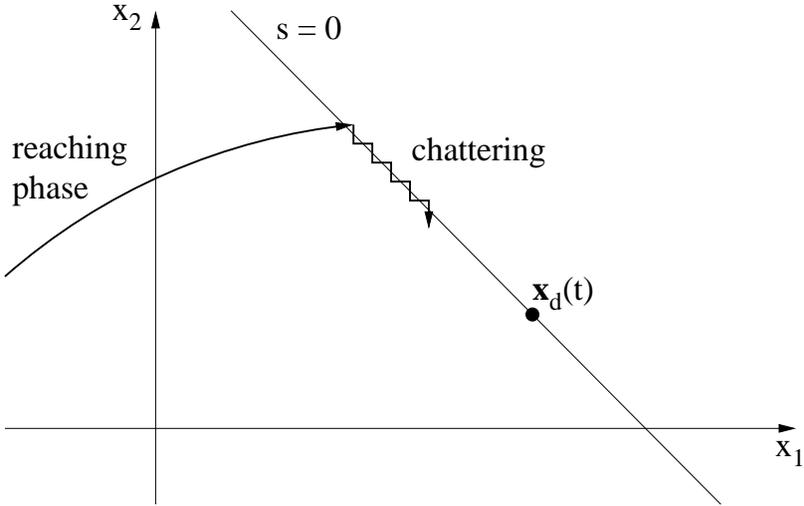


Figure 4.5: Chattering due to delay in control switching.

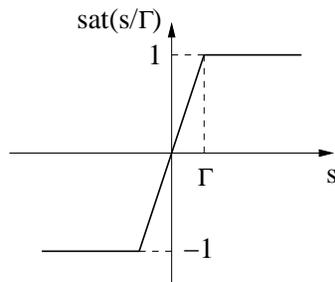


Figure 4.6: Saturation function $\text{sat}(s/\Gamma)$.

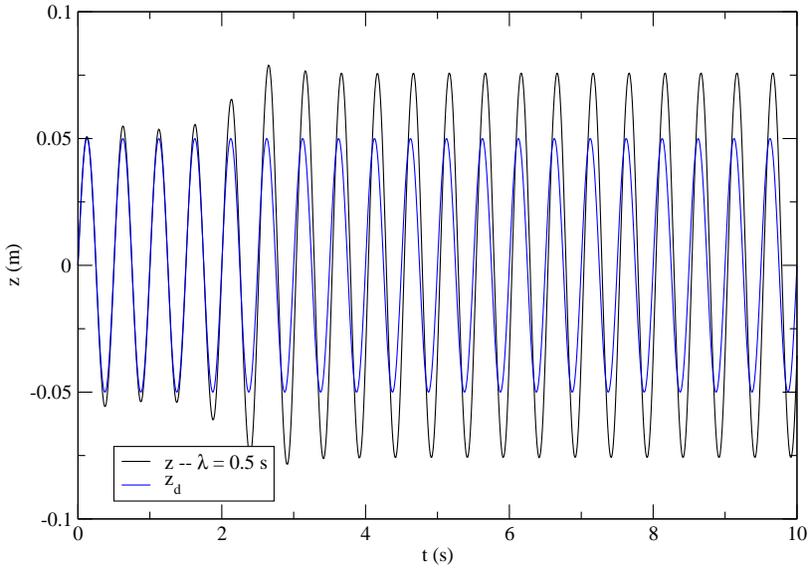


Figure 4.7: Simulated response of the sliding mode controlled nonlinear mass-spring system when tracking a sinusoidal trajectory (of frequency 2 Hz) in the presence of unmodeled (slow) first order actuator dynamics with time constant 0.06 s. The initial state was chosen to coincide with the desired initial state (i.e. the mass starts at the desired position and with the desired velocity).

4.2.5 Actuator dynamics

If the actuator dynamics is fast with respect to the system dynamics, it can be neglected (i.e. left unmodeled) and its effects countered using a boundary layer. If the actuator dynamics can't be considered fundamentally faster than the rest of the system, however, it has to be modeled. This is illustrated in fig. 4.7. The simulation in fig. 4.7 included unmodeled first order actuator dynamics with a time constant of 0.06 s. No boundary layer was used. By comparing with fig. 4.4, it is clear that this hurts tracking performance.

This example illustrates that in order to use sliding mode control for pneumatic muscle systems (which have slow actuator dynamics), a model of the actuators has to be taken into account when designing the controller.

4.2.6 Feedback linearization

Control law (4.8) is of the form

$$u = \frac{1}{a} (-b(x_1) + v). \quad (4.11)$$

If we substitute (4.11) in the eq. (4.3), the governing equations (4.2) and (4.3) become

$$\begin{aligned}\dot{x}_1 &= x_2 \\ \dot{x}_2 &= v.\end{aligned}$$

We see that the nonlinear term $b(x_1)$ was cancelled out, and the resulting system is linear, with input v . u can be thought of as an inner loop control, linearizing the system, while v is the outer loop controller that controls the linearized system. The new control input v could be designed in many ways (using linear control techniques, for example), with v chosen to be

$$v = \dot{x}_{2d} + \frac{e_2}{\lambda} + K \operatorname{sgn} s$$

in the case of the sliding mode controller discussed above. Since control law (4.11) linearizes the system by means of feedback, this technique is called feedback linearization².

The idea of feedback linearization by cancelling out nonlinearities can be simply applied to systems in the so-called controllability canonical form³. One of its properties is a state vector that contains the system's output and its derivatives up to order n , with n the order of the system. With state vector $\mathbf{x} = [x_1 \ x_2]^T = [z \ \dot{z}]^T$, system (4.2)-(4.3) clearly has this property, which was repeatedly used in the design of the sliding mode controller (for instance when stating that $e_2 = \dot{e}_1$).

If we include first order actuator dynamics with time constant T (similar to eq. (3.2)) in the model, (4.2)-(4.3) becomes

$$\dot{x}_1 = x_2 \tag{4.12}$$

$$\dot{x}_2 = b(x_1) + a \cdot u \tag{4.13}$$

$$\dot{u} = -\frac{u}{T} + \frac{u_d}{T}, \tag{4.14}$$

with u_d (the desired actuator force) the input of the system. It is clear from the above equations that we cannot simply choose u_d to cancel the nonlinear term

²Computed torque or inverse dynamics control (see for instance Spong et al. (2006)) is one of the most common applications of feedback linearization in robotics.

³A SISO system is said to be in controllability canonical form (see for instance Slotine and Li (1991), chapter 6) if its dynamics can be written as

$$x^{(n)} = f(\mathbf{x}) + b(\mathbf{x})u$$

with u the scalar control input, x the scalar output, $\mathbf{x} = [x \ \dot{x} \ \dots \ x^{(n-1)}]^T$ the state vector and $f(\mathbf{x})$ and $b(\mathbf{x})$ possibly nonlinear functions of the state. Note that no derivatives of the control input u are present.

$b(x_1)$. The new state vector $\mathbf{x} = [x_1 \ x_2 \ u]^T = [z \ \dot{z} \ u]^T$ no longer contains only derivatives of the output z , hence the system is not in controllability canonical form. Standard sliding mode design techniques cannot be applied straight away.

State space representations are not unique, however. If we use the transformation $\boldsymbol{\xi} = \phi(\mathbf{x})$ given by

$$\begin{aligned}\xi_1 &= x_1 \\ \xi_2 &= x_2 \\ \xi_3 &= b(x_1) + a \cdot u\end{aligned}$$

and with inverse $\mathbf{x} = \phi^{-1}(\boldsymbol{\xi})$

$$\begin{aligned}x_1 &= \xi_1 \\ x_2 &= \xi_2 \\ u &= \frac{1}{a}(\xi_3 - b(\xi_1))\end{aligned}$$

in eqs. (4.12)-(4.14) we get

$$\begin{aligned}\dot{\xi}_1 &= x_2 \\ &= \xi_2 \\ \dot{\xi}_2 &= b(x_1) + a \cdot u \\ &= \xi_3 \\ \frac{1}{a} \left(\dot{\xi}_3 - \frac{db(\xi_1)}{d\xi_1} \dot{\xi}_1 \right) &= -\frac{u}{T} + \frac{u_d}{T} \\ &= -\frac{1}{aT} (\xi_3 - b(\xi_1)) + \frac{u_d}{T}\end{aligned}$$

or

$$\dot{\xi}_1 = \xi_2 \tag{4.15}$$

$$\dot{\xi}_2 = \xi_3$$

$$\begin{aligned}\dot{\xi}_3 &= \frac{db(\xi_1)}{d\xi_1} \xi_2 - \frac{1}{T} (\xi_3 - b(\xi_1)) + \frac{a}{T} u_d \\ &= b'(\boldsymbol{\xi}) + a' \cdot u_d,\end{aligned} \tag{4.16}$$

with $b'(\boldsymbol{\xi}) = \frac{db(\xi_1)}{d\xi_1}\xi_2 - \frac{1}{T}(\xi_3 - b(\xi_1))$ and $a' = a/T$. This is an alternative state space representation for (4.12)-(4.14). Nonlinearities can be cancelled by choosing

$$u_d = \frac{1}{a'}(-b'(\boldsymbol{\xi}) + v),$$

(with v the control law for the linearized system) since this representation is in controllability canonical form.

For more information about feedback linearization (the general case, conditions that have to be fulfilled, etc.) please refer to for instance Sastry (1999); Slotine and Li (1991); Khalil (2002).

A sliding mode controller for (4.15)-(4.16) can be designed by proposing the following sliding surface:

$$s(\boldsymbol{\xi}, t) = \ddot{e} + \alpha_1 \dot{e} + \alpha_0 e,$$

with $e = \xi_{1d} - \xi_1 = z_d - z$. Since $s(\boldsymbol{\xi}, t) = 0$ represents the desired dynamics when on the sliding, α_1 and α_0 have to be chosen so that e converges to zero if $s(\boldsymbol{\xi}, t) = 0$. In general, this is the case if the polynomial $p^2 + \alpha_1 p + \alpha_0$ is Hurwitz⁴. If we want the exponential convergence with time constant λ we had before, we can put both roots of the polynomial at $-1/\lambda$ by choosing $\alpha_1 = 2/\lambda$ and $\alpha_0 = 1/\lambda^2$.

Again, setting $\dot{s}(\boldsymbol{\xi}, t) = 0$, with $\dot{s}(\boldsymbol{\xi}, t)$ given by

$$\begin{aligned} \dot{s}(\boldsymbol{\xi}, t) &= \dot{\xi}_{3d} - \dot{\xi}_3 + \alpha_1 \ddot{e} + \alpha_0 \dot{e} \\ &= \dot{\xi}_{3d} - b'(\boldsymbol{\xi}) - a' \cdot u_d + \alpha_1 \ddot{e} + \alpha_0 \dot{e} \end{aligned}$$

allows us to determine the equivalent control

$$u_{eq} = \frac{1}{a'} \left(\dot{\xi}_{3d} - b'(\boldsymbol{\xi}) + \alpha_1 \ddot{e} + \alpha_0 \dot{e} \right),$$

which we can complement with a switching term to find the sliding mode control law

$$u = \frac{1}{a'} \left(\dot{\xi}_{3d} - b'(\boldsymbol{\xi}) + \alpha_1 \ddot{e} + \alpha_0 \dot{e} + K \operatorname{sgn} s \right).$$

A tracking simulation for this controller is shown in fig. 4.8. By comparing with fig. 4.7 we see that incorporating the actuator dynamics improves performance.

It is not without disadvantages though, since most of the robustness is lost. The uncertainties in eq. (4.13) are what is called unmatched, since they don't act in the input channel⁵ (Draženović, 1969). Feedback linearization will not be robust with

⁴A polynomial with real positive coefficients and roots which are either negative or pairwise conjugate with negative real parts.

⁵More precisely, they don't act in the input channel when considering state space representation (4.12)-(4.14) (where the input is u_d). After feedback linearization, i.e. in (4.15)-(4.16), the matching conditions (Draženović, 1969) lose their meaning since they would be automatically satisfied (Sira-Ramírez et al., 1996).

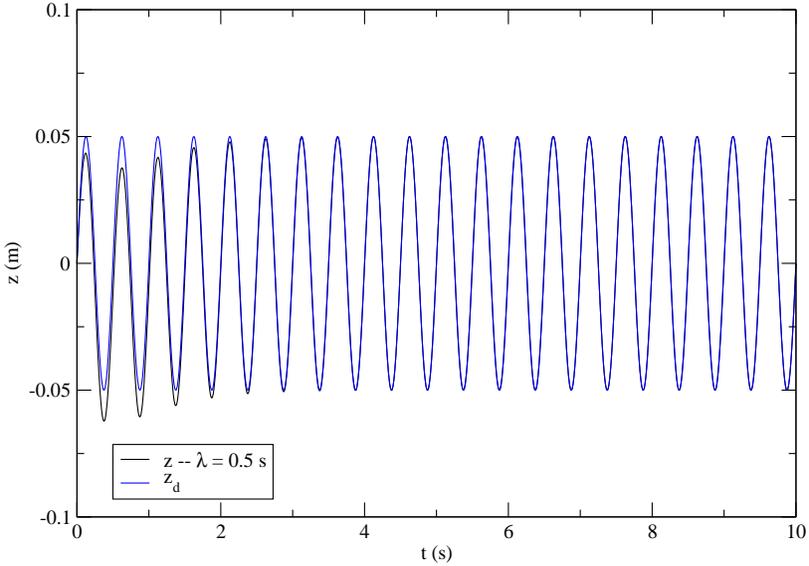


Figure 4.8: Simulated response of the sliding mode controlled nonlinear mass-spring system when tracking a sinusoidal trajectory. First order actuator dynamics is taken into account in the controller. A slight mismatch between the initial state and the desired initial state was introduced intentionally.

respect to these uncertainties (Sastry, 1999). Only uncertainties in the actuator dynamics, represented by eq. (4.14), are matched, and can be overcome by choosing the switching gain K high enough.

4.2.7 Internal dynamics

Let's suppose that the actuator force u is exerted by two antagonistically placed PPAM muscles (in a setup as shown in fig. 4.9), and that the desired pressure in each muscle is set using the Δp approach (see section 3.3.1). To keep the example simple, we assume the pressure dynamics in both muscles to be first order (as in eq. (3.2)), i.e. pressure variations due to volume changes as modeled in (3.5) are ignored. Using eqs. (2.2) and (3.2), the system's equations of motion then become

$$\dot{x}_1 = x_2 \quad (4.17)$$

$$\dot{x}_2 = b(x_1) + a \cdot (p_1 l_0^2 f_{t0}(\epsilon_1(x_1)) - p_2 l_0^2 f_{t0}(\epsilon_2(x_1))) \quad (4.18)$$

$$\dot{p}_1 = -\frac{p_1}{T} + \frac{p_m + \Delta p}{T} \quad (4.19)$$

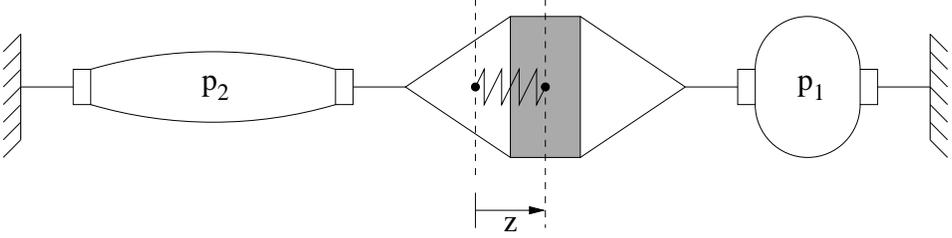


Figure 4.9: Antagonistic setup of muscles to exert force on the nonlinear mass-spring system.

$$\dot{p}_2 = -\frac{p_2}{T} + \frac{p_m - \Delta p}{T}. \quad (4.20)$$

This is a fourth-order system with state vector $\mathbf{x} = [x_1 \ x_2 \ p_1 \ p_2]^T$. The relative degree of the system is 3, which means that the output $z = x_1$ has to be differentiated 3 times for the input Δp to appear (Sastry, 1999):

$$\begin{aligned} \ddot{x}_2 &= \ddot{z} \\ &= \frac{db(x_1)}{dx_1} x_2 + a\dot{p}_1 l_0^2 f_{t0}(\epsilon_1(x_1)) + a p_1 l_0^2 \frac{df_{t0}(\epsilon_1(x_1))}{dx_1} x_2 \\ &\quad - a\dot{p}_2 l_0^2 f_{t0}(\epsilon_2(x_1)) - a p_2 l_0^2 \frac{df_{t0}(\epsilon_2(x_1))}{dx_1} x_2 \\ &= \frac{db(x_1)}{dx_1} x_2 + a l_0^2 x_2 \left(p_1 \frac{df_{t0}(\epsilon_1(x_1))}{dx_1} - p_2 \frac{df_{t0}(\epsilon_2(x_1))}{dx_1} \right) \\ &\quad + a \left(-\frac{p_1}{T} + \frac{p_m + \Delta p}{T} \right) l_0^2 f_{t0}(\epsilon_1(x_1)) - a \left(-\frac{p_2}{T} + \frac{p_m - \Delta p}{T} \right) l_0^2 f_{t0}(\epsilon_2(x_1)) \\ &= \frac{db(x_1)}{dx_1} x_2 + a l_0^2 x_2 \left(p_1 \frac{df_{t0}(\epsilon_1(x_1))}{dx_1} - p_2 \frac{df_{t0}(\epsilon_2(x_1))}{dx_1} \right) \\ &\quad + \frac{a}{T} (p_m - p_1) l_0^2 f_{t0}(\epsilon_1(x_1)) - \frac{a}{T} (p_m - p_2) l_0^2 f_{t0}(\epsilon_2(x_1)) \\ &\quad + \frac{a}{T} \Delta p \cdot l_0^2 (f_{t0}(\epsilon_1(x_1)) + f_{t0}(\epsilon_2(x_1))) \\ &= b''(\mathbf{x}) + a''(\mathbf{x}) \Delta p, \end{aligned}$$

with $b''(\mathbf{x})$ and $a''(\mathbf{x})$ defined by the above equation. We see that the system can be feedback linearized using a control law of the form

$$\Delta p = \frac{1}{a''(\mathbf{x})} (-b''(\mathbf{x}) + v).$$

The corresponding state transformation $\xi = \phi(\mathbf{x})$ is given by

$$\begin{aligned}\xi_1 &= x_1 \\ \xi_2 &= x_2 \\ \xi_3 &= b(x_1) + a \cdot (p_1 l_0^2 f_{t0}(\epsilon_1(x_1)) - p_2 l_0^2 f_{t0}(\epsilon_2(x_1)))\end{aligned}$$

and allows us to write the system as

$$\begin{aligned}\dot{\xi}_1 &= \xi_2 \\ \dot{\xi}_2 &= \xi_3 \\ \dot{\xi}_3 &= b'' + a'' \Delta p.\end{aligned}$$

Since the original system was fourth order, an extra system state exists. In this case, this state can be taken to be

$$\eta = p_1 + p_2,$$

with dynamics given by the sum of eqs. (4.19) and (4.20):

$$\dot{\eta} = -\frac{\eta}{T} + 2\frac{p_m}{T}.$$

We see that the evolution of η does not depend on the input Δp . It is called the internal dynamics (Slotine and Li, 1991; Sastry, 1999). This division of the system in a part where the input-output relation is linearized by feedback and another part that is independent of the input (internal dynamics) happens when the relative degree is less than the order.

Since the internal dynamics doesn't depend on the input, they cannot be stabilized by it. Thus, it is important that the internal dynamics are stable, otherwise feedback linearization can generally not be used.

4.3 Sliding mode control of the manipulator

4.3.1 Introduction

Most of the simulation-based work on sliding mode control of pneumatic artificial muscle systems available in literature (Cai and Yamaura, 1996; Repperger et al., 1998; Cai and Dai, 2000; Carbonell et al., 2001b; Cai and Dai, 2003; Lilly and Quesada, 2004; Lilly and Yang, 2005; Yang, 2006) does not consider pressure dynamics, a notable exception being a study by Sira-Ramírez et al. (1996) (who modeled pressure dynamics as a first order system in his simulations). Pressure dynamics is thus

considered to be parasitic, or much faster than the dynamics of the rest of the system, an assumption we feel is not valid in the case of sliding mode control of a pneumatic muscle system. A sliding mode controller will generally cause switching in the desired pressure at a much higher rate than any pressure regulating valve can impose. As an illustration, we note that Lilly and Quesada (2004); Lilly and Yang (2005) use the muscle model proposed by Reynolds et al. (2003) for McKibben-type artificial muscles. This model was validated by subjecting it to a triangle-shaped input pressure signal with a frequency of 0.17 Hz (period $\approx 5.88\text{ s}$), which most pressure regulating valves would be able to track without any problems. Pressure dynamics can safely be ignored at this frequency. A sliding mode controller, however, could switch the desired pressure at frequencies orders of magnitude higher, so in this case pressure dynamics must be considered. Ignoring it will most likely lead to an unusable controller, as was illustrated in the example discussed above (see section 4.2.5).

In all experimental studies of sliding mode control applied to pneumatic artificial muscle systems known to the author (Nouri et al., 1994; Hamerlain, 1995; Tondur and Lopez, 2000; Chettouh et al., 2006, 2008a,b), the pressure dynamics is implicitly taken into account by lumping together the valve, McKibben muscle and inertial load connected to the muscle and modeling the whole as a linear second order system.

In this section, we use the system model including pressure dynamics as introduced in chapter 3. Due to the included actuator dynamics, a state transformation is necessary to feedback linearize the system. A sliding mode tracking controller is then designed.

4.3.2 Complete system model

From (3.1) we can write

$$\begin{aligned}\ddot{\mathbf{q}} &= -H^{-1}C\dot{\mathbf{q}} - H^{-1}\mathbf{G} + H^{-1}\boldsymbol{\tau} \\ &= -A\dot{\mathbf{q}} - \mathbf{B} + H^{-1}\boldsymbol{\tau}\end{aligned}\quad (4.21)$$

with $A = H^{-1}C$ and $\mathbf{B} = H^{-1}\mathbf{G}$ (the inverse H^{-1} always exists due to the positive definiteness of the inertia matrix H). Together with (2.17) and (3.7) we get the full model of the system to be controlled:

$$\begin{aligned}\ddot{\mathbf{q}} &= -A\dot{\mathbf{q}} - \mathbf{B} + H^{-1} \begin{bmatrix} p_1 \cdot m_{\tau 1}(q_1) + p_2 \cdot m_{\tau 2}(q_1) \\ p_3 \cdot m_{\tau 3}(q_2) + p_4 \cdot m_{\tau 4}(q_2) \end{bmatrix} \\ \dot{p}_1 &= -\frac{p_1}{T_1} + \frac{p_{m_1} + \Delta p_1}{T_1} - n(P_{atm} + p_1) \cdot \frac{1}{V_1} \cdot \frac{dV_1}{dq_1} \dot{q}_1 \\ \dot{p}_2 &= -\frac{p_2}{T_2} + \frac{p_{m_1} + \Delta p_1}{T_2} - n(P_{atm} + p_2) \cdot \frac{1}{V_2} \cdot \frac{dV_2}{dq_1} \dot{q}_1 \\ \dot{p}_3 &= -\frac{p_3}{T_3} + \frac{p_{m_2} + \Delta p_2}{T_3} - n(P_{atm} + p_3) \cdot \frac{1}{V_3} \cdot \frac{dV_3}{dq_2} \dot{q}_2 \\ \dot{p}_4 &= -\frac{p_4}{T_4} + \frac{p_{m_2} + \Delta p_2}{T_4} - n(P_{atm} + p_4) \cdot \frac{1}{V_4} \cdot \frac{dV_4}{dq_2} \dot{q}_2\end{aligned}\quad (4.22)$$

This is a MIMO system with two inputs (Δp_1 and Δp_2) and two outputs (q_1 and q_2). Expressions for A , B and H^{-1} can be found in appendix C, section C.1.

4.3.3 Feedback linearization

Feedback linearizing (4.22) as a MIMO system would be the logical step to take. The result of this operation is, however, very dependent on the accuracy of the model. In this case, modeling errors (such as the linear approximation in (3.2)) are too important to give meaningful results (this was verified experimentally by implementing a sliding mode controller based on a MIMO-feedback linearization of (4.22)).

The approach we have taken here is to consider (4.22) to consist of two SISO systems, which are individually feedback linearized (only with respect to their own state variables) and individually controlled (one sliding mode controller per SISO system, i.e. per link). This approach (similar to what is done in Slotine and Li (1991), p. 398) essentially neglects the coupling that exists between both links, thus reducing the necessary modeling accuracy. Coupling effects are encountered as disturbances for the individual controllers.

When considering (4.22) to consist of two (coupled) input-affine SISO systems, we can write them as

$$\dot{\mathbf{x}}_i = \mathbf{f}_i + \mathbf{g}_i u_i \quad (4.23)$$

$$y_i = h_i(\mathbf{x}_i) \quad (4.24)$$

with $i = 1, 2$ (1 for upper arm, 2 for lower arm), $\mathbf{x}_i = [q_i \ \omega_i \ p_{2i-1} \ p_{2i}]^T$ being the state vectors (with $\omega_i = \dot{q}_i$), $u_i = \Delta p_i$ the scalar inputs, $y_i = h_i(\mathbf{x}_i) = q_i$ the scalar system outputs, and

$$\mathbf{f}_i = \begin{bmatrix} \omega_i \\ -a_{i,1}\omega_1 - a_{i,2}\omega_2 - b_i + h_{i,1}^{-1}\tau_1 + h_{i,2}^{-1}\tau_2 \\ \frac{p_{m_i} - p_{2i-1}}{T_{2i-1}} - \frac{n}{V_{2i-1}}(P_{atm} + p_{2i-1}) \frac{dV_{2i-1}}{dq_i} \omega_i \\ \frac{p_{m_i} - p_{2i}}{T_{2i}} - \frac{n}{V_{2i}}(P_{atm} + p_{2i}) \frac{dV_{2i}}{dq_i} \omega_i \end{bmatrix}$$

$$\mathbf{g}_i = \begin{bmatrix} 0 \\ 0 \\ 1/T_{2i-1} \\ -1/T_{2i} \end{bmatrix}$$

τ_1 and τ_2 are defined in eqs. (2.16)-(2.17), and expressions for the elements $a_{i,j}$, $h_{i,j}^{-1}$ and b_i of the matrices A and H and of vector B , respectively, can be found in appendix C, section C.1.

Because of the presence of the pressure dynamics these SISO systems are not in controllability canonical form (see footnote 3 on p. 90). This means we cannot

immediately apply standard sliding mode control techniques. We have to feedback linearize the systems first.

Both 4th order SISO systems (4.23) have relative degree 3, which implies that⁶ (Sastry, 1999)

$$L_{g_i} L_{f_i}^j h_i(\mathbf{x}_i) = 0 \quad j = 0, 1 \quad (4.25)$$

$$L_{g_i} L_{f_i}^2 h_i(\mathbf{x}_i) \neq 0 \quad (4.26)$$

Since both systems have relative degree 3, the coordinate transformation

$$\xi_{i1} = h_i(\mathbf{x}_i) \quad (4.27)$$

$$\xi_{i2} = L_{f_i} h_i(\mathbf{x}_i) \quad (4.28)$$

$$\xi_{i3} = L_{f_i}^2 h_i(\mathbf{x}_i) \quad (4.29)$$

$$\eta_i(\mathbf{x}_i) \quad \text{with} \quad L_{g_i} \eta_i(\mathbf{x}_i) \equiv 0 \quad (4.30)$$

transforms (4.23)-(4.24) to the required form (see for instance Sastry (1999)):

$$\dot{\xi}_{i1} = \xi_{i2} \quad (4.31)$$

$$\dot{\xi}_{i2} = \xi_{i3} \quad (4.32)$$

$$\dot{\xi}_{i3} = b_i(\boldsymbol{\xi}_i, \eta_i) + a_i(\boldsymbol{\xi}_i, \eta_i) u_i \quad (4.33)$$

$$\dot{\eta}_i = r_i(\boldsymbol{\xi}_i, \eta_i) \quad (4.34)$$

with $b_i(\boldsymbol{\xi}_i, \eta_i) = L_{f_i}^3 h_i$, $a_i(\boldsymbol{\xi}_i, \eta_i) = L_{g_i} L_{f_i}^2 h_i$, $r_i(\boldsymbol{\xi}_i, \eta_i) = L_{f_i} \eta_i$ and $\boldsymbol{\xi}_i = [\xi_{i1} \quad \xi_{i2} \quad \xi_{i3}]^T$. Since the practical calculations quickly become very complex all expressions were calculated with symbolic mathematics software. The same software was used to show that (4.27)-(4.30) is in fact a diffeomorphism (by showing that its Jacobian is nonsingular), as is required for feedback linearization Sastry (1999); Slotine and Li (1991).

Full details about the coordinate transformations, including a discussion of the stability of the internal dynamics $\dot{\eta}_i = r_i(\boldsymbol{\xi}_i, \eta_i)$, can be found in appendix C, section C.2.

⁶The scalar function

$$L_{\mathbf{f}} h(\mathbf{x}) = \frac{\partial h}{\partial \mathbf{x}} \mathbf{f}(\mathbf{x}) = \sum_{i=1}^n \frac{\partial h}{\partial x_i} f_i(\mathbf{x})$$

is called the Lie derivative of the scalar function $h(\mathbf{x})$ with respect to the vector field $\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}) \quad f_2(\mathbf{x}) \quad \cdots \quad f_n(\mathbf{x})]^T$. It is simply the directional derivative of h in the direction of \mathbf{f} . The Lie derivative can be taken multiple times, for instance $L_{\mathbf{f}}^2 h(\mathbf{x}) = L_{\mathbf{f}} L_{\mathbf{f}} h(\mathbf{x}) = L_{\mathbf{f}} (L_{\mathbf{f}} h)(\mathbf{x})$. In general $L_{\mathbf{f}}^k h(\mathbf{x})$ is defined as $L_{\mathbf{f}} (L_{\mathbf{f}}^{k-1} h)(\mathbf{x})$, with $L_{\mathbf{f}}^0 h(\mathbf{x}) = h(\mathbf{x})$. See for instance Sastry (1999); Slotine and Li (1991) for more information.

4.3.4 Controller

The systems (4.31)-(4.34) are now in a form suitable for sliding mode control techniques. To design a sliding mode controller that makes these systems track their respective desired output trajectories $y_{im}(t) = q_{im}(t)$, we use $e_i(t) = y_{im}(t) - y_i(t)$ to define the sliding surfaces $s_i(\mathbf{x}_i, t) = 0$, with $s_i(\mathbf{x}_i, t)$ given by

$$s_i(\mathbf{x}_i, t) = \ddot{e}_i + \alpha_{i1}\dot{e}_i + \alpha_{i0}e_i \quad (4.35)$$

The coefficients α_{i0} and α_{i1} are chosen so that the polynomials $p^2 + \alpha_{i1}p + \alpha_{i0}$ are Hurwitz (see footnote p. 92). This way, if the state trajectory is on the sliding surface (if $s_i = 0$), the error will tend to zero according to the error dynamics (4.35).

In view of eq. (4.25)-(4.26), we have

$$\begin{aligned} \dot{y}_i &= \frac{\partial h_i}{\partial \mathbf{x}_i} \dot{\mathbf{x}}_i = L_{\mathbf{f}_i} h_i(\mathbf{x}_i) + L_{\mathbf{g}_i} h_i(\mathbf{x}_i) u_i = L_{\mathbf{f}_i} h_i(\mathbf{x}_i) \\ \ddot{y}_i &= L_{\mathbf{f}_i}^2 h_i(\mathbf{x}_i) + L_{\mathbf{g}_i} L_{\mathbf{f}_i} h_i(\mathbf{x}_i) u_i = L_{\mathbf{f}_i}^2 h_i(\mathbf{x}_i) \end{aligned}$$

so (4.35) becomes

$$s_i(\mathbf{x}_i, t) = \ddot{y}_{im} - L_{\mathbf{f}_i}^2 h_i(\mathbf{x}_i) + \alpha_{i1}(\dot{y}_{im} - L_{\mathbf{f}_i} h_i(\mathbf{x}_i)) + \alpha_{i0}(y_{im} - h_i).$$

When on the sliding surface the system dynamics are described by $\dot{s}_i = 0$, which becomes (using $a_i = L_{\mathbf{g}_i} L_{\mathbf{f}_i}^2 h_i(\mathbf{x}_i)$, $b_i = L_{\mathbf{f}_i}^3 h_i(\mathbf{x}_i)$ and $L_{\mathbf{g}_i} L_{\mathbf{f}_i} h_i(\mathbf{x}_i) = 0$):

$$\dot{s}_i(\mathbf{x}_i, t) = \ddot{y}_{im} - L_{\mathbf{f}_i}^3 h_i(\mathbf{x}_i) - L_{\mathbf{g}_i} L_{\mathbf{f}_i}^2 h_i(\mathbf{x}_i) u_i + \quad (4.36)$$

$$\alpha_{i1}(\ddot{y}_{im} - L_{\mathbf{f}_i}^2 h_i(\mathbf{x}_i) - L_{\mathbf{g}_i} L_{\mathbf{f}_i} h_i(\mathbf{x}_i)) + \alpha_{i0}(\dot{y}_{im} - \dot{h}_i) \quad (4.37)$$

$$= \ddot{y}_{im} - b_i - a_i u_i + \alpha_{i1}(\ddot{y}_{im} - \ddot{y}_i) + \alpha_{i0}(\dot{y}_{im} - \dot{h}_i) \quad (4.38)$$

$$= \ddot{y}_{im} - b_i - a_i u_i + \alpha_{i1}\ddot{e}_i + \alpha_{i0}\dot{e}_i \quad (4.39)$$

$$= 0. \quad (4.40)$$

The equivalent control (or continuous control law that would maintain $\dot{s}_i = 0$ if the dynamics were exactly known) is thus given by

$$u_{eq,i} = \frac{1}{a_i} (\ddot{y}_{im} - b_i + \alpha_{i1}\ddot{e}_i + \alpha_{i0}\dot{e}_i)$$

($a_i = L_{\mathbf{g}_i} L_{\mathbf{f}_i}^2 h_i \neq 0$ because of (4.26), see also appendix C, section C.2). By adding a discontinuous switching term we finally get the sliding mode control law:

$$u_i = \frac{1}{a_i} (\ddot{y}_{im} - b_i + \alpha_{i1}\ddot{e}_i + \alpha_{i0}\dot{e}_i + K_i \operatorname{sgn}(s_i(\mathbf{x}_i, t))) \quad (4.41)$$

Substitution of this control law in (4.7) gives

$$\dot{s}_i(x_i, t) = -K_i \operatorname{sgn}(s_i(\mathbf{x}_i, t)),$$

so if K_i is large enough to overcome system uncertainties, modeling inaccuracies, interactions between the links and perturbations we have

$$\dot{s}_i s_i < 0,$$

which implies that the sliding surface will be attractive and will be reached in finite time.

As is well known, once the sliding surface is reached the term $K \operatorname{sgn}(s_i(\mathbf{x}_i, t))$ in (4.41) will cause excessive control chattering. To reduce this problem, a boundary layer (see Slotine and Li (1991)) is introduced by replacing $\operatorname{sgn}(s_i)$ with $\operatorname{sat}(s_i/\Gamma_i)$, where Γ_i are constants determining the width of the boundary layers. The sat function is defined in eq. (4.10). Of course, these boundary layers diminish tracking precision.

4.4 Results

4.4.1 Simulation

Before implementing, the controller was tested in simulation. The system model (4.22) was used to simulate the system's motion, so no extra unmodeled dynamics was introduced. Of course, the system parameters are never exactly known, so differences between the parameters used by the controller and the ones used by the simulation model were introduced. Since in practice it is difficult to measure angular velocity signals, white noise was added to these signals in the simulation. Boundary layers were introduced to smoothen the control signals.

Figure 4.10 shows the performance of the developed controller when the system had to track a circular trajectory (diameter 20 cm) in Cartesian space during a period of 5 seconds. In this particular simulation run, the mass of both links was chosen 20% higher than the values used in the controller. The left part of fig. 4.10 shows how the system tracks the trajectory (only one period shown), while the right part shows the position error e , given by

$$e = \sqrt{(x - x_d)^2 + (y - y_d)^2}.$$

x and y indicate the Cartesian position of the tool center point, x_d and y_d give the desired position.

The simulations show that the controller is capable of achieving relatively good tracking performance even when considerable noise is injected and parameter values differ significantly from their expected values, provided that any unmodeled

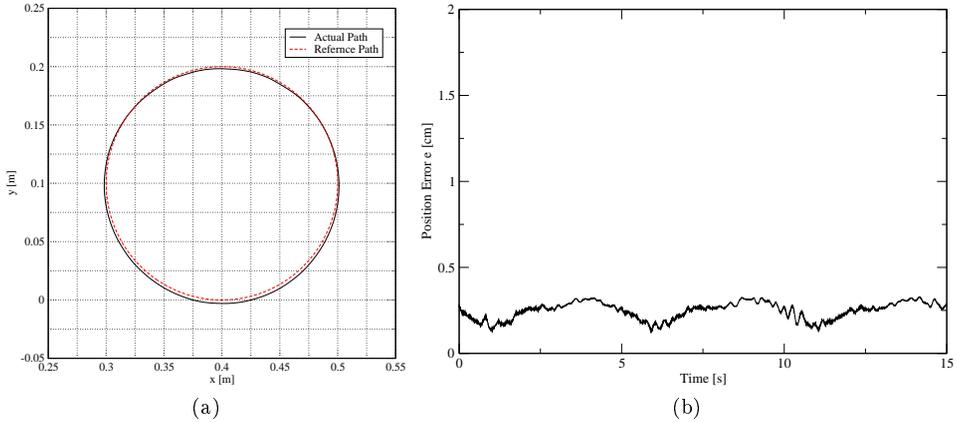


Figure 4.10: Simulation results for tracking a circular trajectory. Fig. (a) shows the desired and the actual trajectory for one revolution, fig. (b) shows the position error for three revolutions.

dynamics (if present) can be considered to be much faster than the dynamics of the rest of the system.

4.4.2 Experiments

On the actual arm, we expect unmodeled dynamics to be present. The main contributors are the pressure regulating valves since they were modeled as simple first order systems, a rough approximation at best.

When testing the sliding mode controller, the chattering problem turned out to be quite severe. This can partly be interpreted as an attempt of the controller to compensate for the unmodeled dynamics present in the system. Experimentation and tuning showed that choosing the valve time constants T_i 20 to 50 percent lower than the (estimated) actual value significantly reduced chattering without affecting tracking performance too much. Significant boundary layers were nevertheless necessary.

To illustrate performance, the position error e recorded during the same experiment as the one simulated in section 4.4.1 is shown in figure 4.11. It is clear that the necessary introduction of boundary layers causes a significant tracking error. Part of the error can also be explained by the fact that the controller is not robust with respect to uncertainties in the mechanical model, since they are unmatched (see also section 4.2.6).

The results show that in the context of sliding mode control of pneumatic muscle systems simulation results should be considered with care. Pressure dynamics is

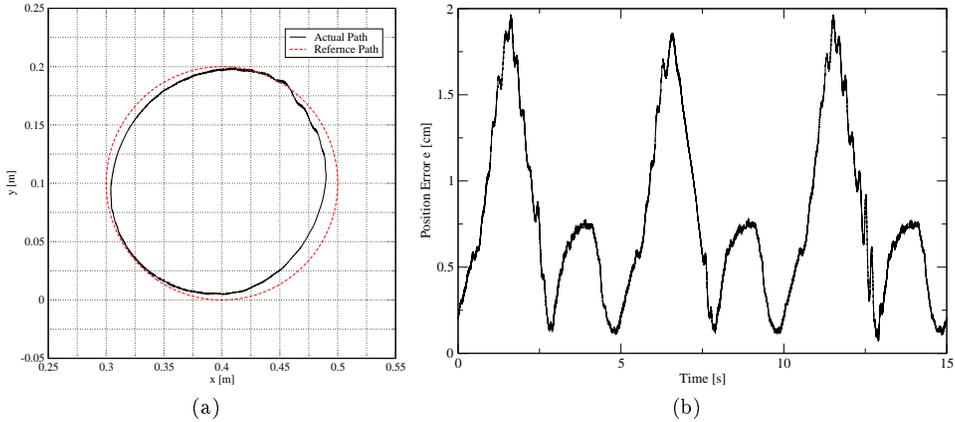


Figure 4.11: Measurements taken while tracking a circular trajectory. Fig. (a) shows the desired and the actual trajectory for one revolution, fig. (b) shows the measured position error for three revolutions.

too slow to be ignored, and difficult to model accurately. Introducing approximate models makes it possible to realize an effective controller, but more research is still necessary to improve tracking performance.

The presented experimental results were not obtained using the hardware configuration as described in section 2.4, the most important difference being the servo valves. In the experiments, the gauge pressure in muscles 1 and 3 was controlled by Kolvenbach KPS 3/4-10 valves with internal pressure sensor. The pressure in the other muscles was controlled using Tecno G1/8,NW2 valves, also with internal pressure sensor.

Due to the controller's high complexity (as can be seen in appendix C), the need for extensive manual tuning and the chattering problem, it was decided not to re-implement it on the new hardware.

4.5 Conclusion

In this chapter, a sliding mode controller for the 2-DOF planar pneumatic manipulator was presented. The decision to try sliding mode control was inspired by its good safety features, its robustness and encouraging results reported in literature.

In practice, it turns out that sliding mode control is very hard to implement. Pressure dynamics has to be taken into account to make it work, at the cost of an enormous increase in complexity. Additionally, robustness is lost, which is one of the most salient properties of sliding mode control. In spite of this,

experiments show that with manual tuning reasonable results can be obtained, although chattering remains a problem.

Most of these issues can be overcome by using proxy-based sliding mode control, which is the subject of the next chapter.

Chapter 5

Proxy-Based Sliding Mode Control

5.1 Introduction

Proxy-Based Sliding Mode Control (PSMC) was introduced by Kikuuwe and Fujimoto (2006) as a robot control method that can combine accurate tracking with a smooth and slow response to large positional errors.

The slow response is based on sliding mode control, but instead of controlling the robot the sliding mode controller acts on the proxy, a virtual object considered to be attached to the robot by means of a PID-type virtual coupling. This concept results in a discrete-time controller free of the chattering that plagues sliding mode control.

The sliding mode part of the controller dominates the response to large position errors, which should be slow for safety reasons, without impairing tracking performance in normal operation, which is mainly determined by the virtual coupling.

This chapter describes PSMC, how it was applied to the pneumatic arm and how well it performs. Safety aspects are discussed in the next chapter.

As an application of the manipulator being used in contact with humans, an interactive mode was implemented that allows the users to manually reposition the end-effector. This was done with admittance control, using PSMC for the inner position control loop.

The chapter is organized as follows: section 5.2 starts with a detailed analysis of the proxy-based sliding mode controller, its behavior and its properties. Section 5.3 describes both task-space and joint-space versions of PSMC, as well as a PID controller, to be used for comparing performance. Experimental results are presented in section 5.3.4, and discussed in section 5.3.5. The interactive mode is described in section 5.4, which also introduces the admittance controller and discusses two ways of estimating the force exerted on the end-effector.

5.2 Proxy-based sliding mode control

5.2.1 Introduction

Commonly used robotics tracking control methods have not generally been designed with safety in mind. A substantial position error of the end effector, caused by an unforeseen event, can provoke a violent, and thus unsafe, response from the robot. In section 4.2.2, an example illustrated that with a suitable choice of sliding surface, sliding mode control can provide a smooth, slow recovery in such situations, which increases safety. Pure sliding mode control can be very difficult to implement for pneumatic muscle systems, however, as was seen in chapter 4.

Sliding mode's main problem is chattering (see section 4.2.4), which is caused by (inevitable) delays in the discontinuous switching of the control output. Kikuuwe and Fujimoto (2006) have proposed proxy-based sliding mode control as a solution to this problem. It extends PID control with the slow recovery capability of sliding mode control without compromising tracking performance, and was tested on a 2-DOF electrically actuated robot without passive compliance.

5.2.2 Replacing the signum function

The discontinuous signum function $\text{sgn}(\cdot)$ is traditionally defined as (Khalil, 2002)

$$\text{sgn}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0. \end{cases}$$

Proxy-based sliding mode control (PSMC) is based on a modified definition of the signum function,

$$\text{sgn}(x) \begin{cases} = 1 & \text{if } x > 0 \\ \in [-1, 1] & \text{if } x = 0 \\ = -1 & \text{if } x < 0, \end{cases} \quad (5.1)$$

where $\text{sgn}(x)$ can take any value in the interval $[-1, 1]$ when $x = 0$. Using this definition, Kikuuwe and Fujimoto (2006) show the following equivalence:

$$y = \text{sgn}(x - y) \iff y = \text{sat}(x), \quad (5.2)$$

with the unit saturation function $\text{sat}(\cdot)$ defined as in eq. (4.10). See appendix D for their proof and for a vectorial version of (5.2).

Fig. 5.1 illustrates the significance of (5.2): if the discontinuous switching function $\text{sgn}(\cdot)$ is enclosed in a feedback loop without time delay, it can be replaced by the unit saturation function, which is continuous.

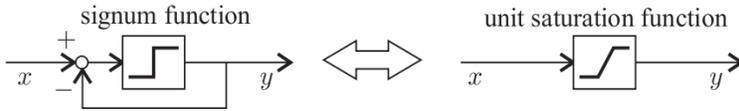


Figure 5.1: Equivalence of $y = \text{sgn}(x - y)$ and $y = \text{sat}(x)$. Figure reproduced from Kikuuwe and Fujimoto (2006).

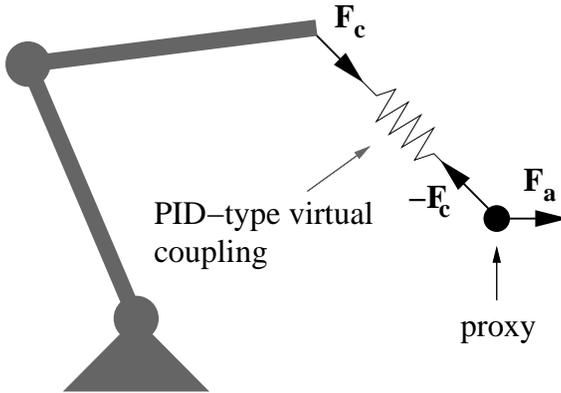


Figure 5.2: Principle of Proxy-Based Sliding Mode Control.

5.2.3 Proxy

In order to remove all delays in the feedback loop around the sgn element, the loop must be closed within the controller software itself (i.e. without passing through any (physical) sensors or actuators). In proxy-based sliding mode control, this is achieved by introducing a proxy or “god-object”, a concept borrowed from haptics (Zilles and Salisbury, 1995).

The proxy represents a virtual object that is controlled by an ideal sliding mode controller, without chattering. It is considered to be connected to the robot’s end-effector by means of a PID-type virtual coupling (a concept that originated in haptics as well, see Colgate et al. (1995)). This is illustrated in figure 5.2 for a 2-DOF robot in the horizontal plane¹.

The sliding mode controller exerts a force \mathbf{F}_a on the proxy. The PID-type virtual coupling will cause an interaction force \mathbf{F}_c between end effector and proxy, depending on their relative positions. The (statical) torques that would be produced in

¹In case of a PD-coupling, the proportional part (P) is usually represented by a spring, while the derivative part (D) of the coupling is represented by a damper. In proxy-based sliding mode control, the coupling is of the PID-type (in general). Since there is no similar physical analog for the integral part (I) of the coupling, the whole PID-type virtual coupling is simply depicted as a spring in fig. 5.2.

the robot joints if \mathbf{F}_c were physically present are given by the well-known relation (see for instance Spong et al. (2006))

$$\boldsymbol{\tau} = \mathbf{J}^T(\mathbf{q}) \mathbf{F}_c \quad (5.3)$$

(with $\mathbf{J}(\mathbf{q})$ the the robot's Jacobian matrix). Actually applying these torques will cause the end effector's position to be servo-controlled to follow the proxy's position.

If \mathbf{r}_p and $\dot{\mathbf{r}}_p$ are the proxy's position and velocity, and \mathbf{r}_d and $\dot{\mathbf{r}}_d$ are the desired position and velocity, a sliding mode control law that can be used to control the proxy is given by²

$$\mathbf{F}_a = F \mathbf{sgn}(\mathbf{s}) \quad (5.4)$$

with

$$\mathbf{s} = (\mathbf{r}_d - \mathbf{r}_p) + \lambda (\dot{\mathbf{r}}_d - \dot{\mathbf{r}}_p). \quad (5.5)$$

The quantity F in eq. (5.4) can be considered as an actuator force magnitude limit, representing the maximum force (in norm) that will be exerted by the sliding mode controller. As explained in section 4.2.1, once the proxy is kept on the sliding surface $\mathbf{s} = 0$, its position and velocity errors will exponentially decay to zero with time constant $\lambda > 0$. Thus, it will gently converge to the desired trajectory.

The force produced by the PID-type virtual coupling is

$$\mathbf{F}_c = K_p(\mathbf{r}_p - \mathbf{r}) + K_i \int (\mathbf{r}_p - \mathbf{r}) dt + K_d(\dot{\mathbf{r}}_p - \dot{\mathbf{r}}) \quad (5.6)$$

with \mathbf{r} and $\dot{\mathbf{r}}$ the end effector's actual position and velocity, respectively. In order to eliminate the integral in (5.6), we introduce

$$\mathbf{a} = \int (\mathbf{r}_p - \mathbf{r}) dt, \quad (5.7)$$

(which can be considered to be a state of the proxy-virtual coupling system) to obtain

$$\mathbf{F}_c = K_p \dot{\mathbf{a}} + K_i \mathbf{a} + K_d \ddot{\mathbf{a}}. \quad (5.8)$$

By setting

$$\boldsymbol{\sigma} = (\mathbf{r}_d - \mathbf{r}) + \lambda (\dot{\mathbf{r}}_d - \dot{\mathbf{r}}) \quad (5.9)$$

eq. (5.4) can also be written as a function of \mathbf{a} ,

$$\mathbf{F}_a = F \mathbf{sgn}(\boldsymbol{\sigma} - \dot{\mathbf{a}} - \lambda \ddot{\mathbf{a}}). \quad (5.10)$$

The equations of motion of the proxy (initially modeled as a point mass m) are given by

$$m \ddot{\mathbf{r}}_p = \mathbf{F}_a - \mathbf{F}_c. \quad (5.11)$$

²The signum function of a vector is defined in eq. (D.6) on page 190.

If we would want to implement a controller using (5.8) and (5.10), the motion of the virtual proxy mass would have to be simulated in software (by numerically integrating (5.11)). Kikuuwe and Fujimoto (2006), however, realized that the proxy mass can be set to zero. Equation (5.11) then gives us

$$\mathbf{F}_a = \mathbf{F}_c, \quad (5.12)$$

showing that the forces acting on the proxy are always balanced. This and other implications of having a massless proxy are investigated in more detail in section D.2.

Since both forces are equal, we set

$$\mathbf{F}_a = \mathbf{F}_c \equiv \mathbf{f}. \quad (5.13)$$

Eqs. (5.10) and (5.8) then become

$$\mathbf{f} = F \operatorname{sgn}(\boldsymbol{\sigma} - \dot{\mathbf{a}} - \lambda \ddot{\mathbf{a}}) \quad (5.14)$$

$$\mathbf{f} = K_p \dot{\mathbf{a}} + K_i \mathbf{a} + K_d \ddot{\mathbf{a}} \quad (5.15)$$

In order to calculate which torques to apply, we have to know \mathbf{f} (see eq. (5.3)).

5.2.4 Discrete-time controller

One way to calculate \mathbf{f} is using a discrete-time representation of (5.14)-(5.15).

We write the value of vector \mathbf{x} at timestep k as $\mathbf{x}[k]$ (i.e. $\mathbf{x}[k] = \mathbf{x}(kT)$ if T is the sampling period). By introducing the backward difference operator ∇ , defined by

$$\nabla \mathbf{x}[k] = \mathbf{x}[k] - \mathbf{x}[k-1], \quad (5.16)$$

we can approximate $\dot{\mathbf{x}}$ at timestep k by

$$\dot{\mathbf{x}}[k] = \frac{\nabla \mathbf{x}[k]}{T} \quad (5.17)$$

and $\ddot{\mathbf{x}}$ by

$$\begin{aligned} \ddot{\mathbf{x}}[k] &= \frac{\nabla \dot{\mathbf{x}}[k]}{T} \\ &= \frac{\nabla^2 \mathbf{x}[k]}{T^2} \end{aligned} \quad (5.18)$$

with

$$\nabla^2 \mathbf{x}[k] = \nabla(\nabla \mathbf{x}[k])$$

$$\begin{aligned}
&= \nabla \mathbf{x}[k] - \nabla \mathbf{x}[k-1] \\
&= \mathbf{x}[k] - 2\mathbf{x}[k-1] + \mathbf{x}[k-2].
\end{aligned}$$

Using these definitions, a discrete time representation of (5.14)-(5.15) is given by

$$\mathbf{f}[k] = F \mathbf{sgn} \left(\boldsymbol{\sigma}[k] - \frac{\nabla \mathbf{a}[k]}{T} - \lambda \frac{\nabla^2 \mathbf{a}[k]}{T^2} \right) \quad (5.19)$$

$$\mathbf{f}[k] = K_p \frac{\nabla \mathbf{a}[k]}{T} + K_i \mathbf{a}[k] + K_d \frac{\nabla^2 \mathbf{a}[k]}{T^2}, \quad (5.20)$$

with $\boldsymbol{\sigma}[k]$ given by (see eq. (5.9))

$$\boldsymbol{\sigma}[k] = (\mathbf{r}_d[k] - \mathbf{r}[k]) + \lambda (\dot{\mathbf{r}}_d[k] - \dot{\mathbf{r}}[k]) \quad (5.21)$$

(we don't have to approximate $\dot{\mathbf{r}}_d[k]$ and $\dot{\mathbf{r}}[k]$ using (5.17) since they are known). Eqs. (5.19) and (5.20) can be considered as a system of two algebraic equations in two unknowns, $\mathbf{f}[k]$ and $\mathbf{a}[k]$.

A first step in the solution (which is described in detail in section D.3) is to solve (5.20) for $\mathbf{a}[k]$:

$$\mathbf{a}[k] = \frac{\mathbf{a}[k-1](K_p T + K_d) + K_d \nabla \mathbf{a}[k-1] + T^2 \mathbf{f}[k]}{K_i T^2 + K_p T + K_d}. \quad (5.22)$$

Substitution in (5.19) gives (calculation shown in section D.3)

$$\mathbf{f}[k] = F \mathbf{sgn}(\mathbf{f}^*[k] - \mathbf{f}[k]), \quad (5.23)$$

with

$$\mathbf{f}^*[k] = \frac{K_i T^2 + K_p T + K_d}{T + \lambda} \boldsymbol{\sigma}[k] + K_i \mathbf{a}[k-1] + \frac{\lambda(K_i T + K_p) - K_d}{T(T + \lambda)} \nabla \mathbf{a}[k-1]. \quad (5.24)$$

Eq. (5.23) shows that the discretization has allowed the ‘‘closure of a feedback loop’’ around the signum function within the software of the controller (cf. section 5.2.2). Since $F > 0$ we can rewrite (5.23) as

$$\begin{aligned}
\frac{\mathbf{f}[k]}{F} &= \mathbf{sgn}(\mathbf{f}^*[k] - \mathbf{f}[k]) \\
&= \mathbf{sgn} \left(\frac{\mathbf{f}^*[k]}{F} - \frac{\mathbf{f}[k]}{F} \right).
\end{aligned}$$

and apply the vectorial version of (5.2) (given in eq. (D.5)) to get

$$\mathbf{f}[k] = F \mathbf{sat} \left(\frac{\mathbf{f}^*[k]}{F} \right)$$

$$= \begin{cases} \mathbf{f}^* [k] & \text{if } \|\mathbf{f}^* [k]\| \leq F \\ F \frac{\mathbf{f}^* [k]}{\|\mathbf{f}^* [k]\|} & \text{if } \|\mathbf{f}^* [k]\| > F. \end{cases} \quad (5.25)$$

Once $\mathbf{f} [k]$ is known from (5.25), it can be used to calculate $\mathbf{a} [k]$ using (5.22).

The procedure to calculate the output of the proxy-based sliding mode controller at timestep k can be summarized as follows:

$$\boldsymbol{\sigma} [k] = (\mathbf{r}_d [k] - \mathbf{r} [k]) + \lambda (\dot{\mathbf{r}}_d [k] - \dot{\mathbf{r}} [k]) \quad (5.26)$$

$$\begin{aligned} \mathbf{f}^* [k] &= \frac{K_i T^2 + K_p T + K_d}{T + \lambda} \boldsymbol{\sigma} [k] + K_i \mathbf{a} [k - 1] \\ &\quad + \frac{\lambda (K_i T + K_p) - K_d}{T (T + \lambda)} \nabla \mathbf{a} [k - 1] \end{aligned} \quad (5.27)$$

$$\mathbf{f} [k] = \begin{cases} \mathbf{f}^* [k] & \text{if } \|\mathbf{f}^* [k]\| \leq F \\ F \frac{\mathbf{f}^* [k]}{\|\mathbf{f}^* [k]\|} & \text{if } \|\mathbf{f}^* [k]\| > F. \end{cases} \quad (5.28)$$

$$\mathbf{a} [k] = \frac{\mathbf{a} [k - 1] (K_p T + K_d) + K_d \nabla \mathbf{a} [k - 1] + T^2 \mathbf{f} [k]}{K_i T^2 + K_p T + K_d}. \quad (5.29)$$

The calculated value of $\mathbf{f} [k]$ can then be used to calculate the torques that have to be applied in the joints using eq. (5.3),

$$\boldsymbol{\tau} [k] = \mathbf{J}^T (\mathbf{q} [k]) \mathbf{f} [k]. \quad (5.30)$$

5.2.5 Continuous-time controller

It is also possible to obtain a continuous-time solution of (5.14)-(5.15). From (5.15) we have

$$\ddot{\mathbf{a}} = -\frac{K_p \dot{\mathbf{a}} + K_i \mathbf{a}}{K_d} + \frac{1}{K_d} \mathbf{f}.$$

Substitution in (5.14) gives (bearing in mind that $\lambda > 0$, $K_d > 0$ and $F > 0$)

$$\begin{aligned} \frac{1}{F} \mathbf{f} &= \text{sgn} \left(\boldsymbol{\sigma} - \dot{\mathbf{a}} + \frac{\lambda}{K_d} (K_p \dot{\mathbf{a}} + K_i \mathbf{a}) - \frac{\lambda}{K_d} \mathbf{f} \right) \\ &= \text{sgn} \left(\frac{\lambda}{K_d} \left(\frac{K_d}{\lambda} (\boldsymbol{\sigma} - \dot{\mathbf{a}}) + K_p \dot{\mathbf{a}} + K_i \mathbf{a} - \mathbf{f} \right) \right) \\ &= \text{sgn} \left(\frac{1}{F} \left(\frac{K_d}{\lambda} (\boldsymbol{\sigma} - \dot{\mathbf{a}}) + K_p \dot{\mathbf{a}} + K_i \mathbf{a} \right) - \frac{1}{F} \mathbf{f} \right), \end{aligned}$$

and application of (D.5) results in

$$\mathbf{f} = F \text{sat} \left(\frac{1}{F} \left(\frac{K_d}{\lambda} (\boldsymbol{\sigma} - \dot{\mathbf{a}}) + K_p \dot{\mathbf{a}} + K_i \mathbf{a} \right) \right)$$

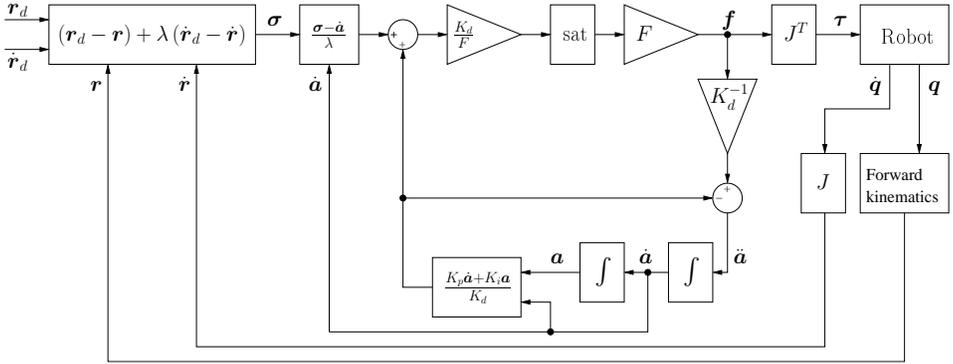


Figure 5.3: Schematic overview of the proxy-based sliding mode controller in continuous time.

$$\begin{aligned}
 &= F \text{sat} \left(\frac{K_d}{F} \left(\frac{\sigma - \dot{\mathbf{a}}}{\lambda} + \frac{K_p \dot{\mathbf{a}} + K_i \mathbf{a}}{K_d} \right) \right) \\
 &= F \text{sat} \left(\frac{\mathbf{f}^*}{F} \right)
 \end{aligned}$$

with

$$\mathbf{f}^* = K_d \left(\frac{\sigma - \dot{\mathbf{a}}}{\lambda} + \frac{K_p \dot{\mathbf{a}} + K_i \mathbf{a}}{K_d} \right).$$

Thus, the continuous-time version of the PSMC controller can be summarized as

$$\begin{aligned}
 \mathbf{f}^* &= K_d \left(\frac{\sigma - \dot{\mathbf{a}}}{\lambda} + \frac{K_p \dot{\mathbf{a}} + K_i \mathbf{a}}{K_d} \right) \\
 \mathbf{f} &= F \text{sat} \left(\frac{\mathbf{f}^*}{F} \right) \\
 \ddot{\mathbf{a}} &= -\frac{K_p \dot{\mathbf{a}} + K_i \mathbf{a}}{K_d} + \frac{1}{K_d} \mathbf{f},
 \end{aligned}$$

with σ given by eq. (5.9) and \mathbf{a} by eq. (5.7). A schematic overview of the controller is given in fig. 5.3.

Since the implementation is always in discrete time, the rest of this chapter focuses on the discrete-time version of PSMC that was described in section 5.2.4.

5.2.6 Discussion

5.2.6.1 Sliding surface

The PSMC controller (5.26)-(5.29) is only concerned with controlling the proxy such that it exhibits the desired dynamics given by $\mathbf{s} = 0$, with \mathbf{s} defined in (5.5). The influence of the robotic manipulator is considered to be a disturbance, acting upon the proxy by means of the virtual coupling.

As explained in section D.2, the (massless) proxy's dynamics are defined by the force-equilibrium condition (5.12) given by (in continuous time)

$$F \mathbf{sgn}(\mathbf{s}) = \mathbf{F}_c, \quad (5.31)$$

which is thus satisfied at all times. In view of the definition of $\mathbf{sgn}(\cdot)$ eq. (D.6), $\|\mathbf{F}_c\|$ can only be less than F if $\mathbf{s} = 0$. If the virtual coupling between manipulator and proxy exerts a force lower than F (in norm) then \mathbf{s} is equal to zero and the proxy is said to be “on the sliding surface”. Thus, with $\|\mathbf{F}_c\| < F$ the proxy exhibits the desired dynamics $\mathbf{s} = 0$.

From (5.31) we also infer that $\|\mathbf{F}_c\| \leq F$, so the virtual coupling can never exert a higher force (in norm) than F . \mathbf{s} can only be different from zero (or the proxy “off the sliding surface”) if $\|\mathbf{F}_c\| = F$. $\|\mathbf{F}_c\| = F$ doesn't automatically imply $\mathbf{s} \neq 0$, though.

Alternatively, we can see this by looking at $\mathbf{s}[k]$, the discrete-time version of \mathbf{s} . According to (5.19), $\mathbf{s}[k]$ is given by

$$\mathbf{s}[k] = \boldsymbol{\sigma}[k] - \frac{\nabla \mathbf{a}[k]}{T} - \lambda \frac{\nabla^2 \mathbf{a}[k]}{T^2}.$$

As shown in section D.3 (eq. (D.20)), this is equivalent to

$$\mathbf{s}[k] = \frac{T + \lambda}{K_i T^2 + K_p T + K_d} (\mathbf{f}^*[k] - \mathbf{f}[k]), \quad (5.32)$$

with $\mathbf{f}^*[k]$ and $\mathbf{f}[k]$ defined by (5.27) and (5.28), respectively. Note that $\mathbf{f}[k] = \mathbf{F}_c[k]$, according to eq. (5.13).

From (5.32) we see that $\mathbf{s}[k] \neq 0$ if $\mathbf{f}^*[k] \neq \mathbf{f}[k]$. From (5.27) and (5.28) we know this implies $\|\mathbf{f}[k]\| = F$. $\mathbf{s}[k] = 0$ when $\mathbf{f}^*[k] = \mathbf{f}[k]$, which is the case if $\|\mathbf{f}[k]\| \leq F$.

We have thus reached the same conclusion: if the virtual coupling exerts a force strictly less (in norm) than F then the proxy is “on the sliding surface”. If it exerts a force equal to F , then the proxy may be “off the sliding surface”, but it isn't sure.

$\mathbf{f}^*[k]$ can be seen as the force that is needed for the proxy to achieve its desired dynamics. If this force isn't available (i.e. if it is higher (in norm) than the force limit F) then \mathbf{s} cannot be kept to zero, which implies that the proxy cannot be kept “on the sliding surface”.

In this section, statements about being on or off the sliding surface have been quoted, since when \mathbf{s} equals zero (and stays zero) the proxy can't really be considered to be in the "sliding mode". This is explained in section D.4. Since picturing the proxy on the sliding surface in state-space is still a helpful representation, we'll continue to use it (but without the quotes). On or off the sliding surface will be considered equivalent to $\mathbf{s} = 0$ and $\mathbf{s} \neq 0$, respectively.

5.2.6.2 Compliance

Consider the manipulator of fig. 5.2 to be in a state of static equilibrium, with both its end-effector and the proxy coinciding with the (constant) desired position. For simplicity, we'll also assume (for now) that the virtual coupling is purely proportional (i.e. $K_d = 0$ and $K_i = 0$).

Suppose an external force F_e is exerted on the end-effector in the x -direction that causes a small deviation Δx in the end-effector's x -position. Because of the deviation, the virtual coupling will exert a (virtual) force $K_p \Delta x$ between proxy and end-effector to pull the end-effector back to the proxy. Using (5.3), the controller causes the robot to apply this force, counteracting F_e . The larger the deviation, the larger the resisting force of the manipulator, but with a limit of F , since the virtual coupling can't exert forces higher than the force limit F . Increasing deviations thus cause increasing resisting forces, but only up to the limit F . All deviations Δx where $|\Delta x| > F/K_p$ are resisted with a force equal in magnitude to the force limit F , and cause the proxy to leave its sliding surface (it is dragged away from the desired position by the virtual coupling).

We can also see this behavior in the PSMC equations. Using the assumptions $K_d = 0$, $K_i = 0$ and $\dot{\mathbf{r}}_d = 0$, and using (5.26), (5.27) becomes

$$\begin{aligned} \mathbf{f}^* [k] &= \frac{K_p T}{T + \lambda} ((\mathbf{r}_d [k] - \mathbf{r} [k]) - \lambda \dot{\mathbf{r}} [k]) + \frac{\lambda K_p}{T + \lambda} \cdot \frac{\nabla \mathbf{a} [k - 1]}{T} \\ &= \frac{K_p}{T + \lambda} \left(T (\mathbf{r}_d [k] - \mathbf{r} [k]) - \lambda T \dot{\mathbf{r}} [k] + \lambda \frac{\nabla \mathbf{a} [k - 1]}{T} \right). \end{aligned}$$

If we assume that the external force is applied very gradually, we can assume all motion happens very slowly, i.e. $\dot{\mathbf{r}} \approx 0$. In view of (5.7) and (5.17), we can approximate $\nabla \mathbf{a} [k - 1]/T$ by $\mathbf{r}_p - \mathbf{r}$, so $\mathbf{f}^* [k]$ becomes

$$\mathbf{f}^* [k] = \frac{K_p}{T + \lambda} (T (\mathbf{r}_d [k] - \mathbf{r} [k]) + \lambda (\mathbf{r}_p [k - 1] - \mathbf{r} [k - 1]))$$

Under the assumption $\dot{\mathbf{r}} \approx 0$, we have $\mathbf{r} [k - 1] \approx \mathbf{r} [k]$. As long as the deviation doesn't pull the proxy off its sliding surface, we also have $\mathbf{r}_p = \mathbf{r}_d$, and since $\dot{\mathbf{r}}_d = 0$ we have $\mathbf{r}_d [k - 1] = \mathbf{r}_d [k]$. The above equation thus becomes

$$\mathbf{f}^* [k] = \frac{K_p}{T + \lambda} (T (\mathbf{r}_d [k] - \mathbf{r} [k]) + \lambda (\mathbf{r}_d [k] - \mathbf{r} [k]))$$

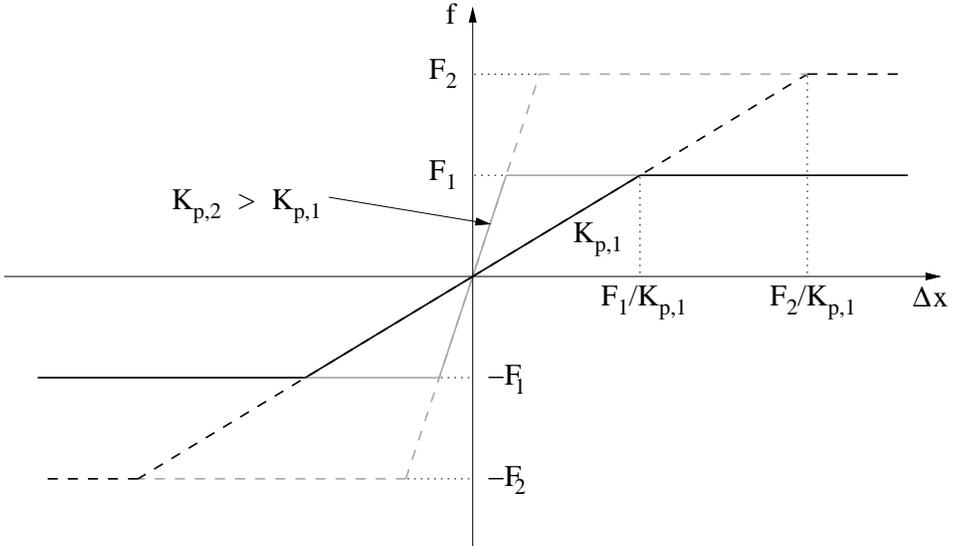


Figure 5.4: Compliant behavior of PSMC: the resisting force f is shown as a function of position error $\Delta x = x_d - x$ for two values of the force limit and two values of the proportional gain K_p . x_d and x represent the desired and actual position, respectively.

$$= K_p (\mathbf{r}_d[k] - \mathbf{r}[k]).$$

Using eq. (5.28), the resisting force is then given by

$$\mathbf{f}[k] = \begin{cases} K_p (\mathbf{r}_d[k] - \mathbf{r}[k]) & \text{if } \|K_p (\mathbf{r}_d[k] - \mathbf{r}[k])\| \leq F \\ F \frac{K_p (\mathbf{r}_d[k] - \mathbf{r}[k])}{\|K_p (\mathbf{r}_d[k] - \mathbf{r}[k])\|} & \text{if } \|K_p (\mathbf{r}_d[k] - \mathbf{r}[k])\| > F. \end{cases}$$

As long as the force limit is not reached, the virtual coupling (a virtual spring with spring constant K_p) acts like a real spring. This compliant behavior of the controller is illustrated in fig. 5.4 for the one-dimensional case.

Similarly, if the virtual coupling is of the PD-type (i.e. $K_i = 0$) then it will simulate a spring-damper system (as long as the force stays below the limit). We see that there is a great similarity between PSMC with a PD-type virtual coupling and impedance control (Hogan, 1985) without an inertial component.

If the virtual coupling is of the PID-type, the integral component causes the resisting force to have its own dynamics, it can no longer simply be calculated from current position and velocity errors. The integral component can be useful when modeling errors are present. Consider applying the PSMC controller described above to the 2-DOF pneumatic manipulator. Since it operates in the vertical

plane, eq. (5.30) has to be complemented with a gravity compensation term:

$$\boldsymbol{\tau}[k] = \boldsymbol{\tau}_{gc}[k] + \mathbf{J}^T(\mathbf{q}[k]) \mathbf{f}[k].$$

Assuming the desired position is fixed ($\dot{\mathbf{r}}_d = 0$), a PD-type virtual coupling will allow for a steady state position error if the gravity compensation $\boldsymbol{\tau}_{gc}[k]$ isn't perfect. If the coupling has an integral part, the force exerted by the coupling will vary in such a way that it pulls the end effector towards the proxy. As long as this force doesn't exceed F (in norm), the position error will go to zero. Otherwise, the proxy will leave its sliding surface and the position error persists.

Modeling errors can influence the compliant behavior of the controller. To see this, let's assume that errors in $\boldsymbol{\tau}_{gc}[k]$ cause the end-effector to be vertically below its (fixed) desired position. Due to the position error, the virtual coupling exerts a force on the end-effector, pulling it upwards. The integral term causes this force to increase to a value $F_i < F$ until the desired position is reached. Now imagine that an external force pushes the end-effector down. Since the gravity compensation is not perfect, a portion of the force "available" to provide compliance to the external influence is already used to eliminate the steady state error. In this case, the resisting force would saturate at $F - F_i$ for deviations higher than $(F - F_i)/K_p$ (assuming the influence of the gravity compensation error doesn't change as the end-effector is moved).

5.2.6.3 Tracking

In order to describe the controller's tracking behavior, we'll distinguish between two modes, normal tracking and response to large position errors.

The manipulator is considered to be in normal tracking mode if the proxy is on the sliding surface, and if its position and velocity are "close" to the desired values. Since "close" is inherently subjective, we'll define it as meaning that the relative errors between actual end desired values for the proxy's position and velocity are under 5%. The proxy's trajectory converges to the desired one with time constant λ according to the first-order dynamics $\dot{\mathbf{s}} = 0$ (with \mathbf{s} defined in (5.5)). This implies that the position and velocity errors will drop below 5% after the proxy has been on its sliding surface for a period of time longer than 3λ (since $e^{-3\lambda/\lambda} = e^{-3} \approx 0.05$). Thus, we'll consider the controller to be in normal tracking mode after \mathbf{s} has been zero for a time of at least 3λ .

If the proxy is not on its sliding surface, or hasn't been on it for longer than 3λ , the manipulator is considered to be in the response to large position errors mode.

Normal tracking mode When the controller is in normal tracking mode the proxy isn't influenced by the manipulator, since the interaction force stays under the limit F . It tracks the desired trajectory with ever decreasing error. Since the

end-effector is connected to the proxy by a PID-type virtual coupling, it behaves as if it were PID-controlled (in task space) to follow the desired trajectory.

Response to large position errors Assume the manipulator of fig. 5.2 to be in a state of static equilibrium, with both its end-effector and the proxy coinciding with the (constant) desired position. If a human operator pushes the end-effector far enough away from the desired position such that the resisting force saturates at F , the proxy is pulled away from the desired position. The presence of the resisting force F can be seen as the sliding mode part of the controller exerting its maximum force in order to counteract the position error.

When the operator suddenly releases the end-effector, the proxy starts moving towards the goal position, dragging the end-effector behind it. When it reaches the sliding surface, it starts an exponential convergence (with time constant λ) to the desired trajectory (or position), and will come “close” to it after a time of around 3λ . The end-effector follows because of the PID-type coupling.

As in ordinary sliding mode control (see section 4.2.2), the exponential convergence increases safety when using PSMC: large position errors cause a slow convergence to the desired position. The convergence rate is set by the parameter λ . The slow response doesn’t compromise tracking performance in normal operation, however, since that is (mainly) determined by the parameters of the PID-type virtual coupling.

5.2.6.4 Relation with traditional controllers

By setting $\lambda = 0$ and $F \rightarrow \infty$, the controller becomes equivalent to a discrete-time PID controller (Kikuuwe and Fujimoto, 2006). With $K_i = 0$ and $\lambda = K_d/K_p$ they can be seen as force-limited PID control, or as sliding mode control with a boundary layer (Kikuuwe and Fujimoto, 2006). Proxy-based sliding mode control can thus be seen as an extension of these conventional methods.

Proxy-based sliding mode control can be considered to consist of two parts: a PID controller that makes the end-effector track the proxy position, and a second part that determines the proxy position in such a way that a smooth convergence to the desired trajectory is achieved without exceeding the force limit F . This is similar to the concept of reference governors (see for instance Kapasouris et al. (1988); Bemporad and Mosca (1994)). The separation in two parts is shown explicitly in fig. 5.5, which is equivalent to fig. 5.3 (cf. eq. (5.15)).

The main advantage of the method is the separation of “local” and “global” dynamics. The local dynamics, i.e. the response to small positional errors, is determined by the virtual coupling (parameters K_p , K_i and K_d), while the global dynamics (response to large positional errors) is determined by the sliding mode parameter λ . Thus, it is possible to combine responsive and accurate tracking during normal operation with smooth, slow and safe recovery from large position errors that can

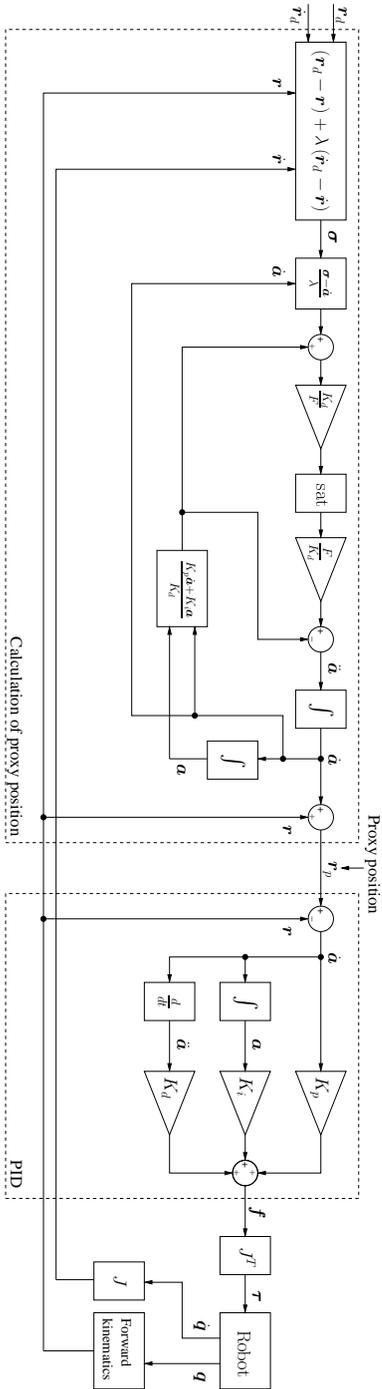


Figure 5.5: Proxy-based sliding mode controller (in continuous time) as a two-part controller, similar to the reference governor approach.

occur after abnormal events.

It is important to note that the discontinuous $\text{sgn}(\cdot)$ function does not appear in the controller equations (5.26)-(5.29). The output is continuous, so the controller does not induce chattering in the proxy's motion.

5.2.6.5 Summary

As long as the actuator force magnitude limit F is large enough to overcome disturbances, the controller will drive the proxy to its sliding surface. Once on the sliding surface, it will exhibit a smooth, exponential convergence (determined by the time constant λ) to the target trajectory. With a properly tuned PID-type virtual coupling, the end effector will follow the proxy closely.

If an external disturbance force larger than F is applied to the end effector (because of a contact with the environment or a human operator, for instance), the virtual coupling will transfer this force to the proxy, pulling it off its sliding surface (since the controller cannot compensate forces larger than F , see eq. (5.4)). The end effector, following the proxy by means of the virtual coupling, thus reacts compliantly to the disturbance, by being pushed out of position while resisting with a force of magnitude F . Since F is the magnitude of the maximum interaction force with the environment, its value can be seen as a compromise between robustness against disturbances and safety for humans (or the environment) in contact with the robot.

5.3 Controllers

The remaining part of this chapter focuses on the implementation and control performance of PSMC, which has been implemented in two different versions. In order to have a reference for performance comparison, PID control with gravity compensation has been implemented as well.

This section gives a description of the three controllers (one PID- and two PSMC-based) that were implemented.

5.3.1 PID control

In the case of the PID controller with gravity compensation, both links are controlled by the following control law:

$$\Delta p = \Delta p_{gc} + \Delta p_{PID},$$

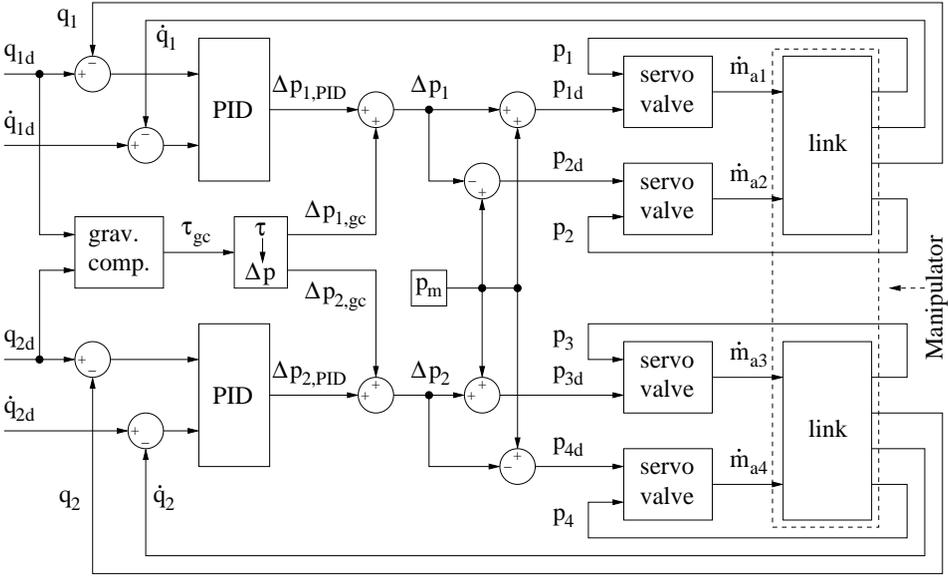


Figure 5.6: Schematic representation of the PID controller. p_{jd} represents the desired muscle pressure in muscle j , and \dot{m}_{aj} the air flow rate between servo valve j and its associated pneumatic muscle.

where Δp_{gc} provides static gravity compensation (calculated using eqs. (3.18) and (3.19)), and Δp_{PID} equals

$$\Delta p_{PID} = K_p e + K_d \dot{e} + K_i \int_0^t e d\tau.$$

In this expression, $e = q_d - q$, with q_d the desired joint angle and q the measured joint angle.

Fig. 5.6 shows a schematic representation of the PID-based control system.

5.3.2 Proxy-based sliding mode control – Task space implementation

The task space implementation is the proxy-based sliding mode controller as discussed in section 5.2, but with the addition of a gravity compensation term:

$$\tau[k] = \tau_{gc}[k] + \tau_{psmc}[k].$$

$\boldsymbol{\tau}_{gc}[k]$ is the torque needed for static gravity compensation, while $\boldsymbol{\tau}_{psmc}[k]$ is the torque calculated by the Proxy-Based Sliding Mode controller (cf. eq. (5.30)):

$$\boldsymbol{\tau}_{psmc}[k] = J^T(\mathbf{q}[k]) \mathbf{f}[k],$$

with $\mathbf{f}[k]$ calculated from equations (5.26)-(5.29).

In order to apply this torque, we have to know the corresponding actuator gauge pressures. We can calculate them by rewriting (2.17) in view of the Δp - approach introduced in section 3.3.1 (with $p_{2i-1} = p_m + \Delta p_i$ and $p_{2i} = p_m - \Delta p_i$ for joint index i):

$$\boldsymbol{\tau} = \begin{bmatrix} p_m(m_{\tau_1}(q_1) + m_{\tau_2}(q_1)) + \Delta p_1(m_{\tau_1}(q_1) - m_{\tau_2}(q_1)) \\ p_m(m_{\tau_3}(q_2) + m_{\tau_4}(q_2)) + \Delta p_2(m_{\tau_3}(q_2) - m_{\tau_4}(q_2)) \end{bmatrix}$$

so we have

$$\Delta p_1[k] = \frac{\tau_1[k] - p_m(m_{\tau_1}(q_1[k]) + m_{\tau_2}(q_1[k]))}{m_{\tau_1}(q_1[k]) - m_{\tau_2}(q_1[k])} \quad (5.33)$$

$$\Delta p_2[k] = \frac{\tau_2[k] - p_m(m_{\tau_3}(q_2[k]) + m_{\tau_4}(q_2[k]))}{m_{\tau_3}(q_2[k]) - m_{\tau_4}(q_2[k])} \quad (5.34)$$

with $\boldsymbol{\tau}[k] = [\tau_1[k] \quad \tau_2[k]]^T$.

Fig. 5.7 shows a schematic representation of the Task-Space Proxy-Based Sliding Mode control system. Eqs. (5.33) and (5.34) are represented in the diagram by the block marked with “ $\tau \rightarrow \Delta p$ ”.

5.3.3 Proxy-based sliding mode control – Joint space implementation

The tracking performance obtained using the previously described PSMC controller is not entirely satisfactory, as will be discussed in section 5.3.4. This can mainly be attributed to eqs. (5.33) and (5.34). They perform the conversion of the desired actuator torques to muscle gauge pressures by means of the muscle torque functions (which were introduced in section 2.3.2.5, expressions for the m_{τ_i} are given in appendix A), but this conversion is not very accurate.

Since the outputs of almost all controllers described in the robotics literature are desired actuator torques, this is a typical problem that is often encountered when trying to use a robotics-based control method on a pneumatic muscle system.

As the task-space implementation of PSMC calculates a force that has to be converted to torques using eq. (5.3), there is no way to avoid the torque to pressure conversion in task space. That’s why it was decided to implement a joint-space version of PSMC, where both links are controlled separately. This can be made far less sensitive to model inaccuracies.

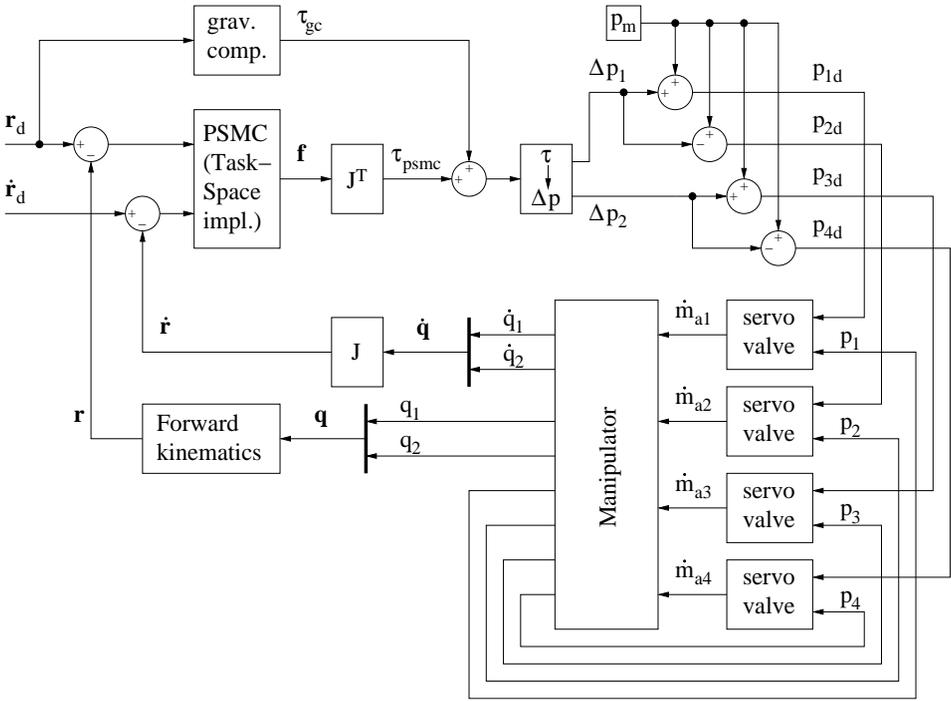


Figure 5.7: Schematic representation of the Task-Space Proxy-Based Sliding Model controller.

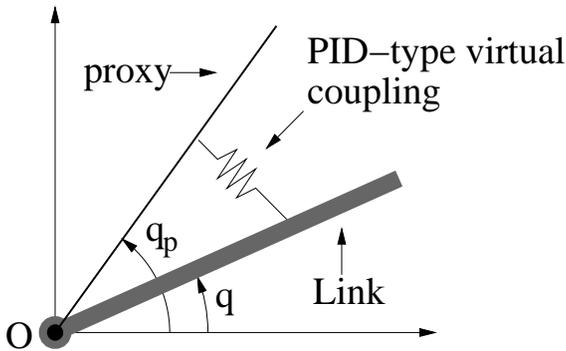


Figure 5.8: Proxy-Based Sliding Mode Control - Joint-space implementation.

Consider a single robot link in the horizontal plane, as shown in fig. 5.8. In this case, we consider the proxy to be a virtual link attached to the real link by means of a torsional PID-type virtual coupling that exerts a torque τ_c . Similar to task-space PSMC, we consider the proxy to be controlled by a sliding mode controller that exerts a torque τ_a . The proxy's equation of motion is then given by

$$I\ddot{q}_p = \tau_a - \tau_c$$

with I the proxy's moment of inertia (about an axis perpendicular to the link's plane of rotation and through point O (cf. fig. 5.8)). By analogy with eqs. (5.4) and (5.6), τ_a and τ_c are given by

$$\begin{aligned}\tau_a &= \tau_{lm} \operatorname{sgn}((q_d - q_p) + \lambda(\dot{q}_d - \dot{q}_p)) \\ \tau_c &= K_p(q_p - q) + K_i \int (q_p - q) dt + K_d(\dot{q}_p - \dot{q}).\end{aligned}$$

By setting

$$\begin{aligned}a &= \int (q_p - q) dt, \\ \sigma &= (q_d - q) + \lambda(\dot{q}_d - \dot{q})\end{aligned}$$

and $I = 0$ (the equivalent of having the massless proxy in task-space PSMC) we arrive at

$$\tau = \tau_{lm} \operatorname{sgn}(\sigma - \dot{a} - \lambda\ddot{a}) \quad (5.35)$$

$$\tau = K_p\dot{a} + K_i a + K_d\ddot{a}, \quad (5.36)$$

exactly as in section 5.2. Since equations (5.35)-(5.36) are simply a one-dimensional form of (5.14)-(5.15) (although angles are used instead of Cartesian coordinates), we can essentially reuse the discrete-time solution (5.26)-(5.29).

So far, this doesn't seem to solve the problem, though, the outputs of the controllers³ are still torques that needs to be converted to muscle pressures. The difference with the task-space implementation is that the torques are directly calculated by the PSMC controllers, not by means of eq. (5.3). Each controller calculates the desired torque to be applied by the muscles in its associated joint. Since this torque is applied by setting an appropriate value of Δp , it is also possible to have the controller output the required value of Δp directly. All that is required is a change of units for the gains in the virtual coupling (for instance, instead of expressing K_p in Nm/rad, we express it in bar/rad).

The Proxy-Based Sliding mode control law now becomes

$$\sigma[k] = (q_d[k] - q[k]) + \lambda(\dot{q}_d[k] - \dot{q}[k]) \quad (5.37)$$

³Since there are two individually controlled links, we now have two separate PSMC controllers.

$$\begin{aligned} \Delta p_{psmc}^* [k] &= \frac{K_d + K_p T + K_i T^2}{\lambda + T} \sigma [k] + K_i a [k - 1] \\ &+ \frac{(K_p + K_i T) \lambda - K_d}{(\lambda + T) T} \nabla a [k - 1] \end{aligned} \quad (5.38)$$

$$\Delta p_{psmc} [k] = \begin{cases} \Delta p_{psmc}^* [k] & \text{if } |\Delta p_{psmc}^* [k]| \leq \Delta p_{lm} \\ \Delta p_{lm} \operatorname{sgn} (\Delta p_{psmc}^* [k]) & \text{if } |\Delta p_{psmc}^* [k]| > \Delta p_{lm} \end{cases} \quad (5.39)$$

$$\begin{aligned} a [k] &= \frac{1}{K_d + K_p T + K_i T^2} ((K_d + K_p T) a [k - 1] \\ &+ K_d \nabla a [k - 1] + T^2 \Delta p_{psmc}(k)). \end{aligned} \quad (5.40)$$

Δp_{lm} is a pressure limit similar to the force limit F in equations (5.26)-(5.29).

Having the controllers calculate values of Δp directly effectively bypasses the torque to pressure conversion. Of course, this means that the controller has to be able to cope with the non-linearity of the system. Provided that a gravity compensating feedforward term is used, this didn't turn out to be a problem.

Including the gravity compensation, the control law becomes

$$\Delta p [k] = \Delta p_{gc} [k] + \Delta p_{psmc} [k], \quad (5.41)$$

with $\Delta p_{psmc} [k]$ calculated from (5.37)-(5.40) and $\Delta p_{gc} [k]$ the Δp value necessary for static gravity compensation (which is given by (3.18) and (3.19) for joints 1 and 2, respectively). Although the accuracy of $\Delta p_{gc} [k]$ depends on how accurately the manipulator and actuators were modeled, the proxy-based sliding mode term $\Delta p_{psmc} [k]$ is completely model independent (i.e. purely feedback based).

Both links are separately controlled using control law (5.41), which means they both have their own private proxy. A link's proxy will be pulled off its sliding surface when the Δp value corresponding to an external disturbance torque acting on the link exceeds Δp_{lm} . This causes the link to comply with the external torque, while resisting with the torque generated by the gauge pressure difference Δp_{lm} .

The fact that the "limit value" is a pressure (as opposed to a force in task-space PSMC) is a disadvantage of the joint-based PSMC implementation, because the relation between torque and pressure is complex and angle-dependent (see section 2.3.2.5). Thus, the maximum "resisting torque" to external influences is not a constant, but a function of the joint angle.

Fig. 5.9 shows a schematic representation of the Joint-Space Proxy-Based Sliding Mode control system. Its structure is very similar to the PID-based control system, as shown in fig. 5.6.

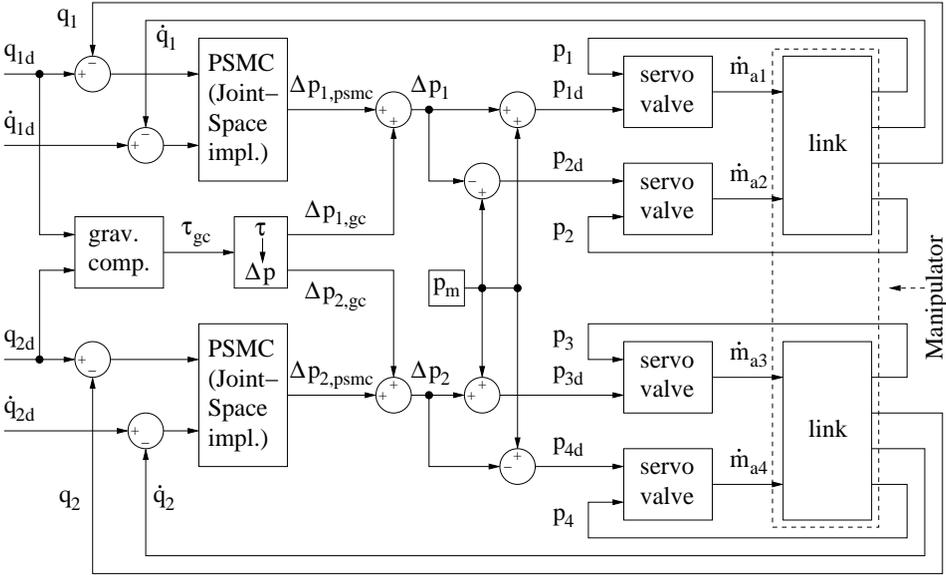


Figure 5.9: Schematic representation of the Joint-Space Proxy-Based Sliding Mode controller.

5.3.4 Experimental results

In order to evaluate control performance, a number of experiments were performed with the three described controllers. This section gives an overview of the tests that were performed and presents the results. They will be discussed in the next section.

Table 5.1 lists the values of the control parameters that were used.

The first experiment was tracking a circular trajectory (in task space) with a diameter of 20 cm. Fig. 5.10 shows both joint angles for a circle-period P of 10 seconds (10 seconds per revolution). The positional error $\|\mathbf{r}_d - \mathbf{r}\|$ (with \mathbf{r} the Cartesian position and \mathbf{r}_d the desired Cartesian position) is shown in fig. 5.11 for periods of 10 and 5 seconds.

In order to test the influence of the sliding mode parameter λ , the same experiment was performed with several values of the parameter. Fig. 5.12 shows the measurements for $\lambda = 0.4 s$ and $\lambda = 0.8 s$.

The Proxy-Based Sliding Mode controllers allow for very high integral gains. Results for the joint space implementation using $K_i = 12 bar/rad \cdot s$ in both joints are shown in fig. 5.13, and position errors are shown in fig. 5.14.

Table 5.2 provides a summary of tracking errors for the different controllers.

In a second experiment, the desired trajectory was discontinuously switched be-

Parameter	Value (link 1)	Value (link 2)
K_p (bar/rad)	2	1.5
K_i ($bar/rad \cdot s$)	2	2
K_d ($bar \cdot s/rad$)	0.1	0.1

(a) PID

Parameter	Value
λ (s)	0.4
K_p (N/m)	200
K_i (N/ms)	100
K_d (Ns/m)	10
F (N)	15

(b) PSMC - Task space implementation

Parameter	Value (link 1)	Value (link 2)
λ (s)	0.4	0.4
K_p (bar/rad)	2	1.5
K_i ($bar/rad \cdot s$)	2	2
K_d ($bar \cdot s/rad$)	0.1	0.1
Δp_{lm} (bar)	0.2	0.2

(c) PSMC - Joint space implementation

Table 5.1: Parameter values. Unless otherwise noted, the above parameter values were used in the experiments.

tween two circular paths. The original circular trajectory had a period of 3.333 seconds, while the second had a period of 10 seconds. Figure 5.15 shows the result. In fig. 5.16, data from the same experiment are shown, but for different values of λ .

We also tested the response to step inputs. An example is shown in figure 5.17, where the desired value for q_1 is switched between $30\pi/180$ and $70\pi/180$, while the desired value for q_2 is kept constant at $-80\pi/180$. The angular velocities recorded during the step are shown in fig. 5.18. Step responses in the case of the high integral gain are shown in fig. 5.19 for several values of λ .

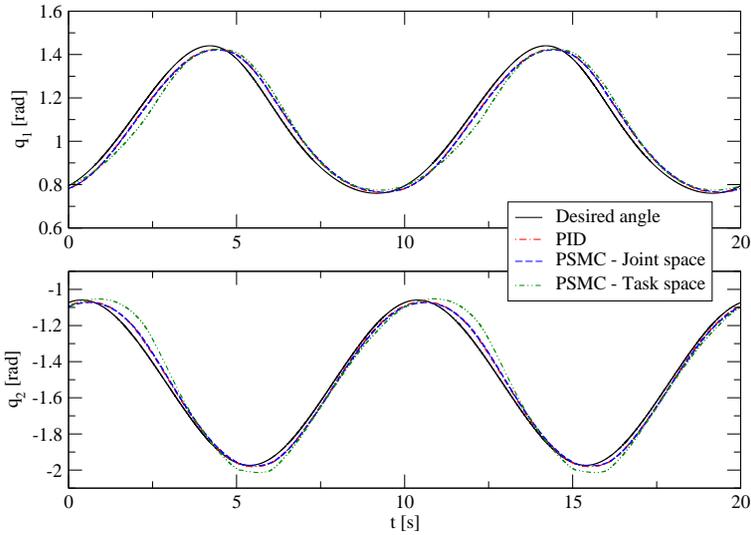


Figure 5.10: Measured and desired joint angles while tracking a circular trajectory (period = 10s).

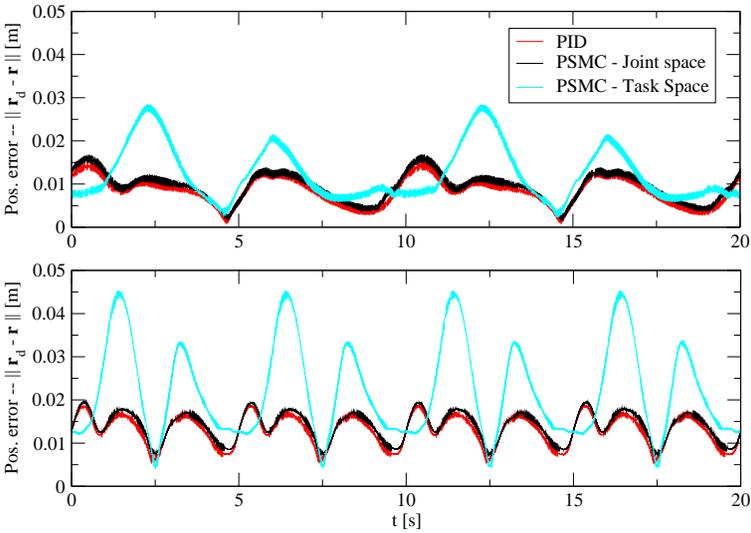


Figure 5.11: Position errors while tracking circular trajectories with a period of 10 seconds (top) and 5 seconds (bottom).

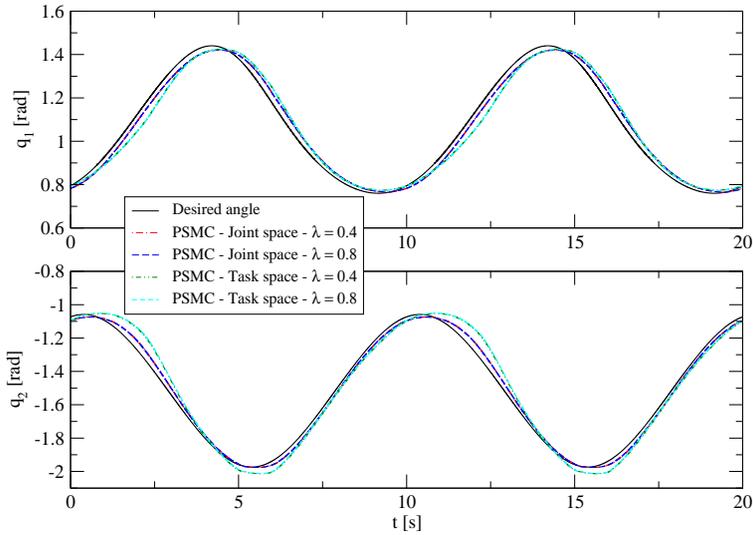


Figure 5.12: Measured and desired joint angles while tracking a circular trajectory (period = 10s). Two values of the sliding mode parameter value λ have been used for both Proxy-Based sliding mode controllers (in the joint space implementation, the same value of λ was used for both links).

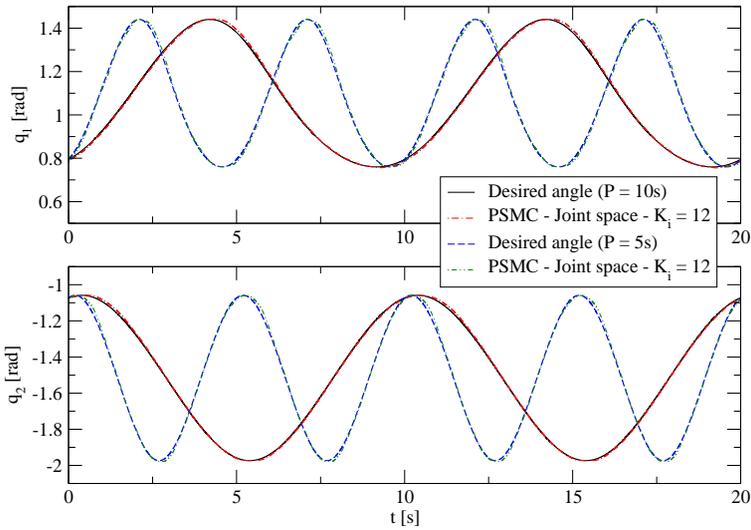


Figure 5.13: Measured and desired joint angles while tracking circular trajectories with periods of 5 and 10 seconds using the Joint space Proxy-Based Sliding Mode controller with high integral gain ($K_i = 12 \text{ bar/rad} \cdot \text{s}$).

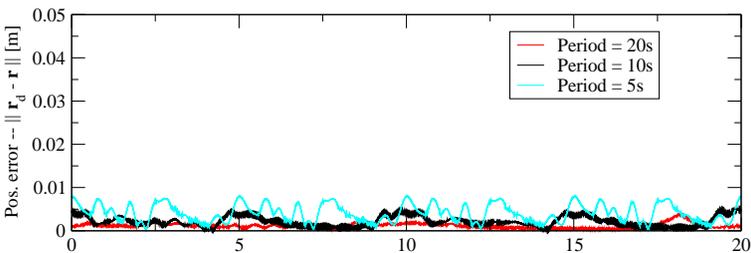


Figure 5.14: Position errors while tracking circular trajectories with periods of 5 and 10 seconds using the Joint space Proxy-Based Sliding Mode controller with high integral gain ($K_i = 12 \text{ bar/rad} \cdot \text{s}$). (the scale on this graph is the same as in fig. 5.11)

Controller	$P = 5\text{ s}$		$P = 10\text{ s}$		$P = 20\text{ s}$	
	Error (m)		Error (m)		Error (m)	
	Mean	Max	Mean	Max	Mean	Max
PIID	0.0130	0.0188	0.0084	0.0146	0.0048	0.0096
PSMC - Joint space - $\lambda = 0.4\text{ s}$, $K_i = 2\text{ bar/rad} \cdot \text{s}$	0.0140	0.0197	0.0094	0.0161	0.0054	0.0104
PSMC - Joint space - $\lambda = 0.8\text{ s}$, $K_i = 2\text{ bar/rad} \cdot \text{s}$	0.0138	0.0193	0.0094	0.0167	0.0055	0.0104
PSMC - Joint space - $\lambda = 1.5\text{ s}$, $K_i = 2\text{ bar/rad} \cdot \text{s}$	0.0136	0.0189	0.0095	0.0168	0.0058	0.0115
PSMC - Joint space - $\lambda = 0.4\text{ s}$, $K_i = 12\text{ bar/rad} \cdot \text{s}$	0.0041	0.0082	0.0022	0.0053	0.0011	0.0039
PSMC - Joint space - $\lambda = 0.8\text{ s}$, $K_i = 12\text{ bar/rad} \cdot \text{s}$	0.0038	0.0078	0.0022	0.0059	0.0014	0.0057
PSMC - Joint space - $\lambda = 1.5\text{ s}$, $K_i = 12\text{ bar/rad} \cdot \text{s}$	0.0037	0.0080	0.0021	0.0061	0.0016	0.0060
PSMC - Task space - $\lambda = 0.4\text{ s}$, $K_i = 2\text{ bar/rad} \cdot \text{s}$	0.0221	0.0447	0.0123	0.0277	0.0060	0.0144
PSMC - Task space - $\lambda = 0.8\text{ s}$, $K_i = 2\text{ bar/rad} \cdot \text{s}$	0.0222	0.0474	0.0125	0.0285	0.0061	0.0145
PSMC - Task space - $\lambda = 1.5\text{ s}$, $K_i = 2\text{ bar/rad} \cdot \text{s}$	0.0222	0.0517	0.0125	0.0295	0.0063	0.0149

Table 5.2: Mean and maximum of the position error $\|\mathbf{r}_d - \mathbf{r}\|$ for various controllers and circle-periods P . Each value was determined using a dataset representing 40 seconds of measurements.

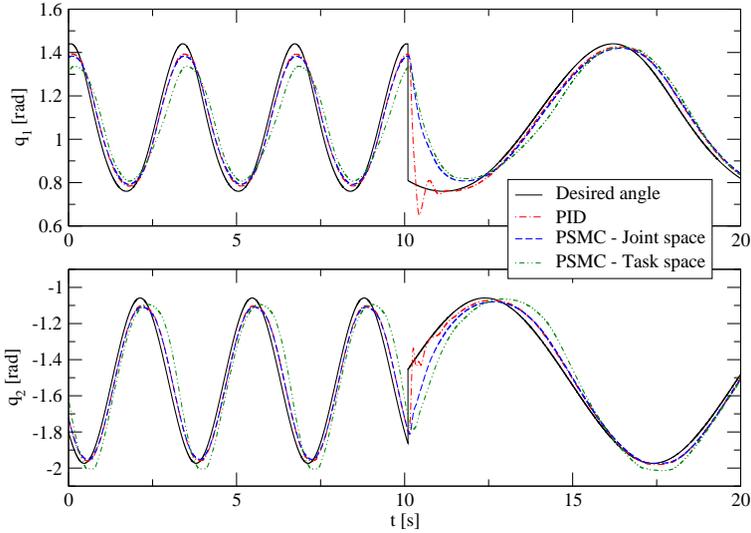


Figure 5.15: System response to a discontinuous change in desired trajectory.

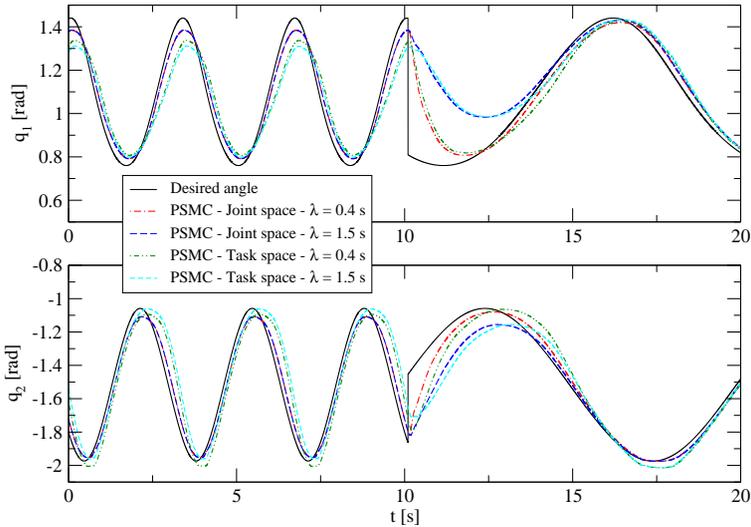


Figure 5.16: System response to a discontinuous change in desired trajectory. Two values of the sliding mode parameter value λ have been used for both Proxy-Based sliding mode controllers (in the joint space implementation, the same value of λ was used for both links).

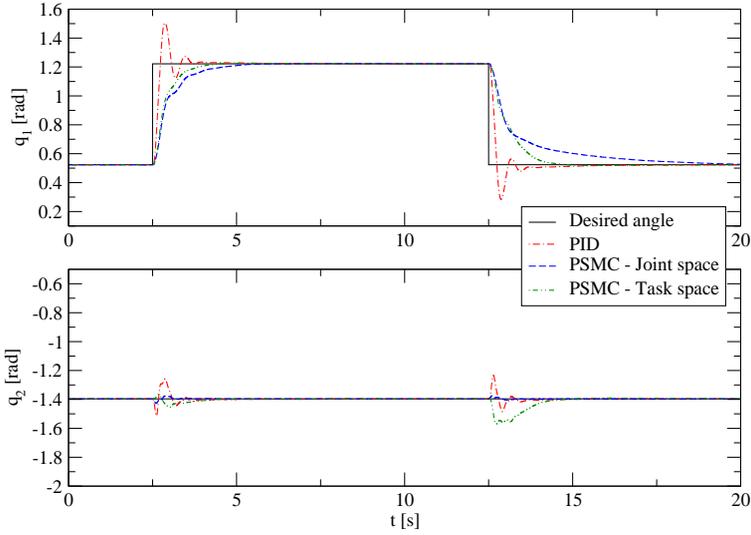


Figure 5.17: System response to a step change in desired angle values.

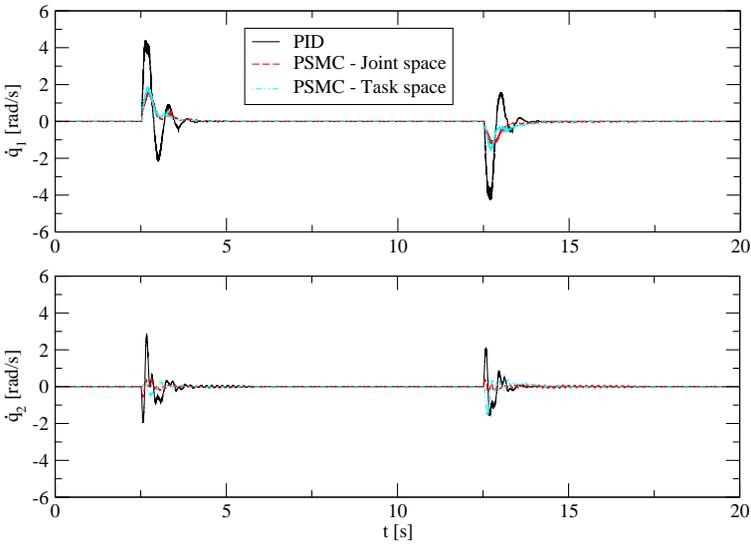


Figure 5.18: Angular velocities measured for the step response shown in fig. 5.17.

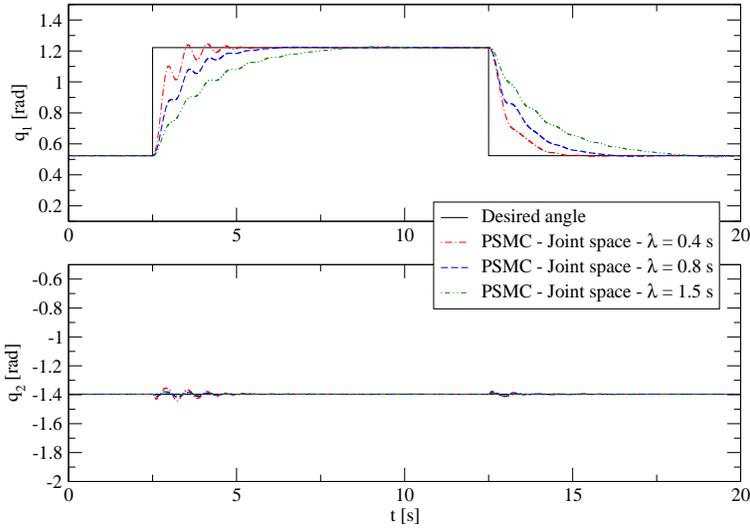


Figure 5.19: System response to a step change in desired angle values. The controller used was the joint-based implementation of PSMC, using high integral gain ($K_i = 12 \text{ bar}/\text{rad} \cdot \text{s}$).

5.3.5 Discussion

Due to the high flexibility of the joints, the system can quickly become unstable when increasing the gains of the PID controller. The gains were thus progressively increased (by hand), and cannot be set much higher than the values shown in table 5.1 without causing instability in certain regions of the workspace.

Since it is difficult to obtain significantly better tracking results by additional parameter tuning, the results obtained with the PID controller (and given gains) were used as a reference to compare the behavior of both Proxy-Based Sliding Mode controllers. In this section, only performance with respect to tracking accuracy is discussed. Safety aspects will be investigated in chapter 6.

It is clear from figures 5.10 and 5.11 that the task-space implementation of Proxy-Based Sliding Mode Control (PSMC) has the biggest tracking error of the three controllers. This was the case in all tracking experiments, as summarized in table 5.2.

As noted above, the main reason for the bad performance of the task-space implementation is the torque to pressure conversion (represented by the block marked with “ $\tau \rightarrow \Delta p$ ” in fig. 5.7), which depends on the muscle torque functions. This (nonlinear) conversion is not very accurate for several reasons:

- It depends on the PPAM parameters. These were estimated (cf. section

3.5.1.1) by means of a model that doesn't include hysteresis. Sections 3.5.1.6 and 3.6 have already shown, however, that hysteresis is not an unimportant effect in the controlled system.

- The torque to pressure conversion is static, i.e. it doesn't take into account any valve or pressure dynamics (cf. section 3.3).

As shown in figs. 5.9 and 5.6, in the joint space implementation and in the PID controller only the feedforward part (gravity compensation) involves the (inaccurate) torque to Δp conversion. The feedback part (PSMC or PID) calculates Δp values instead of torques, and thus interacts with the nonlinear muscle-valve dynamics directly. This results in much better tracking performance.

For ease of comparison, the gains used in the joint-space Proxy-Based Sliding Mode controller were the same as the ones used in the PID controller. As figures 5.10 and 5.11 show, both controllers behave almost identically, with the PID controller performing slightly better.

Due to its decoupling of local and global dynamics, however, the gains of the joint based PSMC controller can be chosen somewhat higher than in the PID controller. Especially the integral gain can be taken much higher without causing instability. Figs. 5.13 and 5.14 show that this greatly improves tracking performance. For $P = 10\text{ s}$, the average position error drops from 0.0094 m (with $K_i = 2\text{ bar/rad}\cdot\text{s}$ for both links) to 0.0022 m (with $K_i = 12\text{ bar/rad}\cdot\text{s}$). The price to pay is a slightly oscillatory step response, as shown in fig. 5.19.

Figures 5.15 and 5.17 illustrate the most important property of Proxy-Based Sliding Mode control: the smooth and slow response to sudden changes in desired position. Increasing the value of λ makes the response slower, as shown in fig. 5.16. In situations of normal tracking (i.e. when the actual position is close to the desired position (cf. section 5.2.6.3), and the proxy is on its sliding surface), the value of lambda hardly influences control performance, as fig. 5.12 and table 5.2 illustrate.

The parameter λ can thus be used to smoothen the system's response to unforeseen events, while still achieving good tracking performance in normal situations. Most traditional control methods cannot do both at the same time. In the case of PID control, for instance, lowering the gains may improve safety, but will hurt performance. Making the D-action very high would also (theoretically) limit joint velocity, but this is not feasible in practice due to noise in the velocity signals.

5.4 Interaction

As an application with interaction between a human and the system, a scenario was conceived where the user can grab the end-effector, move it while the robot follows, and leave it at a position of his choice within the operating area. This was

achieved by means of admittance control (using only a damping term) based on the joint-space implementation of PSMC.

5.4.1 Admittance control

Hogan (1985) divides physical (mechanical) systems in two classes – those that produce a position in response to an imposed force, called admittances, and those that produce a force in response to an imposed position, called impedances.

The goal of impedance control (Hogan, 1985) is to control the dynamic interaction between a robot and its environment. More specifically, one usually tries to achieve an end-effector impedance given by (for Cartesian coordinate i)

$$f_{e,i} = k_i \cdot (x_i - x_{0,i}) + b_i \cdot (\dot{x}_i - \dot{x}_{0,i}) + m_i \cdot (\ddot{x}_i - \ddot{x}_{0,i}). \quad (5.42)$$

In this equation, $f_{e,i}$ is the i -th component of the external force \mathbf{f}_e acting upon the end-effector, x_i , \dot{x}_i , \ddot{x}_i are the i -th components of the end-effector position, velocity and acceleration, respectively, and $x_{0,i}$, $\dot{x}_{0,i}$, $\ddot{x}_{0,i}$ represent the components of the non-contact position, velocity and acceleration (i.e. they represent the desired motion of the end-effector if no external force were present). The constants k_i , b_i and m_i represent the desired stiffness, damping and inertia, respectively. If we call $F_{e,i}(s) = \mathcal{L}\{f_{e,i}\}$ and $V_i(s) = \mathcal{L}\{\dot{x}_i - \dot{x}_{0,i}\}$, the above equation corresponds to the following mechanical impedance (Spong et al., 2006)

$$Z_i(s) = \frac{F_{e,i}(s)}{V_i(s)} = \frac{k_i}{s} + b_i + m_i \cdot s.$$

In many cases, the inertial part is omitted ($m = 0$), which means that the desired impedance equals that of a spring-damper system.

Conceptually, there are two ways in which the robot can be controlled to achieve the desired end-effector impedance (Lawrence, 1988):

- Either it is made to act as a force-source, where the desired force for the robot to exert on its environment is given by $-\mathbf{f}_e$, as can be calculated using eq. (5.42), based on sensing of the motion. The robot thus acts as an impedance, producing force in response to an imposed position.
- Or the robot is made to act as a position-source, with the components of the desired position vector determined by eq. (5.42), based on sensing of the external force \mathbf{f}_e . The robot thus acts as an admittance, producing a position in response to an imposed force. This is called position-based impedance control (Lawrence, 1988; Valency and Zacksenhouse, 2003) or admittance control (Villani and De Schutter, 2008).

In the case of the pneumatic manipulator considered in this work, admittance control is clearly easier to implement, since a position controller has already been

developed. It is also less model dependent, and is more suitable for systems where torque control is difficult, as highlighted by Heinrichs et al. (1997).

For the interaction application, only damping is needed, so the desired mechanical admittance becomes (using the same damping coefficient b for both x and y)

$$Z_i^{-1}(s) = \frac{V_i(s)}{F_{e,i}(s)} = \frac{1}{b},$$

where $i = x, y$. By introducing

$$P_i(s) = \mathcal{L}\{x_i - x_{0,i}\} = \frac{V_i(s)}{s}$$

this can be written as

$$P_i(s) = \frac{1}{b \cdot s} F_{e,i}(s),$$

or with $X_i(s) = \mathcal{L}\{x_i\}$ and $X_{0,i}(s) = \mathcal{L}\{x_{0,i}\}$

$$X_i(s) = \frac{1}{b \cdot s} F_{e,i}(s) + X_{0,i}(s).$$

In the time domain, this becomes

$$x_i(t) = \frac{1}{b} \int_0^t f_{e,i}(t') dt' + x_{0,i}(t). \quad (5.43)$$

We cannot expect to obtain an accurate representation of the desired admittance, however, since admittance control requires a stiff (high gain) position controller (Richardson et al., 2003), which PSMC is not (cf. section 5.2.6.2). Additionally, due to the limited control bandwidth, no controller would be able to make the system appear stiff, so the intrinsic compliance (which is not included in the desired admittance) can always be felt by the user.

5.4.2 Force estimation

In order to implement (5.43), we have to know the external force \mathbf{f}_e that acts on the end-effector. Since the end-effector isn't equipped with a force sensor, the force has to be estimated based on the measured actuator forces. In this section, two approaches are discussed to achieve this.

5.4.2.1 Recursive least-squares

A simple way to estimate the external force is to use recursive least squares estimation with exponential forgetting.

We start by adding a term to the dynamic model of the robot (including friction, as given by eq. (3.29)) to incorporate the influence of the external force (Spong et al., 2006):

$$H(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) + \boldsymbol{\tau}_f(\dot{\mathbf{q}}) = \boldsymbol{\tau} + J^T(\mathbf{q})\mathbf{f}_e.$$

In order to eliminate the acceleration $\ddot{\mathbf{q}}$, we use the filtering technique that was discussed in section 3.5.2.2. Writing $\langle x(t) \rangle_{F(s)}$ to indicate the signal that results from filtering time signal $x(t)$ using a filter with transfer function $F(s)$, this gives

$$\langle H(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) + \boldsymbol{\tau}_f(\dot{\mathbf{q}}) \rangle_{F(s)} = \langle \boldsymbol{\tau} \rangle_{F(s)} + \langle J^T(\mathbf{q})\mathbf{f}_e \rangle_{F(s)}.$$

Transfer function $F(s)$ is given in eq. (3.21). The left-hand side of the above equation can be written as $K_f(\mathbf{q}, \dot{\mathbf{q}})\boldsymbol{\theta}$ (as in eq. (3.27), but using the modified version of $W_3(\mathbf{q}, \dot{\mathbf{q}})$ given in section B.4.1 (due to the inclusion of friction)), where $\boldsymbol{\theta}$ is the vector of dynamic parameters that was estimated in section 3.5.2. Writing the filtered version of $\boldsymbol{\tau}$ as $\boldsymbol{\tau}_f$ (i.e. $\boldsymbol{\tau}_f = \langle \boldsymbol{\tau} \rangle_{F(s)}$) we have

$$\langle J^T(\mathbf{q})\mathbf{f}_e \rangle_{F(s)} = K_f(\mathbf{q}, \dot{\mathbf{q}})\boldsymbol{\theta} - \boldsymbol{\tau}_f.$$

If we assume that the external force \mathbf{f}_e varies slowly, we can consider it to be constant with respect to the low-pass filtering,

$$\langle J^T(\mathbf{q}) \rangle_{F(s)}\mathbf{f}_e = K_f(\mathbf{q}, \dot{\mathbf{q}})\boldsymbol{\theta} - \boldsymbol{\tau}_f.$$

By setting

$$A_f = \langle J^T(\mathbf{q}) \rangle_{F(s)} \quad (5.44)$$

and

$$\mathbf{y}_f = K_f(\mathbf{q}, \dot{\mathbf{q}})\boldsymbol{\theta} - \boldsymbol{\tau}_f, \quad (5.45)$$

we get

$$A_f\mathbf{f}_e = \mathbf{y}_f. \quad (5.46)$$

If N measurements are available, the weighted least-squares estimate of the external force is then given by (Ljung, 1999)

$$\hat{\mathbf{f}}_e = \arg \min_{\mathbf{f}_e} \sum_{k=1}^N \beta(N, k) \cdot (\mathbf{y}_f[k] - A_f[k]\mathbf{f}_e)^2, \quad (5.47)$$

where the numbers $\beta(N, k)$ are the weights. If the weights are chosen as

$$\beta(N, k) = \mu^{N-k}$$

with $0 \leq \mu \leq 1$ then data from the past will be exponentially discounted.

A recursive algorithm to find $\hat{\mathbf{f}}_e$ according to (5.47) is then given by (see for instance Ljung (1999))

$$\begin{aligned}\hat{\mathbf{f}}_e[k] &= \hat{\mathbf{f}}_e[k-1] + L[k] \left(\mathbf{y}_f[k] - A_f[k] \hat{\mathbf{f}}_e[k-1] \right) \\ L[k] &= P[k-1] A_f^T[k] (\mu I + A_f[k] P[k-1] A_f^T[k])^{-1} \\ P[k] &= \frac{1}{\mu} \left(P[k-1] \right. \\ &\quad \left. - P[k-1] A_f^T[k] (\mu I + A_f[k] P[k-1] A_f^T[k])^{-1} A_f[k] P[k-1] \right),\end{aligned}$$

with $L[k]$ and $P[k]$ 2 by 2 matrices.

Ljung (1999) calls

$$T_0 = \frac{1}{1 - \mu}$$

the “memory time constant” of the exponential forgetting algorithm, expressed in number of samples. Measurement taken more than T_0 timesteps ago contribute with a weight less than $e^{-1} \approx 36\%$ of that of the most recent measurement. We will assume that the external force remains nearly constant for periods of around 100 samples (or 0.1 s), which leads to a forgetting factor μ equal to 0.99.

Note that the recursive least-squares algorithm allows us to estimate \mathbf{f}_e without inversion of the inertia matrix $H(\mathbf{q})$, and without knowledge of the angular accelerations (due to the filtering).

5.4.2.2 Generalized momentum based observer

A second way to estimate the external force acting on the end-effector is based on a generalized momentum based disturbance observer introduced in De Luca and Mattone (2003, 2005); De Luca et al. (2006). It assumes the presence of a disturbance torque $\boldsymbol{\tau}_d$ in the joints (which we will assume to be produced by an interaction force at the end-effector). Including the disturbance torque, the dynamic equation (3.29) becomes

$$H(\mathbf{q}) \ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) + \boldsymbol{\tau}_f(\dot{\mathbf{q}}) = \boldsymbol{\tau} + \boldsymbol{\tau}_d. \quad (5.48)$$

The observer is based on the robot’s generalized momentum, which is defined as

$$\mathbf{p} = H(\mathbf{q}) \dot{\mathbf{q}}.$$

Its time derivative is then given by

$$\dot{\mathbf{p}} = H(\mathbf{q}) \ddot{\mathbf{q}} + \dot{H}(\mathbf{q}) \dot{\mathbf{q}}. \quad (5.49)$$

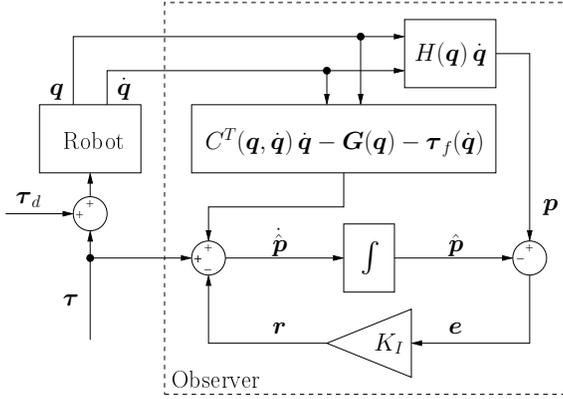


Figure 5.20: Schematic overview of the generalised momentum based observer.

Using eqs. (5.48) and (B.2) this can be written as

$$\begin{aligned}
 \dot{\mathbf{p}} &= \boldsymbol{\tau} - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{G}(\mathbf{q}) - \boldsymbol{\tau}_f(\dot{\mathbf{q}}) + \dot{\mathbf{H}}(\mathbf{q})\dot{\mathbf{q}} + \boldsymbol{\tau}_d \\
 &= \boldsymbol{\tau} + \left(\dot{\mathbf{H}}(\mathbf{q}) - \mathbf{C}(\mathbf{q}, \dot{\mathbf{q}}) \right) \dot{\mathbf{q}} - \mathbf{G}(\mathbf{q}) - \boldsymbol{\tau}_f(\dot{\mathbf{q}}) + \boldsymbol{\tau}_d \\
 &= \boldsymbol{\tau} + \mathbf{C}^T(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{G}(\mathbf{q}) - \boldsymbol{\tau}_f(\dot{\mathbf{q}}) + \boldsymbol{\tau}_d.
 \end{aligned} \tag{5.50}$$

This equation shows the advantage of using the generalized momentum: the time evolution of \mathbf{p} is decoupled with respect to the disturbance torque, the i -th component of $\dot{\mathbf{p}}$ only depends on the i -th component of $\boldsymbol{\tau}_d$.

Assume now that we want to design an observer for \mathbf{p} instead of for $\boldsymbol{\tau}_d$. We can then imitate the procedure used in linear systems: construct a model of (5.50) (ignoring the term $\boldsymbol{\tau}_d$ since it isn't measured) and apply the prediction error $\mathbf{e} = \mathbf{p} - \hat{\mathbf{p}}$ as an extra input (Friedland, 1996). Writing $\hat{\mathbf{p}}$ as the prediction of \mathbf{p} , the observer dynamics are then given by

$$\dot{\hat{\mathbf{p}}} = \boldsymbol{\tau} + \mathbf{C}^T(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} - \mathbf{G}(\mathbf{q}) - \boldsymbol{\tau}_f(\dot{\mathbf{q}}) + \mathbf{K}_I \mathbf{e}, \tag{5.51}$$

where \mathbf{K}_I is a diagonal gain matrix with positive gains. A schematic representation of the observer is shown in fig. 5.20.

Since in (5.51) perfect model and actuator torque knowledge was assumed, we see that the term $\mathbf{K}_I \mathbf{e}$ in (5.51) corresponds to $\boldsymbol{\tau}_d$ in (5.50). Writing $\mathbf{r} = \mathbf{K}_I \mathbf{e}$, we have (using (5.50) and (5.51))

$$\begin{aligned}
 \dot{\mathbf{r}} &= \mathbf{K}_I \left(\dot{\mathbf{p}} - \dot{\hat{\mathbf{p}}} \right) \\
 &= \mathbf{K}_I \boldsymbol{\tau}_d - \mathbf{K}_I \mathbf{r}.
 \end{aligned} \tag{5.52}$$

If we write the i -th diagonal elements of K_I as $K_{I,i}$, and with $R_i(s) = \mathcal{L}\{r_i\}$ and $T_{d,i}(s) = \mathcal{L}\{\tau_{d,i}\}$ the above can be written in the Laplace domain for each component,

$$R_i(s) = \frac{K_{I,i}}{s + K_{I,i}} T_{d,i}(s). \quad (5.53)$$

We see that \mathbf{r} is the result of passing the disturbance torque through a first-order low pass filter. Of course, this only holds for a perfect model. All modeling and measurement errors will be attributed to a (nonexistent) disturbance torque by the observer.

By integrating (5.51) with respect to time (assuming zero initial conditions for $\dot{\hat{\mathbf{p}}}$) we have

$$\begin{aligned} \mathbf{r} &= K_I(\mathbf{p} - \hat{\mathbf{p}}) \\ &= K_I \left(\mathbf{p} - \int_0^t (\boldsymbol{\tau} + C^T(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} - \mathbf{G}(\mathbf{q}) - \boldsymbol{\tau}_f(\dot{\mathbf{q}}) + \mathbf{r}) dt \right), \end{aligned} \quad (5.54)$$

which can be used to implement the disturbance observer. In order to use the estimated system parameters $\boldsymbol{\theta}$ (cf. section 3.5.2), the model dependent parts of the above equation were expressed in a form that is linear in the parameters,

$$\mathbf{r} = K_I \left(\mathbf{p} - \int_0^t (\boldsymbol{\tau} + W_4(\mathbf{q}, \dot{\mathbf{q}}) \boldsymbol{\theta} + \mathbf{r}) dt \right),$$

with

$$W_4(\mathbf{q}, \dot{\mathbf{q}}) \boldsymbol{\theta} = C^T(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} - \mathbf{G}(\mathbf{q}) - \boldsymbol{\tau}_f(\dot{\mathbf{q}}). \quad (5.55)$$

An expression for $W_4(\mathbf{q}, \dot{\mathbf{q}})$ can be found in section B.4.1.

If the estimation of the disturbance torque has to be very accurate, the gains in K_I should be chosen as high as possible. Since in practice the measured signals are noisy (especially the torque measurement), we'll use low gains to obtain a suitably low-pass filtered signal. The values $K_{I,i}$ on the diagonal of K_I were both chosen equal to $10\pi \text{ rad/s}$. The cutoff frequency (5 Hz) is then equal to the one used in the filter of section 3.5.2.2 (which is used to remove the dependency on accelerations in the dynamic equation).

Note that just as in the previous method, the observer doesn't require inversion of the inertia matrix, nor knowledge of the angular accelerations.

In the admittance control application, we assume the disturbance torque is generated by an external force \mathbf{f}_e in the end-effector. \mathbf{f}_e is then given by (Spong et al., 2006)

$$\mathbf{f}_e = (J^T(\mathbf{q}))^{-1} \mathbf{r}. \quad (5.56)$$

Since there are no singularities in the workspace $J^T(\mathbf{q})$ is always nonsingular.

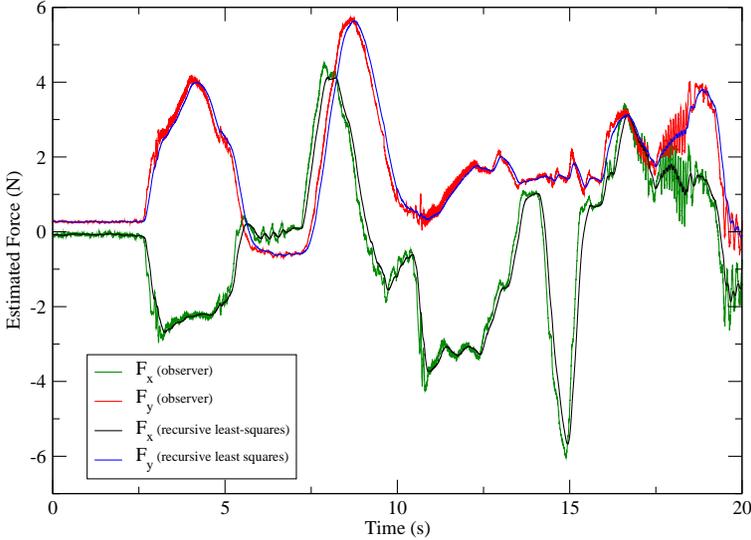


Figure 5.21: External force at the end-effector during interaction as estimated by both algorithms.

5.4.2.3 Comparison

To illustrate the difference between the force-estimation algorithms, the estimated forces (using both methods) during an interaction experiment are shown in fig. 5.21. It is clear that the results of both methods agree very well. The output of the observer is noisier than that of the recursive least-squares algorithm, which is smoother but also shows a time-lag with respect to the observer. Both smoothness and time-lag are caused by the exponential forgetting: since many measurements are taken into account, noise is averaged out, but it takes time for new data to influence the result.

Although both force estimation algorithms look very different, the good match between their results indicates that they have more in common than may be obvious at first sight. If we leave out the exponential forgetting and use only the last datapoint available, eq. (5.46) gives

$$\mathbf{f}_e = A_f^{-1} \mathbf{y}_f, \quad (5.57)$$

which we can compare to eq. (5.56) of the observer:

- \mathbf{y}_f corresponds exactly to \mathbf{r} : both are equal to the low-pass filtered disturbance torque. For \mathbf{r} , this was shown in eq. (5.53). For \mathbf{y}_f , it follows from its definition (eq. (5.45)) as the difference between the (filtered) torque prediction $K_f(\mathbf{q}, \dot{\mathbf{q}}) \boldsymbol{\theta}$ and the (filtered) measured torque $\boldsymbol{\tau}_f$.

We can see the fact that $\mathbf{y}_f = \mathbf{r}$ more explicitly by writing \mathbf{y}_f as

$$\mathbf{y}_f = \langle H(\mathbf{q}) \ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) + \boldsymbol{\tau}_f(\dot{\mathbf{q}}) \rangle_{F(s)} - \langle \boldsymbol{\tau} \rangle_{F(s)}.$$

With the choice made for K_I , eq. (5.53) implies $\mathbf{r} = \langle \boldsymbol{\tau}_d \rangle_{F(s)}$. From (5.50), (5.49) and (B.2) we then have

$$\begin{aligned} \mathbf{r} &= \langle \boldsymbol{\tau}_d \rangle_{F(s)} \\ &= \langle \dot{\mathbf{p}} - C^T(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) + \boldsymbol{\tau}_f(\dot{\mathbf{q}}) - \boldsymbol{\tau} \rangle_{F(s)} \\ &= \left\langle H(\mathbf{q}) \ddot{\mathbf{q}} + \dot{H}(\mathbf{q}) \dot{\mathbf{q}} - C^T(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) + \boldsymbol{\tau}_f(\dot{\mathbf{q}}) - \boldsymbol{\tau} \right\rangle_{F(s)} \\ &= \langle H(\mathbf{q}) \ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) + \boldsymbol{\tau}_f(\dot{\mathbf{q}}) \rangle_{F(s)} - \langle \boldsymbol{\tau} \rangle_{F(s)} \\ &= \mathbf{y}_f. \end{aligned}$$

- A_f in (5.57) corresponds to $J^T(\mathbf{q})$ in (5.56). Here both methods are slightly different, since A_f is defined in (5.44) as the low-pass filtered version of $J^T(\mathbf{q})$. If we would use a filtered version of $J^T(\mathbf{q})$ in (5.56) (which makes sense since \mathbf{r} is also the output of a filter) then the only difference between both methods would be that the least-squares algorithm uses multiple datapoints.

We see that the essential differences are the filtering of $J^T(\mathbf{q})$ in the least-squares algorithm, and of course the fact that the least-squares method uses data from multiple measurements.

Note that the calculation of \mathbf{r} using eq. (5.54) is much easier to implement than the direct calculation of \mathbf{y}_f using eqs. (5.45) and (3.27). Since both are equal, it is possible to calculate \mathbf{y}_f using (5.54) when using the recursive least-squares method.

5.4.3 Interaction experiment

The admittance control was implemented based on eq. (5.43). The joint-space PSMC was used as position controller.

Even in the absence of an external force, the output of the force estimation algorithms is never exactly zero. In order to prevent the estimation errors from continuously changing the desired position, a deadzone was placed on the estimated external force \mathbf{f}_e : if $\|\mathbf{f}_e\| \leq F_{min}$, \mathbf{f}_e was assumed to be zero. F_{min} was chosen equal to 3 N in the experiments. After some initial testing, the damping coefficient b was taken equal to 40 Ns/m.

The interaction experience is very difficult to quantify or represent in graphs, it has to be experienced. It was found that it was quite easy to move the end-effector around by applying force on it. When exerting low forces, the system would

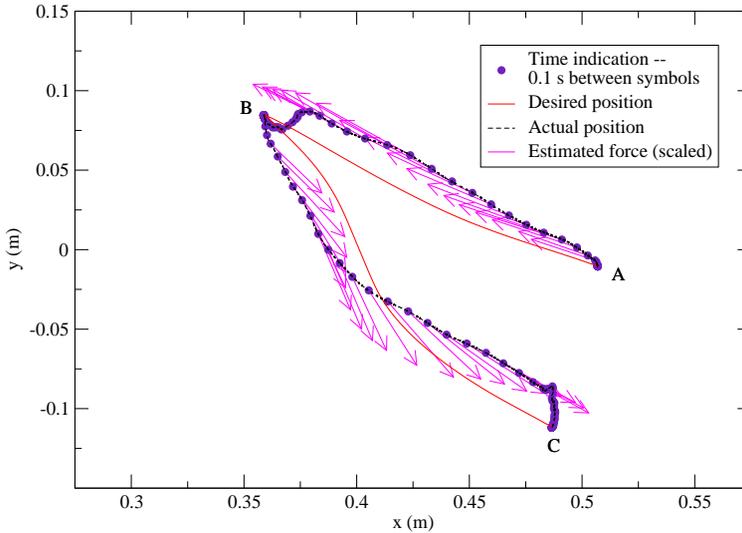


Figure 5.22: Actual and desired end-effector positions during an interaction experiment (manipulator under admittance control). The estimated forces have been scaled, an arrow with length 0.1 m (according to the figure’s axes) represents a force of 10 N .

sometimes react a bit jumpy. This happened when the norm of the estimated force alternated between being just over and just under the limit F_{min} .

Due to errors in the force estimation, the desired position doesn’t evolve in exactly the same direction as the applied force, causing a discrepancy between actual and desired position. If the end-effector is then suddenly released, it keeps moving (slightly) for a short time until it has reached the desired position.

No significant differences in interaction experience were noticed between the two force estimation algorithms, both performed well.

Fig. 5.22 shows results of an experiment where the end-effector was moved from point A to point B . In a second phase it was moved from B to C . The dots on the position trace, spaced 0.1 seconds apart, give an idea of the elapsed time. At each dot the estimated force at that time is shown if it exceeded the limit F_{min} . The effect of the deadzone is clearly visible near B (when the manipulator is moving towards C).

5.5 Conclusion

This chapter gave a thorough introduction to proxy-based sliding mode control Kikuwe and Fujimoto (2006). Its derivation and most important properties were discussed. Due to the separation of local and global dynamics, it can combine good tracking performance with a slow response to large position errors. It achieves this without chattering, which is a major advantage with respect to ordinary sliding mode control.

The “standard” PSMC (i.e. the task space implementation) did not provide very good tracking performance when applied to the pneumatic manipulator, mainly due to the highly model dependent torque to pressure conversion. For this reason, an adapted and much less model dependent version of PSMC, the joint-space implementation, was developed. It is very similar to PID control (both in structure and tracking performance), but can outperform PID since it allows for higher gains without causing instability. The most important reason for choosing PSMC is safety, however, not tracking performance. The safety aspects of PSMC will be discussed in the next chapter.

Based on admittance control and joint-space PSMC, a system was developed that allows an operator to manually move around and reposition the end-effector. In order to implement the admittance control, the interaction force between the user and the system has to be known. Since the system doesn’t have a force sensor at the end-effector, the interaction force has to be estimated. Two force estimation algorithms were discussed, one using recursive least squares with exponential forgetting and the other an observer based on generalized momentum. Although different at first glance, they proved to be very similar, and the interactive application worked well with both of them.

Chapter 6

Safety

6.1 Introduction

In situations where robots share their workspace with humans, and where physical human-robot interaction is possible or even necessary, safety is of paramount importance.

In the previous chapter, it was shown that proxy-based sliding mode control provides a gradual, smooth response to large position errors. Intuitively, this suggests that it makes the system safer, since in case of collision the impact velocity will be lower.

On the other hand, the manipulator is very lightweight and compliant. Since in the literature these properties are often considered to improve safety, maybe the arm is intrinsically “safe enough”. It is possible that it doesn’t need the extra safety provided by PSMC.

In this chapter, we investigate this issue quantitatively. By means of impact simulations with initial conditions taken from measured data, we investigate the safety of the system when using PID and PSMC control. These simulations point out if the system is intrinsically safe enough, or if it is necessary to use a “safe” controller such as PSMC.

The chapter starts with a brief overview of robot safety, specifically the quantification of safety and how safety aspects influence design and control. It proceeds with a discussion about the safety of the manipulator, detailing how the impact simulations were done and what can be learned from the results. The influence of joint compliance is also discussed, as well as the limitations of the approach.

6.2 Robot safety

Most robots in use today operate behind secure barriers that keep people outside of the work envelope (Hägele et al., 2008; Dhillon et al., 2002). In applications that involve close contact or cooperation between humans and robots, this principle of “safety by segregation” is no longer useful (Bicchi et al., 2008; Pervez and Ryu, 2008). It is clear, however, that without concrete safety guarantees robots cannot be allowed to work in close proximity to humans. In this context, safety becomes more important than traditional robot performance criteria such as speed and accuracy. Combining safety and performance is an important challenge in the design of human friendly robotic systems.

Robot safety in general is very broad (see for instance De Santis et al. (2008)), and covers many aspects ranging from mechanical design over software reliability, compliant coverings and avoiding sharp edges to psychological issues (letting people know which safety features are in place, for instance). An important factor in practice is also the dependability (Laprie, 1985; Avivzienis et al., 2004) of all components and the system as a whole (i.e. it should be able to deal with sensor failure, actuator failure, software failure, etc.). In this work, we will only consider safety, not dependability.

6.2.1 Quantifying safety

In the context of physical human-robot interaction, safety is interpreted in terms of the injuries sustained by the human in case of a collision with the robot. In order to evaluate safety of existing robots, or to optimize robot design or control for safety, the concept has to be defined quantitatively.

Quite some work exists that involves quantitative measures of safety, danger, injury or pain in the context of human-robot collisions (Yamada et al., 1997; Lim and Tanie, 2000; Ikuta et al., 2003; Heinzmann and Zelinsky, 2003; Kulić and Croft, 2006; Wassink and Stramigioli, 2007), but no universally accepted method of quantifying safety exists today. One mostly relies on injury severity indices developed in the automobile industry (cf. EuroNCAP (2008); Haddadin et al. (2007b)). By far the most popular is the Head Injury Criteria or HIC (Versace, 1971), which was introduced to robotics in Zinn et al. (2002); Bicchi and Tonietti (2004). It is defined as

$$\text{HIC} = \max_{\Delta t} \left\{ \Delta t \left(\frac{1}{\Delta t} \int_{t_1}^{t_2} \|\ddot{\mathbf{r}}_H\| dt \right)^{2.5} \right\} \quad (6.1)$$

with $\Delta t = t_2 - t_1 \leq 36 \text{ ms}$. $\|\ddot{\mathbf{r}}_H\|$ is the magnitude of the head acceleration that results from the collision, and is measured in multiples of $g = 9.81 \text{ m/s}^2$.

Mappings exist to translate HIC values to the probability of sustaining an injury of a certain level, with the levels usually expressed in terms of the Abbreviated Injury Scale (AIS), (Haddadin et al., 2007a)). Recent research that considers impacts between robots and crash-test dummies (both experimental and in simulation) shows that the mapping of HIC to injury level used in the automobile industry cannot simply be applied in robotics (Haddadin et al., 2007b; Oberer and Schraft, 2007). The main reason is that human-robot impacts occur at much lower velocities than the ones typically encountered in the car industry. The HIC can still be used to compare levels of safety, though.

6.2.2 Design and control for robot safety

Haddadin et al. (2008b) have shown that industrial robots are considerably less dangerous than previously assumed in case of collisions without clamping (i.e. when no part of the body is being squeezed between the robot and a part of the environment (a wall, for instance)). More specifically, they state that “blunt head impacts without clamping at moderate¹ robot speed are, no matter how massive the robot is, very unlikely to be life-threatening”.

It is clear that there is still a long way to go between “unlikely to be life-threatening” and “suitable for physical human-robot interaction”. Robots that interact directly with people are designed in a different way than conventional (industrial) robots. The two main design criteria that influence safety are:

- Low weight – robots designed to be used in contact with humans typically have low inertia of the moving parts (which limits damage in case of collisions). Examples are the DLR-LWRIII lightweight arm (Hirzinger et al., 2002) and the Whole Arm Manipulator (WAM, Salisbury et al. (1988)).
- Passive compliance – compliant elements in the robot structure decouple the inertia of the impacting link and that of the rest of the robot (i.e. the other links and (in case of electrical actuation) the rotor inertia of the motors). This means that mainly the impacting link is felt, and only a fraction of the inertia of the rest of the robot.

A popular way to introduce compliance is by means of compliant actuators. An overview of compliant actuators is given in Van Ham et al. (2009).

It should be noted that passive compliance isn’t always the result of deliberately introducing elastic elements. It can also be a by-product of a lightweight design or the use of certain transmissions (e.g. harmonic drives or cable transmission) or sensors (e.g. torque sensors), as is the case for the LWRIII. When this is the case for lightweight robots not specifically designed for human interaction it is seen as a disadvantage since mechanical compliance (whether

¹Moderate can be interpreted here as up to 2 m/s.

it is introduced on purpose or not) generally degrades robot performance in the traditional sense (i.e. speed and accuracy). In the case of rigid links and flexible joints with constant stiffness, specialized controllers are available (see for instance Spong (1987); Tomei (1991); De Luca (2000); Albu-Schäffer et al. (2007)) to minimize performance loss.

Since it was shown that increasing joint compliance beyond the level naturally present in the LWR III doesn't diminish injury potential (Haddadin et al., 2007b), Albu-Schäffer et al. (2008) propose to use compliant joints as a way of protecting the robot against impacts (which may break gears, sensors etc.), much more than the human.

The most often used control strategy for physical human-robot interaction is probably impedance control (Hogan (1985), see also section 5.4.1) and its variants, but other methods exist as well. In general, robots are controlled to move relatively slowly when interacting with people, which reduces impact velocity in case of collision. References that discuss control in a safety context include Lim and Tanie (2000); Zollo et al. (2003); Kulić and Croft (2006); Formica et al. (2006); Kiku-uwé and Fujimoto (2006); Albu-Schäffer et al. (2007); Buerger and Hogan (2007); Haddadin et al. (2008a).

All control based safety methods are inherently limited by the available control, actuator and sensor bandwidth, though. In case of a sudden impact, they may not be able to respond fast enough, which means that the natural impedance of the robot will be felt. This can be dangerous for the human, but also for the robot (Haddadin et al., 2007b; Albu-Schäffer et al., 2008).

Integrated approaches that combine specialized hardware and control techniques to improve safety but maintain performance have been reported as well. Distributed Macro-Mini actuation (DM², Zinn et al. (2002, 2004b,a)) consists of partitioning the actuation into separate macro and mini actuators (used in parallel) that provide for low- and high frequency torque generation, respectively. The mini actuators are small motors on the joints, the macro actuators are series elastic actuators (Pratt and Williamson, 1995) placed at the base, at least in the initial concept. In more recent work pneumatic artificial muscles were used as macro actuators (Sardellitti et al., 2007; Shin et al., 2008).

Another integrated design/control approach is variable stiffness or more generally variable impedance actuation (VIA, Bicchi et al. (2001, 2003); Bicchi and Tonietti (2004); Filippini et al. (2008)). VIA uses fast and continuous changes of joint impedance to provide user safety at all times while maximizing (under the constraint of safety) control performance. An important result is the "Fast and soft" concept (Bicchi and Tonietti, 2004). It states that in order to guarantee a (chosen) maximum level of injury risk, joint compliance should be high when moving fast, and can be low when moving slowly.

6.3 Safety of the pneumatic manipulator

The pneumatic manipulator studied in this work fulfills the principal design requirements of a “safe” robotic system. With a total mass of around 2.5 kg for the moving parts it can be considered lightweight. It also has highly compliant joints due to the PPAM actuators, i.e. the system is passively compliant. In addition, there is active compliance (i.e. compliance by control) when using proxy-based sliding mode control (see section 5.2.6.2).

By using PSMC control, the response to large position errors can be made appropriately slow. This can be seen in fig. 5.15 (p. 131), which shows the response to a discontinuous change in trajectory, and in fig. 5.17 (p. 132), which shows the response to a step-input. In both figures, the response of the PSMC controllers is much slower than that of the PID controller.

The slow response to large errors limits the impact velocity in case of a collision, which increases safety. By changing the sliding mode parameter λ used in PSMC, it is possible to tune the response (make it slower or faster) as desired. This hardly affects normal tracking performance, as was noted in the previous chapter.

It is clear that with proper path planning, sudden changes in desired position or trajectory as in the examples above don’t occur. This doesn’t mean they are impossible, though. A typical example is when someone pushes the arm away from its desired position (which is possible due to its compliance), and then suddenly releases it. Under standard PID control, it would much more violently than with PSMC. Another possibility is a problem with the supply of compressed air (due to closing a wrong valve, for instance). Cutting the air supply is not in itself unsafe in this case: due to leaks and valve control actions, the actuators slowly lose pressure, causing the system to gently “relax” and the position error to increase. At repressurization, however, the large position error could cause a violent reaction.

Since PSMC is characterized by a slow response, it is expected to be safer than PID. In the remaining part of this section, we will quantitatively investigate the difference between both control methods with respect to safety by simulating a collision between a human head and the end-effector of the manipulator.

6.3.1 Contact model

In order to simulate a collision, it is necessary to model the contact between head and robot. One of the simplest models to describe the relation between the relative penetration of two bodies and the resulting contact force is the Kelvin-Voigt model, which consists of the parallel connection of a (linear) spring and a (linear) damper:

$$F_n = \begin{cases} k\delta + b\dot{\delta} & \delta \geq 0 \\ 0 & \delta < 0. \end{cases}$$

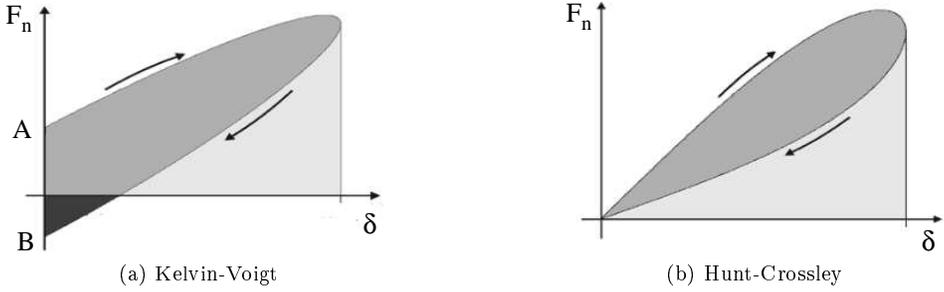


Figure 6.1: Hysteresis loops in the F_n - δ plane generated by a collision. Figures adapted from Diolaiti et al. (2005).

In this equation, F_n is the normal contact force between the bodies, and δ is the relative indentation. An impact generates a hysteresis loop in the F_n - δ plane, as shown in fig. 6.1a. Although popular because of its simplicity, the model has some physical inconsistencies (Gilardi and Sharf, 2002; Diolaiti et al., 2005). The most obvious ones are the discontinuity of the contact force at impact (point A), and the “sticky” negative force as the objects are separating (point B). Both inconsistencies arise at small penetration depths, since there F_n is mainly determined by the damping term.

By making the viscous damping dependent on the penetration depth, as proposed by Hunt and Crossley (1975), these problems can be overcome:

$$F_n = \begin{cases} k\delta^n + b\delta^n\dot{\delta} & \delta \geq 0 \\ 0 & \delta < 0. \end{cases} \quad (6.2)$$

The exponent n is usually close to 1, and takes into account the stiffness variation due to the fact that the contact surface area increases with increasing penetration depth.

Haddadin et al. (2007b) have estimated the parameters k , b and n of the Hunt-Crossley model (6.2) from impact experiments between the LWR-III lightweight arm and the head of a Hybrid III crashtest dummy. The impact characteristics of the Hybrid III’s head are comparable to those of the human frontal area (Haddadin et al., 2007a).

It is gratefully acknowledged that S. Haddadin provided the values of k , b and n that were estimated. These parameters are crucial to simulate realistic impacts, and hence to determine realistic safety characteristics.

6.3.2 Simulation

The collision simulations were performed using approximate models for both the manipulator and the head. The manipulator was modeled as an unactuated 2-DOF flexible joint arm with constant joint stiffnesses. The influence of gravity was not taken into account (i.e. the impact was assumed to happen in the horizontal plane). The initial values of the arm-model in the simulation, and the joint stiffnesses, were taken from data measured using the pneumatic manipulator. The mass of the head was chosen to be 5 kg, and it was assumed that the head's motion after the collision is unrestrained by either the environment or the rest of the body. This is a justified assumption, since it is reported that for short impacts the neck has little influence on the motion of the head immediately after impact (Oberer and Schraft, 2007; Willinger et al., 1999).

Specifically, in order to calculate the HIC value for a hypothetical collision at time t_0 using the data measured in an experiment, the following procedure was used:

- The initial position and angular velocity of the simulated model's links are set to the ones that were actually measured at t_0 .
- The joint stiffnesses in the model are set to the values that were present in the real system at time t_0 . They are calculated from the measured data using eq. (2.18), and are assumed to remain constant during the collision.
- The head is positioned so that it just touches the manipulator's tool-center point at t_0 , but there is no interaction force between them. In order to obtain the highest possible HIC value, the center of mass of the head is placed in the direction of the Cartesian velocity vector of the TCP at t_0 . In other words, the vector \mathbf{r}_{TCP-G} , that defines the location of the head's center of mass G at t_0 with respect to the TCP, is parallel to $\dot{\mathbf{r}}_{TCP}$, the Cartesian velocity of the TCP at t_0 . This is illustrated in fig. 6.2.

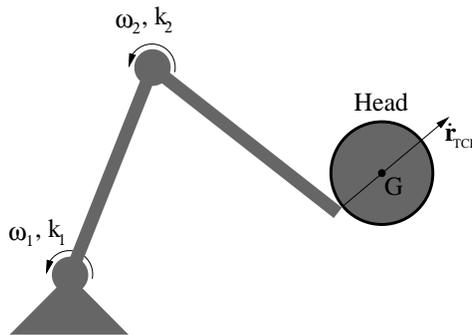


Figure 6.2: Position of the head at the start of the impact simulation.

- The motion of the whole system is simulated for a period of 50 *ms*. The head acceleration is then used to calculate the HIC using (6.1).

6.3.3 Results and discussion

In order to quantify the level of safety that is gained by using PSMC with respect to PID, HIC values were calculated for two situations: a step response (cf. fig. 5.17) and a discontinuous change in desired trajectory (cf. fig. 5.15). In both cases, the PID controller responds rather violently, whereas both proxy-based sliding mode controllers show a smoother response. This is illustrated in fig. 5.18, which shows the link angular velocities during the step response. It is thus expected that the HIC value in case of the PID controller will be higher (i.e. more dangerous) than for the other controllers.

Since an impact simulation and the maximization in (6.1) takes some computing time, we did not calculate the HIC (in case of collision) at each timestep of each measured dataset. Instead, the HIC was calculated at four different instants for each experiment: the time of maximum recorded Cartesian velocity of the TCP, the time of maximum recorded angular velocity of both links, and the time of the maximum recorded value of $\|\dot{\mathbf{q}}\|$, with $\dot{\mathbf{q}} = [\dot{q}_1 \quad \dot{q}_2]^T$. The highest obtained HIC value was retained as the HIC for that experiment. The results are presented in table 6.1.

The highest HIC value that appears in table 6.1 is 4.81. According to the standards of the automotive industry, where the HIC originated, this is very low. A value of 650, for instance, corresponds to a 5% risk of an injury classified as “serious” or worse on the Abbreviated Injury Scale (AIS) Haddadin et al. (2007b)).

It is not unusual to find low HIC values for collisions between robots and humans, even in undoubtedly dangerous cases, so the HIC may not be very well suited to evaluate robot safety Haddadin et al. (2007b); Oberer and Schraft (2007). It can still be used, however, to compare the safety of different controllers.

The data in table 6.1 show that the HIC values obtained with proxy-based sliding mode control are at least a factor of 10 lower than the ones obtained with PID control, and in most cases the differences are much higher. This is a clear indication that PSMC is effectively safer. We also see that the HIC decreases for increasing λ , as expected. Extremely low HICs can be obtained by a relatively high choice of λ . As shown in sections 5.3.4 and 5.3.5, this can be done without significantly hurting tracking performance.

Since the HIC, when applied to robotics, does not provide an “absolute” measure of danger, it is useful to consider the maximum interaction force that occurs during a robot – head collision. These forces, obtained by the same simulations as the HIC values and using the same Hunt-Crossley contact model, are also listed in table 6.1.

We see that the PID controller has a much higher maximum interaction force in

	Step		Switch between trajectories	
	HIC	F_{\max} [N]	HIC	F_{\max} [N]
PID	4.81	1524	3.02	1004
PSMC - Joint space - $\lambda = 0.4 s, K_i = 2 bar / rad \cdot s$	0.23	338	0.10	206
PSMC - Joint space - $\lambda = 0.8 s, K_i = 2 bar / rad \cdot s$	0.05	167	0.02	100
PSMC - Joint space - $\lambda = 1.5 s, K_i = 2 bar / rad \cdot s$	0.01	79	0.02	96
PSMC - Joint space - $\lambda = 0.4 s, K_i = 12 bar / rad \cdot s$	0.48	481	0.14	251
PSMC - Joint space - $\lambda = 0.8 s, K_i = 12 bar / rad \cdot s$	0.10	233	0.03	132
PSMC - Joint space - $\lambda = 1.5 s, K_i = 12 bar / rad \cdot s$	0.02	110	0.04	129
PSMC - Task space - $\lambda = 0.4 s, K_i = 2 bar / rad \cdot s$	0.29	375	0.03	117
PSMC - Task space - $\lambda = 0.8 s, K_i = 2 bar / rad \cdot s$	0.05	170	0.01	81
PSMC - Task space - $\lambda = 1.5 s, K_i = 2 bar / rad \cdot s$	0.01	82	0.01	80

Table 6.1: Values for the HIC and maximal interaction force F_{\max} obtained in several situations.

case of collision (hereafter denoted as F_{\max}) than the Proxy-Based Sliding Mode controllers. Again, increasing λ (for the Proxy-Based Sliding Mode controllers) lowers the “danger”, in this case F_{\max} .

The highest value for F_{\max} in the table is 1524 N , which occurred for the step response with the PID controller. This is a remarkably high force, especially when one considers the low weight of the manipulator (around 1.5 kg for the first link and just over 1 kg for the second) and high joint compliances (joint stiffnesses were $K_1 \approx 43\text{ Nm/rad}$ and $K_2 \approx 27\text{ Nm/rad}$ at the time of the simulated impact). The high value for F_{\max} is caused by the high angular velocities that occur during the step, as shown in fig. 5.18.

1524 N is enough to break several bones in the human face. The maxilla, the weakest bone in the face, has a fracture tolerance of 660 N (Haddadin et al., 2007a). The F_{\max} values for the Proxy-Based Sliding Mode controllers are significantly lower (i.e. safer), and all remain below the maxilla fracture tolerance.

We should note, however, that the used parameters in the Hunt-Crossley model (6.2) were estimated under assumption of a collision with the frontal bone. Other bones in the face will have different impact characteristics, so it is not entirely correct to simply transpose forces calculated for the frontal bone to other bones. Since the fracture tolerance of the maxilla is around 2.3 times lower than the maximal force, we can assume that even with a different impact model the resulting maximum force would remain higher than 660 N .

Since to date no accepted criterion exists to classify a robot as safe or unsafe, we cannot conclude that the pneumatic manipulator is safe when under PSMC control. It is clear, however, that we can consider the PID controller to be unsafe² in case of an unforeseen event (represented here by the step response). We also see that proxy-based sliding mode control significantly improves safety, and that the sliding mode parameter λ can be interpreted as a “safety parameter” (increasing λ will increase safety).

We can conclude that the hardware safety features incorporated in the system (low weight and high compliance) are not by themselves enough to ensure system safety.

6.3.4 Influence of joint stiffness

In order to investigate the influence of joint stiffness, the simulations for the step-response (which was found to be the most unsafe in the previous section) were repeated, but with modified joint stiffnesses. Instead of using the joint stiffness as predicted by eq. (2.18), both joints were set to the same stiffness k , and k was varied between 1 Nm/rad and 6000 Nm/rad . The upper limit of 6000 Nm/rad is in

²Evidently, there is no general criterion here either that strictly allows to declare this unsafe. We have taken the liberty, however, to consider the possibility of fractures of bones in the face as unsafe.

the range typically encountered in flexible joint robots (De Luca and Book, 2008). The resulting impact force is plotted as a function of k in fig. 6.3.

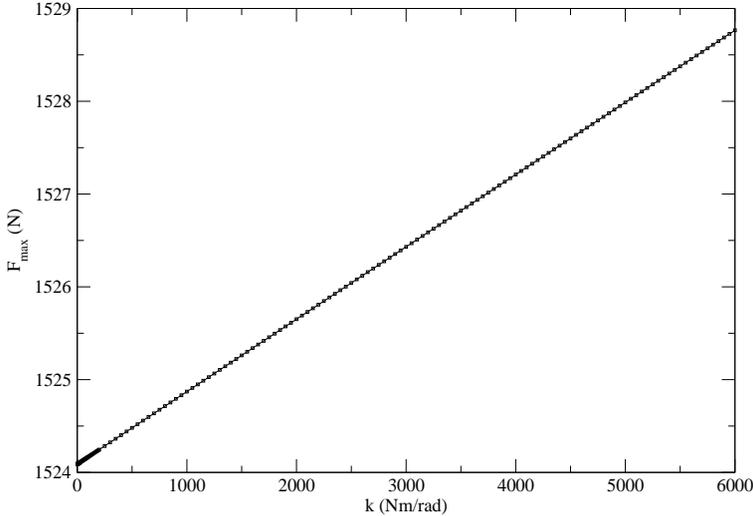


Figure 6.3: Maximum interaction force in case of a robot-head collision during the step response of fig. 5.17 as a function of joint stiffness k .

We see that F_{\max} hardly changes as the joint stiffness becomes higher: an increase of less than 5 N (or 0.33 %) for an increase of 5999 Nm/rad in joint stiffness. Even for the high range of stiffnesses (which could never be achieved with PPAMs), the impacting link is effectively decoupled from the rest of the system. The impact force is almost exclusively determined by the inertia of the impacting link. This is consistent with the findings of Haddadin et al. (2007b); Albu-Schäffer et al. (2008).

It seems that passive compliance plays a double role in robot safety. On the one hand, it can protect both human and robot in some impacts situations. This could be the case for instance if the robot is in normal operation and moving slowly, but collides with a fast moving human. On the other hand, the compliant elements can store energy, which (when released) can lead to higher speeds, and thus higher danger, than would be the case without compliant actuators. This “strikeout” feature is used to greatly increase the throwing distance of a ball by Wolf and Hirzinger (2008); Albu-Schäffer et al. (2008), and to increase the jumping height of a robot by Vanderborght et al. (2009). The effect is also present in the step response of the manipulator. We can see this by looking at figure 6.4, which shows a small part of fig. 5.18 (only the part corresponding to the upward step in q_1 , while the desired value of q_2 remains constant, cf. fig. 5.17).

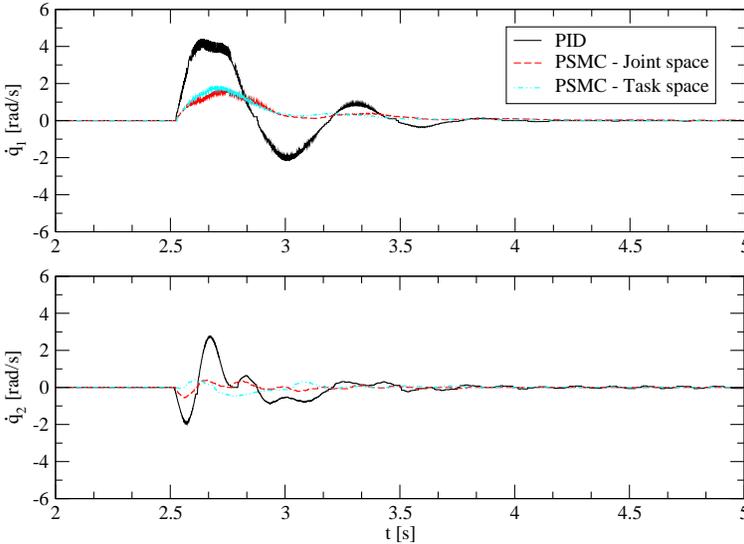


Figure 6.4: Angular velocities measured for the step response shown in fig. 5.17.

If we look at the PID-trace (in black), we see that as q_1 start to rise ($\dot{q}_1 > 0$), q_2 decreases ($\dot{q}_2 < 0$). In this phase, the “spring” of the second joint is being loaded. Soon after, \dot{q}_2 starts increasing, as the spring releases its energy³. The consequence is that both \dot{q}_1 and \dot{q}_2 are close to their maximum simultaneously for a short period of time, which results in a high end-effector velocity, and hence a high impact force in case of collision. The spring loading and strikeout is also visible with PSMC, but there the effect is much more moderate.

6.3.5 Limitations

It should be noted that discussion of safety given above has certain limitations:

- As in most robot safety studies, it only considers the initial impact (only the first 50 ms after impact were simulated, inspired by the definition of the HIC). Of course, overall safety will also depend on what happens after impact. It has been shown on the LWRIII that if the robot detects the impact and responds appropriately, contact forces can be significantly reduced (Haddadin et al., 2008a), for instance.
- Clamping was not considered, although it is more dangerous than an unconstrained collision (Haddadin et al., 2008c). Since the danger in clamping

³Of course, passive compliance isn't the only factor in this behavior, the controller of joint 2 also contributes as it tries to keep q_2 constant.

situations is highly dependent on robot mass, the pneumatic manipulator is expected to be on the safe side with its total mass of around 2.5 kg.

- Only blunt impacts were considered, no sharp edges or sharp objects attached to the robot. It is clear that these could greatly increase the danger of any robotic system.
- No hardware or software errors were considered. It was assumed that all sensors, actuators, controllers, electronics etc. are working.

Since it was shown that the system can be unsafe when under PID control, we can assume that it will also be unsafe in case of serious software errors in the controller.

6.4 Conclusion

In this chapter, impacts between the manipulator and a human head were simulated under some simplifying assumptions. The response of the head at the time of impact and immediately afterwards was calculated. Safety was evaluated by means of the Head Injury Criteria (HIC) and by means of the impact force. Since the mapping of HIC values to probability of injury level used in the automobile industry isn't suitable for robotics, and since at the moment no comparable mapping exists for robotics, the HIC values can only be used to compare levels of safety. Since the impact force quantifies safety in a more absolute way, it was found to be a more useful measure.

Both the HIC values and the impact forces show a significant increase of safety when using PSMC control, as compared to PID control. Based on the impact forces, we found that the system can be considered unsafe when under PID control, since collisions capable of breaking several bones of the face are possible. Using PSMC, all impact forces remain below the breaking strength of all bones in the head. By increasing the PSMC parameter λ , they can be kept far below this level. In the previous chapter, it was shown that this hardly affects tracking performance in normal operation.

The effect of joint stiffness on safety was investigated as well. The maximum impact force turned out to be almost independent of joint stiffness over a wide range of stiffnesses. This indicates that the impact is mainly determined by the inertia of the impacting link.

The fact that the manipulator is unsafe when under PID control (and can thus be assumed to be unsafe in the case of software errors in the controller) indicates that it is not intrinsically safe, in spite of its low weight and high compliance. We see that passive compliance can be regarded as a double edged sword with respect to robot safety. It can improve safety in some cases of sudden impact, but its ability to store and subsequently release energy can also make it more dangerous.

Chapter 7

Conclusion

This dissertation investigated the problem of safe and accurate control of a light-weight, two degree-of-freedom manipulator actuated by pleated pneumatic artificial muscles (PPAMs). The motivation for this work stems from the growing interest in human friendly robotics, a field driven by a whole range of new applications that involve close interaction between robots and humans. Safety is essential in these applications. However, more often than not, safety is at odds with performance in robotics.

Several controllers were investigated in this dissertation. The joint-space implementation of proxy-based sliding mode control (PSMC), presented in chapter 5, achieved the best control performance. As shown in chapter 6, it also provides a significant (and tunable) improvement in safety when compared to PID control. Although no accepted criterion exists by which we could declare the system (i.e. the manipulator under PSMC control) to be safe, we believe that PSMC is a strong candidate to help bridge the gap between safety and performance in robotics.

7.1 Overview

Due to the challenges involved in controlling compliant systems in general, and systems actuated by pneumatic muscles specifically, the performance of a controller cannot be evaluated in simulation. Therefore, a 2-DOF manipulator actuated by PPAMs was built to serve as a test platform. The properties of the PPAM actuator and the design of the pneumatic manipulator were discussed in chapter 2. A number of concepts used in other chapters were introduced as well, including compliance of a PPAM actuated joint and muscle torque functions.

Because good models can lead to better control, chapter 3 detailed how the manipulator, the actuators and the servo valves were modeled, and how the parameters present in both static and dynamic models were estimated. These models were

later used in control of the manipulator (chapters 4 and 5) and in the estimation of external force (chapter 5, section 5.4.2). To compensate for the lack of angular acceleration measurements, a filtering technique was used in the estimation of the dynamic mechanical model. The same technique was later applied in the force estimation of chapter 5.

Since the estimation experiments showed the presence of significant hysteresis in the manipulator, a Preisach-based hysteresis model for the PPAM was proposed. Since it requires a rather cumbersome initialization, it was not used to improve control.

Part II of the text, about control, started with an overview of previous work on control of pneumatic muscle systems, and of the challenges involved.

In chapter 4, an example was used to illustrate that sliding mode control might be interesting from a safety perspective. With a suitably chosen sliding surface, the system state starts converging exponentially to the desired state upon reaching the sliding surface. The time constant of this convergence can be chosen, which allows to limit the velocity of the system in case of response to a large position error.

Unfortunately, muscle-valve systems have slow dynamics, which cannot be ignored in sliding mode control. Therefore, a sliding mode controller based on a model that includes actuator dynamics was proposed. The control design required the complex and highly model dependent feedback linearization of the system. In spite of the use of boundary layers, the chattering problem typical for sliding mode control was quite severe, and tuning the controller was difficult and time-consuming. Due to these problems and the high complexity of the controller, it was decided to abandon sliding mode control.

Proxy-based sliding mode control (PSMC), which in spite of its name is very different from standard sliding mode control, was the subject of chapter 5. Inspired by sliding mode control and elements from haptics, it combines the properties of sliding mode control when responding to large position errors with those of PID control for small position errors. Task- and joint-space versions of PSMC were implemented, and their performance was compared to that of PID control. Especially the joint-space version of PSMC, which isn't as model dependent as the task-space version, achieved good tracking performance. Both PSMC controllers can respond slowly to large position errors, which is beneficial for safety.

Based on admittance control and the joint-space PSMC controller, an interactive mode was developed that allows the user to reposition the end-effector of the robot. For this application, the external force acting on the end-effector has to be known. Since the system is not equipped with a force sensor at the end-effector, the force had to be estimated. An estimation algorithm based on the filtered dynamic model of chapter 3 and on least-squares estimation with exponential forgetting was proposed, and compared with an observer based on generalized momentum. The two algorithms were found to be very similar.

Finally, chapter 6 tackled the subject of safety itself. The safety of the manipulator was investigated by simulating impacts between the manipulator and a human head. Safety was quantified by means of the head injury criteria (HIC) and by the maximum interaction force. Initial values for the simulations were taken from measured data, which allowed to quantify the safety of the manipulator when using different controllers. The simulations showed that the safety of PSMC is tunable, and that it can be made significantly safer than PID control. It was also shown that in spite of its low weight and high compliance, the manipulator is unsafe when under PID control. In case of a collision, the impact force can be high enough to break several facial bones.

Varying joint compliance in the simulations had almost no influence on the results, which indicates that the impact is mainly determined by the inertia of the impacting link.

7.2 General conclusions

Although this work was focused on the control of a manipulator actuated by pneumatic muscles, we believe it is possible to draw some conclusions that are valid for systems actuated by compliant actuators in general:

- Proxy-based sliding mode control effectively improves safety, while still achieving good control performance.
- Low weight and high compliance are not by themselves sufficient to obtain a safe system. This lack of inherent safety implies that controller software errors can have serious consequences.
- Passive compliance can be a double edged sword – in some situations, it improves safety, while in other situations its ability to store energy can make a robot more dangerous.
- As long as there is some form of passive joint compliance, the impact force in the case of a human-robot collision is mainly dependent on the inertia of the impacting link, and independent of the value of the compliance.

7.3 Future work

The work described in this dissertation can be continued in many ways. For control, some possibilities include:

- Implementing and testing proxy-based sliding mode control on other systems. In the near future, PSMC will be implemented on a system actuated by the MACCEPA actuator (Van Ham et al., 2007).

- Investigating the suitability of PSMC for rehabilitation robotics. This work has already started, see Beyl et al. (2009).
- Investigating the estimation of joint torques based on measured muscle pressures and a muscle hysteresis model (cf. section 3.6).

The work on safety in chapter 6 can still be significantly expanded. A good place to start would be to address some of the limitations summed up in section 6.3.5. Another important step to be taken is to perform actual impact tests, instead of simulating them.

Appendix A

Torque Calculations

In section 2.3.2 the procedure to calculate the torque functions $m_{\tau i}$ (with i the muscle's index as shown in fig 2.7) is explained. In this appendix, we present the actual calculations. For easier reference, figure 2.9 is repeated here as fig. A.1.

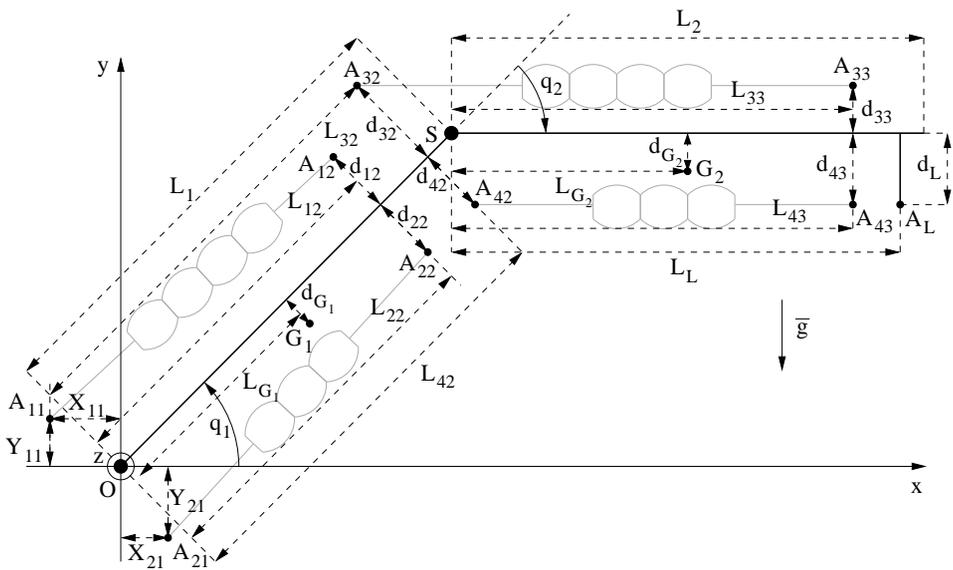


Figure A.1: Schematic representation of the manipulator with all distances, angles and attachment points indicated.

A.1 Muscle torques

In eq. (2.8) on page 20, the torque $\tau_{m,1}$ due to muscle 1 in the first joint is given as

$$\tau_{m,1} = F_1 (\mathbf{e}_1 \times \mathbf{OA}_{12}) \cdot \mathbf{1}_z. \quad (\text{A.1})$$

Similarly, we have

$$\tau_{m,2} = F_2 (\mathbf{e}_2 \times \mathbf{OA}_{22}) \cdot \mathbf{1}_z \quad (\text{A.2})$$

for muscle 2. The torque due to muscles 3 and 4 in the second joint is given by

$$\tau_{m,3} = F_3 (\mathbf{e}_3 \times \mathbf{SA}_{33}) \cdot \mathbf{1}_z \quad (\text{A.3})$$

$$\tau_{m,4} = F_4 (\mathbf{e}_4 \times \mathbf{SA}_{43}) \cdot \mathbf{1}_z. \quad (\text{A.4})$$

In these expressions, F_i is the force exerted by muscle i , and \mathbf{e}_i are unit vectors along the longitudinal axes of the muscles:

$$\mathbf{e}_1 = \frac{\mathbf{OA}_{12} - \mathbf{OA}_{11}}{\|\mathbf{OA}_{12} - \mathbf{OA}_{11}\|} \quad (\text{A.5})$$

$$\mathbf{e}_2 = \frac{\mathbf{OA}_{22} - \mathbf{OA}_{21}}{\|\mathbf{OA}_{22} - \mathbf{OA}_{21}\|} \quad (\text{A.6})$$

$$\mathbf{e}_3 = \frac{\mathbf{OA}_{33} - \mathbf{OA}_{32}}{\|\mathbf{OA}_{33} - \mathbf{OA}_{32}\|} \quad (\text{A.7})$$

$$\mathbf{e}_4 = \frac{\mathbf{OA}_{43} - \mathbf{OA}_{42}}{\|\mathbf{OA}_{43} - \mathbf{OA}_{42}\|}. \quad (\text{A.8})$$

A.2 Torque functions

Eq. (2.10) on page 20 states the definition of the torque function associated to muscle 1, $m_{\tau 1}$, which is derived from (A.1) and (2.2):

$$m_{\tau 1} \left(\epsilon_1(q_1), \frac{l_0}{R} \right) = l_0^2 f_{t0} \left(\epsilon_1(q_1), \frac{l_0}{R} \right) \cdot (\mathbf{e}_1 \times \mathbf{OA}_{12}) \cdot \mathbf{1}_z. \quad (\text{A.9})$$

In order to find an expression for $m_{\tau 1}$, we have to calculate the factor $(\mathbf{e}_1 \times \mathbf{OA}_{12}) \cdot \mathbf{1}_z$ as well as $\epsilon_1(q_1)$.

Similarly, the other torque functions are given by

$$m_{\tau 2} \left(\epsilon_2(q_1), \frac{l_0}{R} \right) = l_0^2 f_{t0} \left(\epsilon_2(q_1), \frac{l_0}{R} \right) \cdot (\mathbf{e}_2 \times \mathbf{OA}_{22}) \cdot \mathbf{1}_z \quad (\text{A.10})$$

$$m_{\tau 3} \left(\epsilon_3(q_2), \frac{l_0}{R} \right) = l_0^2 f_{t0} \left(\epsilon_3(q_2), \frac{l_0}{R} \right) \cdot (\mathbf{e}_3 \times \mathbf{SA}_{33}) \cdot \mathbf{1}_z \quad (\text{A.11})$$

$$m_{\tau 4} \left(\epsilon_4(q_2), \frac{l_0}{R} \right) = l_0^2 f_{t0} \left(\epsilon_4(q_2), \frac{l_0}{R} \right) \cdot (\mathbf{e}_4 \times \mathbf{SA}_{43}) \cdot \mathbf{1}_z. \quad (\text{A.12})$$

In these expressions, ϵ_i is the contraction of muscle i , which depends on the angle of the link actuated by the muscle. Using the following coordinates, which can be derived from figure A.1,

$$A_{11} = [-X_{11}, Y_{11}, 0]^T \quad (\text{A.13})$$

$$A_{12} = [-\sin(q_1) d_{12} + \cos(q_1) L_{12}, \cos(q_1) d_{12} + \sin(q_1) L_{12}, 0]^T \quad (\text{A.14})$$

$$A_{21} = [X_{21}, -Y_{21}, 0]^T \quad (\text{A.15})$$

$$A_{22} = [\sin(q_1) d_{22} + \cos(q_1) L_{22}, -\cos(q_1) d_{22} + \sin(q_1) L_{22}, 0]^T \quad (\text{A.16})$$

$$A_{32} = [-\sin(q_1) d_{32} + \cos(q_1) L_{32}, \cos(q_1) d_{32} + \sin(q_1) L_{32}, 0]^T \quad (\text{A.17})$$

$$A_{33} = [-\sin(q_1 + q_2) d_{33} + \cos(q_1) L_1 + \cos(q_1 + q_2) L_{33}, \\ \cos(q_1 + q_2) d_{33} + \sin(q_1) L_1 + \sin(q_1 + q_2) L_{33}, 0]^T \quad (\text{A.18})$$

$$A_{42} = [\sin(q_1) d_{42} + \cos(q_1) L_{42}, -\cos(q_1) d_{42} + \sin(q_1) L_{42}, 0]^T \quad (\text{A.19})$$

$$A_{43} = [\sin(q_1 + q_2) d_{43} + \cos(q_1) L_1 + \cos(q_1 + q_2) L_{43}, \\ -\cos(q_1 + q_2) d_{43} + \sin(q_1) L_1 + \sin(q_1 + q_2) L_{43}, 0]^T \quad (\text{A.20})$$

$$A_L = [L_L \cos(q_1 + q_2) + d_L \sin(q_1 + q_2) + \cos(q_1) L_1, \\ -d_L \cos(q_1 + q_2) + L_L \sin(q_1 + q_2) + \sin(q_1) L_1, 0]^T \quad (\text{A.21})$$

$$S = [\cos(q_1) L_1, \sin(q_1) L_1, 0]^T, \quad (\text{A.22})$$

and the definition of the \mathbf{e}_i given above, we can write eqs. (A.9)-(A.12) as follows:

$$m_{\tau 1} \left(\epsilon_1(q_1), \frac{l_0}{R} \right) = l_0^2 f_{t0} \left(\epsilon_1(q_1), \frac{l_0}{R} \right) \cdot (\sin(q_1) (L_{12} X_{11} - d_{12} Y_{11}) + \\ \cos(q_1) (d_{12} X_{11} + L_{12} Y_{11})) \\ \cdot \left((-\sin(q_1) d_{12}) + \cos(q_1) L_{12} + X_{11} \right)^2 + \\ (\cos(q_1) d_{12} + \sin(q_1) L_{12} - Y_{11})^2 \Big)^{-1/2} \quad (\text{A.23})$$

$$m_{\tau 2} \left(\epsilon_2(q_1), \frac{l_0}{R} \right) = l_0^2 f_{t0} \left(\epsilon_2(q_1), \frac{l_0}{R} \right) \cdot (-\sin(q_1) (L_{22} X_{21} + d_{22} Y_{21})) + \\ \cos(q_1) (d_{22} X_{21} - L_{22} Y_{21}))$$

$$\begin{aligned} & \cdot \left((\sin(q_1) d_{22} + \cos(q_1) L_{22} - X_{21})^2 + \right. \\ & \left. (-\cos(q_1) d_{22} + \sin(q_1) L_{22} + Y_{21})^2 \right)^{-1/2} \end{aligned} \quad (\text{A.24})$$

$$\begin{aligned} m_{\tau 3} \left(\epsilon_3(q_2), \frac{l_0}{R} \right) &= l_0^2 f_{t0} \left(\epsilon_3(q_2), \frac{l_0}{R} \right) \cdot (\cos(q_2) (d_{33} (L_1 - L_{32}) + d_{32} L_{33}) \\ & - \sin(q_2) (d_{32} d_{33} + (-L_1 + L_{32}) L_{33})) \\ & \cdot \left(d_{32}^2 + d_{33}^2 + (L_1 - L_{32})^2 + L_{33}^2 - 2 \sin(q_2) \right. \\ & \cdot (d_{33} L_1 - d_{33} L_{32} + d_{32} L_{33}) - \\ & \left. 2 \cos(q_2) (d_{32} d_{33} + (-L_1 + L_{32}) L_{33}) \right)^{-1/2} \end{aligned} \quad (\text{A.25})$$

$$\begin{aligned} m_{\tau 4} \left(\epsilon_4(q_2), \frac{l_0}{R} \right) &= l_0^2 f_{t0} \left(\epsilon_4(q_2), \frac{l_0}{R} \right) \cdot (\cos(q_2) (-(d_{43} (L_1 - L_{42})) - d_{42} L_{43}) \\ & - \sin(q_2) (d_{42} d_{43} + (-L_1 + L_{42}) L_{43})) \\ & \cdot \left(d_{42}^2 + d_{43}^2 + (L_1 - L_{42})^2 + L_{43}^2 + 2 \sin(q_2) \right. \\ & \cdot (d_{43} L_1 - d_{43} L_{42} + d_{42} L_{43}) - \\ & \left. 2 \cos(q_2) (d_{42} d_{43} + (-L_1 + L_{42}) L_{43}) \right)^{-1/2} \end{aligned} \quad (\text{A.26})$$

A.3 Contractions

In order to be able to evaluate the torque functions, we still have to find the functions $\epsilon_i(\gamma)$, with $\gamma = q_1$ for muscles 1 and 2 and $\gamma = q_2$ for muscles 3 and 4. $\epsilon_1(q_1)$ is defined in eq. (2.13) (see page 22) as

$$\epsilon_1(q_1) = \frac{L_{conn,1} + n_{m,1} l_0 - d_1(q_1)}{n_{m,1} l_0}. \quad (\text{A.27})$$

The others can be defined similarly,

$$\epsilon_2(q_1) = \frac{L_{conn,2} + n_{m,2} l_0 - d_2(q_1)}{n_{m,2} l_0} \quad (\text{A.28})$$

$$\epsilon_3(q_2) = \frac{L_{conn,3} + n_{m,3} l_0 - d_3(q_2)}{n_{m,3} l_0} \quad (\text{A.29})$$

$$\epsilon_4(q_2) = \frac{L_{conn,4} + n_{m,4} l_0 - d_4(q_2)}{n_{m,4} l_0}, \quad (\text{A.30})$$

with

$$\begin{aligned}
 d_1(q_1) &= \|\mathbf{OA}_{12} - \mathbf{OA}_{11}\| \\
 &= \left((-\sin(q_1) d_{12}) + \cos(q_1) L_{12} + X_{11} \right)^2 + \\
 &\quad \left(\cos(q_1) d_{12} + \sin(q_1) L_{12} - Y_{11} \right)^2 \Big)^{1/2} \tag{A.31}
 \end{aligned}$$

$$\begin{aligned}
 d_2(q_1) &= \|\mathbf{OA}_{22} - \mathbf{OA}_{21}\| \\
 &= \left((\sin(q_1) d_{22} + \cos(q_1) L_{22} - X_{21})^2 + \right. \\
 &\quad \left. (-\cos(q_1) d_{22}) + \sin(q_1) L_{22} + Y_{21} \right)^2 \Big)^{1/2} \tag{A.32}
 \end{aligned}$$

$$\begin{aligned}
 d_3(q_2) &= \|\mathbf{OA}_{33} - \mathbf{OA}_{32}\| \\
 &= \left(d_{32}^2 + d_{33}^2 + (L_1 - L_{32})^2 + L_{33}^2 - \right. \\
 &\quad 2 \sin(q_2) (d_{33} L_1 - d_{33} L_{32} + d_{32} L_{33}) - \\
 &\quad \left. 2 \cos(q_2) (d_{32} d_{33} + (-L_1 + L_{32}) L_{33}) \right)^{1/2} \tag{A.33}
 \end{aligned}$$

$$\begin{aligned}
 d_4(q_2) &= \|\mathbf{OA}_{43} - \mathbf{OA}_{42}\| \\
 &= \left(d_{42}^2 + d_{43}^2 + (L_1 - L_{42})^2 + L_{43}^2 + \right. \\
 &\quad 2 \sin(q_2) (d_{43} L_1 - d_{43} L_{42} + d_{42} L_{43}) - \\
 &\quad \left. 2 \cos(q_2) (d_{42} d_{43} + (-L_1 + L_{42}) L_{43}) \right)^{1/2} \tag{A.34}
 \end{aligned}$$

the total lengths of the different muscle groups. Since the number of muscles in each series arrangement $n_{m,i}$ has been chosen during the design ($n_{m,1} = 4$, $n_{m,2} = 3$, $n_{m,3} = 4$ and $n_{m,4} = 3$), all we still have to do to find ϵ_i is calculate the connection lengths $L_{conn,i}$. From section 2.3.2.4 we have

$$L_{conn,1} = d_{1,max} - n_{m,1} l_0 (1 - \epsilon_{1,min}) \tag{A.35}$$

$$L_{conn,2} = d_{2,min} - n_{m,2} l_0 (1 - \epsilon_{2,max}), \tag{A.36}$$

the other ones are defined as

$$L_{conn,3} = d_{3,max} - n_{m,3} l_0 (1 - \epsilon_{3,min}) \tag{A.37}$$

$$L_{conn,4} = d_{4,min} - n_{m,4} l_0 (1 - \epsilon_{4,max}). \tag{A.38}$$

Parameter	Value (mm)
$d_{1,max}$	345.0
$d_{2,min}$	200.5
$d_{3,max}$	322.3
$d_{4,min}$	265.8
$L_{conn,1}$	117.0
$L_{conn,2}$	74.5
$L_{conn,3}$	94.3
$L_{conn,4}$	139.8

Table A.1: Parameter necessary to calculate the torque functions of the manipulator.

In these expressions, the maximum and minimum muscle lengths in the workspace are given by

$$d_{1,max} = \arg \max_{q_{1,min} \leq q_1 \leq q_{1,max}} d_1(q_1) \quad (\text{A.39})$$

$$d_{2,min} = \arg \min_{q_{1,min} \leq q_1 \leq q_{1,max}} d_2(q_1) \quad (\text{A.40})$$

$$d_{3,max} = \arg \max_{q_{2,min} \leq q_2 \leq q_{2,max}} d_3(q_2) \quad (\text{A.41})$$

$$d_{4,min} = \arg \min_{q_{2,min} \leq q_2 \leq q_{2,max}} d_4(q_2) \quad (\text{A.42})$$

The minimum values of contraction that are acceptable, $\epsilon_{1,min}$ and $\epsilon_{3,min}$, are both chosen equal to 5%, the maximum values $\epsilon_{3,max}$ and $\epsilon_{4,max}$ are set at 30%, as explained in section 2.3.2.4.

For a given set of attachment point locations and a given workspace, we can determine $d_{1,max}$, $d_{2,min}$, $d_{3,max}$ and $d_{4,min}$ from eqs. (A.39)-(A.42) and substitute them in eqs. (A.35)-(A.38). Since we also know the muscle length functions from eqs. (A.31)-(A.34), all the unknowns in the muscle contraction equations (A.27)-(A.30) have now been calculated. Eqs. (A.27)-(A.30) can now be used to evaluate the torque functions (A.23)-(A.26) once muscle parameters l_0 and l_0/R have been chosen.

For the workspace given by (2.6)-(2.7) (page 18), the attachment point locations given in table 2.1 (page 26) and the muscle parameters chosen in section 2.3.3.1 (page 25), the minimum and maximum muscle lengths and the connection lengths are listed in table A.1.

A.4 Maximum load

In section 2.3.3.5 (page 27), the maximum load is calculated. This section provides the equations from section 2.3.3.5 in more detail.

Eq. (2.22) shows the maximum load according to the second joint when muscle 3 is the carrying muscle. For completeness, it is repeated here as $M_{max,2a}$, together with the maximum load according to the first link and the maximum load according to the second link when muscle 4 is the carrying muscle:

$$M_{max,1} = \frac{p_{max} \cdot m_{\tau 1}(q_1) + (\mathbf{OG}_1 \times (-m_1 g \mathbf{1}_y) + \mathbf{OG}_2 \times (-m_2 g \mathbf{1}_y)) \cdot \mathbf{1}_z}{(\mathbf{OA}_L \times g \mathbf{1}_y) \cdot \mathbf{1}_z} \quad (\text{A.43})$$

$$M_{max,2a} = \frac{p_{max} \cdot m_{\tau 3}(q_2) + (\mathbf{SG}_2 \times (-m_2 g \mathbf{1}_y)) \cdot \mathbf{1}_z}{(\mathbf{SA}_L \times g \mathbf{1}_y) \cdot \mathbf{1}_z} \quad (\text{A.44})$$

$$M_{max,2b} = \frac{p_{max} \cdot m_{\tau 4}(q_2) + (\mathbf{SG}_2 \times (-m_2 g \mathbf{1}_y)) \cdot \mathbf{1}_z}{(\mathbf{SA}_L \times g \mathbf{1}_y) \cdot \mathbf{1}_z} \quad (\text{A.45})$$

If $M_{max,2a} > 0$ at a certain point in the workspace, then the maximum load for the manipulator in that point is given by

$$M_{max} = \min(M_{max,1}, M_{max,2a}),$$

otherwise it is given by

$$M_{max} = \min(M_{max,1}, M_{max,2b}).$$

As a function of the coordinates (A.13)-(A.22), equations (A.43)-(A.45) become

$$\begin{aligned} M_{max,1} &= (- (g \sin(q_1) d_{G_1} m_1) - g \cos(q_1) L_{G_1} m_1 - g \sin(q_1 + q_2) d_{G_2} m_2 - \\ &\quad g \cos(q_1) L_1 m_2 - g \cos(q_1 + q_2) L_{G_2} m_2 + p_{max} m_{\tau 1}(q_1)) \\ &\quad \cdot (g (L_L \cos(q_1 + q_2) + d_L \sin(q_1 + q_2) + \cos(q_1) L_1))^{-1} \\ M_{max,2a} &= \frac{- (g \sin(q_1 + q_2) d_{G_2} m_2) - g \cos(q_1 + q_2) L_{G_2} m_2 + p_{max} m_{\tau 3}(q_2)}{g (L_L \cos(q_1 + q_2) + d_L \sin(q_1 + q_2))} \\ M_{max,2b} &= \frac{- (g \sin(q_1 + q_2) d_{G_2} m_2) - g \cos(q_1 + q_2) L_{G_2} m_2 + p_{max} m_{\tau 4}(q_2)}{g (L_L \cos(q_1 + q_2) + d_L \sin(q_1 + q_2))}. \end{aligned}$$

Figure 2.15 on page 30 shows the maximum load throughout the workspace, with the results being presented in task space. Sometimes it is easier to look at the same data in joint space, as is shown in fig. A.2.

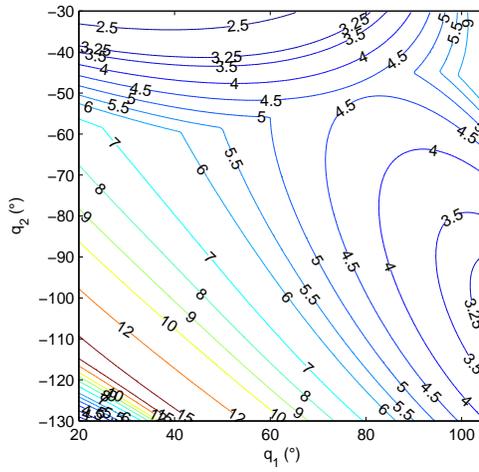


Figure A.2: Contour plot showing the maximum load (in kg) that the manipulator can carry throughout its workspace (based on the same data as fig. 2.15).

Appendix B

Dynamic Model

This appendix gives the expressions for the matrices used in the equation of motion (3.1). Please refer to fig. 2.9 for the definition of the different angles and distances. m_1 and m_2 are the masses of both links, respectively, and I_{zG_1} and I_{zG_2} are their moments of inertia (about axes perpendicular to the $x - y$ plane (see fig. 2.9) and through their respective centers of gravity G_1 and G_2). g denotes the gravitational acceleration.

B.1 Equations of motion

Since there are two degrees of freedom, there are two equations of motion, which can be written as follows using the Lagrange formulation:

$$\frac{d}{dt} \left(\frac{\partial \mathcal{L}}{\partial \dot{q}_i} \right) - \frac{\partial \mathcal{L}}{\partial q_i} = \tau_i \quad i = 1 \dots 2 \quad (\text{B.1})$$

In these equations, $\mathcal{L} = \mathcal{K} - \mathcal{P}$, with \mathcal{K} the total kinetic energy and \mathcal{P} the total potential energy. In the case of our manipulator, the τ_i are given by eq. (2.16). The kinetic energy can be calculated by (with $n = 2$)

$$\mathcal{K} = \frac{1}{2} \sum_{i=1}^n (m_i v_{G_i}^2 + I_{zG_i} \dot{q}_i^2),$$

with v_{G_i} the velocity of the center of mass of link i . This results in

$$\begin{aligned} \mathcal{K} = & \frac{1}{2} \left(\dot{q}_1^2 (I_{zG_1} + (d_{G_1}^2 + L_{G_1}^2) m_1) + (\dot{q}_1 + \dot{q}_2)^2 I_{zG_2} + \right. \\ & \left((\dot{q}_1 + \dot{q}_2)^2 d_{G_2}^2 + \dot{q}_1^2 L_1^2 + (\dot{q}_1 + \dot{q}_2)^2 L_{G_2}^2 + 2\dot{q}_1 (\dot{q}_1 + \dot{q}_2) L_1 (\sin(q_2) d_{G_2} + \right. \\ & \left. \left. \cos(q_2) L_{G_2}) \right) m_2 \right). \end{aligned}$$

The gravitational potential energy \mathcal{P} is given by

$$\begin{aligned}\mathcal{P} &= m_1 g (L_{G_1} \sin q_1 - d_{G_1} \cos q_1) \\ &\quad + m_2 g (L_1 \sin q_1 + L_{G_2} \sin (q_1 + q_2) - d_{G_2} \cos (q_1 + q_2)).\end{aligned}$$

With these expressions, the equations of motion can be determined using (B.1).

It is also possible to obtain the equations of motion immediately in the form of eq. (3.1), i.e.

$$H(\mathbf{q}) \ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) = \boldsymbol{\tau}.$$

The inertia matrix $H(\mathbf{q})$ can be obtained from the following relation with the kinetic energy (Spong et al., 2006):

$$\mathcal{K} = \frac{1}{2} \dot{\mathbf{q}}^T H(\mathbf{q}) \dot{\mathbf{q}}.$$

Vector $\mathbf{G}(\mathbf{q})$ can be found from the gravitational potential energy,

$$g_i(\mathbf{q}) = \frac{\partial \mathcal{P}}{\partial q_i}$$

with $g_i(\mathbf{q})$ the i -th component of $\mathbf{G}(\mathbf{q})$.

Matrix $C(\mathbf{q}, \dot{\mathbf{q}})$ is not unique, a common choice for $C(\mathbf{q}, \dot{\mathbf{q}})$ is given by

$$c_{k,j} = \sum_{i=1}^n \frac{1}{2} \left(\frac{\partial h_{k,j}}{\partial q_i} + \frac{\partial h_{k,i}}{\partial q_j} - \frac{\partial h_{i,j}}{\partial q_k} \right) \dot{q}_i,$$

where $c_{k,j}$ indicates element (k, j) of matrix $C(\mathbf{q}, \dot{\mathbf{q}})$, and similarly $h_{k,j}$ indicates element (k, j) of $H(\mathbf{q})$. Choosing $C(\mathbf{q}, \dot{\mathbf{q}})$ as above assures that the matrix $\dot{H}(\mathbf{q}) - 2C(\mathbf{q}, \dot{\mathbf{q}})$ is skew-symmetric (Spong et al., 2006). It also implies (De Luca et al., 2006)

$$\dot{H}(\mathbf{q}) = C(\mathbf{q}, \dot{\mathbf{q}}) + C^T(\mathbf{q}, \dot{\mathbf{q}}), \quad (\text{B.2})$$

which can be seen as follows (using the symmetry of $H(\mathbf{q})$, i.e. $h_{k,j} = h_{j,k}$):

$$\begin{aligned}c_{k,j} + c_{j,k} &= \sum_{i=1}^n \frac{1}{2} \left(\frac{\partial h_{k,j}}{\partial q_i} + \frac{\partial h_{k,i}}{\partial q_j} - \frac{\partial h_{i,j}}{\partial q_k} + \frac{\partial h_{j,k}}{\partial q_i} + \frac{\partial h_{j,i}}{\partial q_k} - \frac{\partial h_{i,k}}{\partial q_j} \right) \dot{q}_i \\ &= \sum_{i=1}^n \frac{1}{2} \left(\frac{\partial h_{k,j}}{\partial q_i} + \frac{\partial h_{k,i}}{\partial q_j} - \frac{\partial h_{i,j}}{\partial q_k} + \frac{\partial h_{k,j}}{\partial q_i} + \frac{\partial h_{i,j}}{\partial q_k} - \frac{\partial h_{k,i}}{\partial q_j} \right) \dot{q}_i \\ &= \sum_{i=1}^n \frac{\partial h_{k,j}}{\partial q_i} \dot{q}_i \\ &= \dot{h}_{k,j}.\end{aligned}$$

This property will be used in section 5.4.2.2.

For the 2-DOF pneumatic manipulator, the inertia matrix $H(\mathbf{q})$ is given by

$$H(\mathbf{q}) = \begin{bmatrix} h_{1,1} & h_{1,2} \\ h_{2,1} & h_{2,2} \end{bmatrix} \quad (\text{B.3})$$

with

$$h_{1,1} = I_{zG_1} + I_{zG_2} + (d_{G_1}^2 + L_{G_1}^2) m_1 + 2 L_1 (\sin(q_2) d_{G_2} + \cos(q_2) L_{G_2}) m_2 \\ + (d_{G_2}^2 + L_1^2 + L_{G_2}^2) m_2$$

$$h_{1,2} = h_{2,1} \\ = I_{zG_2} + \sin(q_2) d_{G_2} L_1 m_2 + \cos(q_2) L_1 L_{G_2} m_2 + (d_{G_2}^2 + L_{G_2}^2) m_2$$

$$h_{2,2} = I_{zG_2} + (d_{G_2}^2 + L_{G_2}^2) m_2.$$

Similarly, for $C(\mathbf{q}, \dot{\mathbf{q}})$ we have

$$C(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} c_{1,1} & c_{1,2} \\ c_{2,1} & c_{2,2} \end{bmatrix}$$

where (using $\omega_i = \dot{q}_i$)

$$c_{1,1} = L_1 (\cos(q_2) d_{G_2} - \sin(q_2) L_{G_2}) m_2 \omega_2$$

$$c_{1,2} = L_1 (\cos(q_2) d_{G_2} - \sin(q_2) L_{G_2}) m_2 \omega_1 + L_1 (\cos(q_2) d_{G_2} - \sin(q_2) L_{G_2}) m_2 \omega_2$$

$$c_{2,1} = L_1 (-\cos(q_2) d_{G_2} + \sin(q_2) L_{G_2}) m_2 \omega_1$$

$$c_{2,2} = 0$$

and $\mathbf{G}(\mathbf{q})$ is given by

$$\mathbf{G}(\mathbf{q}) = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix}$$

with

$$g_1 = g (\sin(q_1) d_{G_1} m_1 + \sin(q_1 + q_2) d_{G_2} m_2 + \cos(q_1 + q_2) L_{G_2} m_2 \\ + \cos(q_1) (L_{G_1} m_1 + L_1 m_2))$$

$$g_2 = g (\sin(q_1 + q_2) d_{G_2} + \cos(q_1 + q_2) L_{G_2}) m_2.$$

B.2 Adapted torque functions

In order to use eq. (3.9) as a model for muscle force, the definition of the torque functions as given by (A.9)-(A.12) has to be slightly adapted. Since the parameters

in (3.9) are estimated for a single muscle, eq. (3.9) no longer (explicitly) depends on the slenderness l_0/R , and neither does the torque function associated with the muscle.

Using (3.10) and (3.11), and keeping in mind (2.9), the new definition of the torque functions can be derived from (A.1)-(A.4) as

$$\begin{aligned} m_{\tau_1}(\epsilon_1(q_1)) &= [\epsilon^{-1} \quad 1 \quad \epsilon \quad \epsilon^2 \quad \epsilon^3] \hat{\mathbf{a}}_1 \cdot ((\mathbf{e}_1 \times \mathbf{OA}_{12}) \cdot \mathbf{1}_z) \\ m_{\tau_2}(\epsilon_2(q_1)) &= [\epsilon^{-1} \quad 1 \quad \epsilon \quad \epsilon^2 \quad \epsilon^3] \hat{\mathbf{a}}_2 \cdot ((\mathbf{e}_2 \times \mathbf{OA}_{22}) \cdot \mathbf{1}_z) \\ m_{\tau_3}(\epsilon_3(q_2)) &= [\epsilon^{-1} \quad 1 \quad \epsilon \quad \epsilon^2 \quad \epsilon^3] \hat{\mathbf{a}}_3 \cdot ((\mathbf{e}_3 \times \mathbf{SA}_{33}) \cdot \mathbf{1}_z) \\ m_{\tau_4}(\epsilon_4(q_2)) &= [\epsilon^{-1} \quad 1 \quad \epsilon \quad \epsilon^2 \quad \epsilon^3] \hat{\mathbf{a}}_4 \cdot ((\mathbf{e}_4 \times \mathbf{SA}_{43}) \cdot \mathbf{1}_z), \end{aligned}$$

with $\hat{\mathbf{a}}_i$ the estimated muscle parameters of muscle i .

B.3 Equations of motion linear in the parameters

In this section the equations of motion are written in a form that is linear in the parameters, which is useful for parameter estimation (see section 3.5).

In general, each link has 10 inertial parameters (its mass, 6 elements of the inertia tensor and 3 coordinates of the center of mass). Not all of them influence the dynamics, however (for example, since the system considered in this work is constrained to the $x - y$ plane, there are no rotations about the x and y axes, so the moments of inertia around these axes don't matter). The ones that influence joint torque (which is what will be measured to estimate the dynamic parameters) can be identified, but usually not individually.

As is well known, eq. (3.1) can be expressed in a form that is linear with respect to a set of parameters $\boldsymbol{\theta}$ (Khosla and Kanade, 1985; An et al., 1985; Nicolò and Katende, 1983). In order to emphasize its dependence on $\boldsymbol{\theta}$, we write (3.1) as

$$H(\mathbf{q}, \boldsymbol{\theta}) \ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}}, \boldsymbol{\theta}) \dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}, \boldsymbol{\theta}) = \boldsymbol{\tau}.$$

The goal of this section is to determine a matrix K that allows us to write the above equation as

$$K(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) \boldsymbol{\theta} = \boldsymbol{\tau}. \quad (\text{B.4})$$

We can write the moments of inertia of both links around axes perpendicular to the $x - y$ plane and through their hinge points as¹

$$I_{zO,1} = I_{zG_1} + m_1 (d_{G_1}^2 + L_{G_1}^2)$$

¹This is an application of Steiner's theorem, also known as the parallel axis theorem (see for instance Goodman and Warner (2001)).

$$I_{zS,2} = I_{zG_2} + m_2 (d_{G_2}^2 + L_{G_2}^2).$$

Then, following An et al. (1988) and keeping in mind the restriction to the $x - y$ plane, we could choose

$$\boldsymbol{\theta} = [m_1 \quad m_1 L_{G_1} \quad m_1 d_{G_1} \quad I_{zO,1} \quad m_2 \quad m_2 L_{G_2} \quad m_2 d_{G_2} \quad I_{zS,2}]^T,$$

to be changed in

$$\boldsymbol{\theta} = [m_1 L_{G_1} \quad m_1 d_{G_1} \quad I_{zO,1} \quad m_2 \quad m_2 L_{G_2} \quad m_2 d_{G_2} \quad I_{zS,2}]^T$$

after observing that m_1 doesn't appear (by itself) in the equations of motion. This is the most straightforward and intuitive choice of parameters that appear linearly in the equations of motion. It is not the optimal choice though, since it results in a matrix $K(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$ that is not full rank.

In order to avoid this, $\boldsymbol{\theta}$ was chosen as

$$\boldsymbol{\theta} = \begin{bmatrix} m_1 L_{G_1} + m_2 L_1 \\ m_1 d_{G_1} \\ I_{zO,1} + m_2 L_1^2 \\ m_2 L_{G_2} \\ m_2 d_{G_2} \\ I_{zS,2} \end{bmatrix}, \tag{B.5}$$

which results in the following observation matrix:

$$K(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}}) = \begin{bmatrix} g \cos(q_1) & g \sin(q_1) & \ddot{q}_1 & g \cos(q_1 + q_2) + (2\ddot{q}_1 + \ddot{q}_2) \cos(q_2)L_1 - \dot{q}_2(2\dot{q}_1 + \dot{q}_2) \sin(q_2)L_1 & \dots \\ 0 & 0 & 0 & g \cos(q_1 + q_2) + \dot{q}_1 \cos(q_2)L_1 + \dot{q}_1^2 \sin(q_2)L_1 & \dots \\ \dots & g \sin(q_1 + q_2) + \dot{q}_2(2\dot{q}_1 + \dot{q}_2) \cos(q_2)L_1 + (2\ddot{q}_1 + \ddot{q}_2) \sin(q_2)L_1 & \ddot{q}_1 + \ddot{q}_2 & \dots \\ \dots & g \sin(q_1 + q_2) - \dot{q}_1^2 \cos(q_2)L_1 + \dot{q}_1 \sin(q_2)L_1 & \ddot{q}_1 + \ddot{q}_2 & \dots \end{bmatrix}.$$

Although in this case the parameter set $\boldsymbol{\theta}$ was found by inspection of the equations of motion, it should be noted that methods exist that allow such parameters sets to be determined directly (Gautier and Khalil, 1990; Fiset et al., 1996). This is especially useful for systems with more degrees of freedom than the one considered here.

B.4 Filtered equations of motion linear in the parameters

The filtering operation in the method by Slotine and Li (1991) (see 3.5.2.2 on page 51) leads to the following filtered left-hand side of the equations of motion (repeated

here from eq. (3.24):

$$\omega H(\mathbf{q}) \dot{\mathbf{q}} + \int_0^t f(t-r) \left\{ C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) - \dot{H}(\mathbf{q}) \dot{\mathbf{q}} \right\} dr + \int_0^t \dot{f}(t-r) \{ H(\mathbf{q}) \dot{\mathbf{q}} \} dr.$$

For this expression to be useful for identification, we have to write $H(\mathbf{q}) \dot{\mathbf{q}}$, $\dot{H}(\mathbf{q}) \dot{\mathbf{q}}$ and $C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{G}(\mathbf{q})$ as linear functions of $\boldsymbol{\theta}$ (which is defined in eq. (B.5)). By inspection of the expressions for $H(\mathbf{q})$, $C(\mathbf{q}, \dot{\mathbf{q}})$ and $\mathbf{G}(\mathbf{q})$ given above, we have

$$\begin{aligned} H(\mathbf{q}) \dot{\mathbf{q}} &= \begin{bmatrix} 0 & 0 & \dot{q}_1 & 2\dot{q}_1 \cos(q_2)L_1 + \dot{q}_2 \cos(q_2)L_1 & 2\dot{q}_1 \sin(q_2)L_1 + \dot{q}_2 \sin(q_2)L_1 & \dot{q}_1 + \dot{q}_2 \\ 0 & 0 & 0 & \dot{q}_1 \cos(q_2)L_1 & \dot{q}_1 \sin(q_2)L_1 & \dot{q}_1 + \dot{q}_2 \end{bmatrix} \boldsymbol{\theta} \\ &= W_1(\mathbf{q}, \dot{\mathbf{q}}) \boldsymbol{\theta}, \\ \dot{H}(\mathbf{q}) \dot{\mathbf{q}} &= \begin{bmatrix} 0 & 0 & 0 & -2\dot{q}_1 \dot{q}_2 \sin(q_2)L_1 - \dot{q}_2^2 \sin(q_2)L_1 & 2\dot{q}_1 \dot{q}_2 \cos(q_2)L_1 + \dot{q}_2^2 \cos(q_2)L_1 & 0 \\ 0 & 0 & 0 & -\dot{q}_1 \dot{q}_2 \sin(q_2)L_1 & \dot{q}_1 \dot{q}_2 \cos(q_2)L_1 & 0 \end{bmatrix} \boldsymbol{\theta} \\ &= W_2(\mathbf{q}, \dot{\mathbf{q}}) \boldsymbol{\theta} \end{aligned}$$

and

$$\begin{aligned} C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) &= \begin{bmatrix} g \cos(q_1) & g \sin(q_1) & 0 & g \cos(q_1 + q_2) - 2\dot{q}_1 \dot{q}_2 \sin(q_2)L_1 - \dot{q}_2^2 \sin(q_2)L_1 & \dots \\ 0 & 0 & 0 & g \cos(q_1 + q_2) + \dot{q}_1^2 \sin(q_2)L_1 & \dots \\ \dots & g \sin(q_1 + q_2) + 2\dot{q}_1 \dot{q}_2 \cos(q_2)L_1 + \dot{q}_2^2 \cos(q_2)L_1 & 0 \\ & g \sin(q_1 + q_2) - \dot{q}_1^2 \cos(q_2)L_1 & 0 \end{bmatrix} \boldsymbol{\theta} \\ &= W_3(\mathbf{q}, \dot{\mathbf{q}}) \boldsymbol{\theta}, \end{aligned}$$

with $W_1(\mathbf{q}, \dot{\mathbf{q}})$, $W_2(\mathbf{q}, \dot{\mathbf{q}})$ and $W_3(\mathbf{q}, \dot{\mathbf{q}})$ defined by the above equations.

B.4.1 Friction

When incorporating the friction term (3.30) into the equations of motion, parameter vector $\boldsymbol{\theta}$ has to be expanded to include the friction parameters:

$$\boldsymbol{\theta} = \begin{bmatrix} m_1 L_{G_1} + m_2 L_1 \\ m_1 d_{G_1} \\ I_{zO,1} + m_2 L_1^2 \\ m_2 L_{G_2} \\ m_2 d_{G_2} \\ I_{zS,2} \\ c_1 \\ c_2 \\ b_1 \\ b_2 \end{bmatrix}. \quad (\text{B.6})$$

The matrices $W_i(\mathbf{q}, \dot{\mathbf{q}})$ of the filtered model described above have to be slightly changed: both $W_1(\mathbf{q}, \dot{\mathbf{q}})$ and $W_2(\mathbf{q}, \dot{\mathbf{q}})$ simply get a 2 by 4 matrix of zeros appended on the right-hand side, while $W_3(\mathbf{q}, \dot{\mathbf{q}})$, which is now defined by

$$C(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} + \mathbf{G}(\mathbf{q}) + \boldsymbol{\tau}_f(\dot{\mathbf{q}}) = W_3(\mathbf{q}, \dot{\mathbf{q}}) \boldsymbol{\theta},$$

becomes

$$W_3(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} g \cos(q_1) & g \sin(q_1) & 0 & g \cos(q_1 + q_2) - 2\dot{q}_1 \dot{q}_2 \sin(q_2)L_1 - \dot{q}_2^2 \sin(q_2)L_1 & \dots \\ 0 & 0 & 0 & g \cos(q_1 + q_2) + \dot{q}_1^2 \sin(q_2)L_1 & \\ \dots & g \sin(q_1 + q_2) + 2\dot{q}_1 \dot{q}_2 \cos(q_2)L_1 + \dot{q}_2^2 \cos(q_2)L_1 & 0 & \operatorname{sgn}(\dot{q}_1) & 0 & \dot{q}_1 & 0 \\ & g \sin(q_1 + q_2) - \dot{q}_1^2 \cos(q_2)L_1 & 0 & 0 & \operatorname{sgn}(\dot{q}_2) & 0 & \dot{q}_2 \end{bmatrix}.$$

Since it is used in section 5.4.2.2, we also show $W_4(\mathbf{q}, \dot{\mathbf{q}})$ here, which is defined by (cf. eq. (5.55))

$$W_4(\mathbf{q}, \dot{\mathbf{q}}) \boldsymbol{\theta} = C^T(\mathbf{q}, \dot{\mathbf{q}}) \dot{\mathbf{q}} - \mathbf{G}(\mathbf{q}) - \boldsymbol{\tau}_f(\dot{\mathbf{q}}).$$

The matrix is given by

$$W_4(\mathbf{q}, \dot{\mathbf{q}}) = \begin{bmatrix} -g \cos(q_1) & -g \sin(q_1) & 0 & -g \cos(q_1 + q_2) & & & & & & \\ 0 & 0 & 0 & -g \cos(q_1 + q_2) - \dot{q}_1^2 \sin(q_2)L_1 - \dot{q}_1 \dot{q}_2 \sin(q_2)L_1 & \dots & & & & & \\ \dots & -g \sin(q_1 + q_2) & 0 & -\operatorname{sgn}(\dot{q}_1) & 0 & -\dot{q}_1 & 0 & & & \\ & \dot{q}_1^2 \cos(q_2)L_1 + \dot{q}_1 \dot{q}_2 \cos(q_2)L_1 - g \sin(q_1 + q_2) & 0 & 0 & -\operatorname{sgn}(\dot{q}_2) & 0 & -\dot{q}_2 & 0 & & \end{bmatrix}.$$

B.5 Filtered observation matrix

If N measurements are taken, some precautions have to be taken before they are combined together in a single matrix, due to the filtering.

If we write $w_1^{(k)}$ to indicate the k 'th row of matrix W_1 , and $w_{1,j}^{(k)}$ to indicate this row as measured in the j 'th measurement (and similarly for the other matrices), we form the matrices

$$\tilde{W}_{i,(k)} = \begin{bmatrix} w_{i,1}^{(k)} \\ w_{i,2}^{(k)} \\ \vdots \\ w_{i,N}^{(k)} \end{bmatrix},$$

with i going from 1 to 3 (i.e. we combine the rows of matrices W_1 , W_2 and W_3) and k going from 1 to 2 (since all the W_i have two rows). In case of W_1 , for example, we combine the first row of all N measured matrices $W_{1,j}$ together to form $\tilde{W}_{1,(1)}$, and similarly combine the second row of all these matrices to form $\tilde{W}_{1,(2)}$. Similarly, we combine the first rows of all N measured torque vectors $\boldsymbol{\tau}_j$ to form matrix $\tilde{\boldsymbol{\tau}}_{(1)}$, and the second rows to form $\tilde{\boldsymbol{\tau}}_{(2)}$.

Two parts of the filtered observation matrix are then calculated using (3.26):

$$\tilde{K}'_{f,(k)} = \left(\omega \tilde{W}_{1,(k)}(\mathbf{q}, \dot{\mathbf{q}}) + \left\langle \tilde{W}_{3,(k)}(\mathbf{q}, \dot{\mathbf{q}}) - \tilde{W}_{2,(k)}(\mathbf{q}, \dot{\mathbf{q}}) \right\rangle_{F(s)} + \left\langle \tilde{W}_{1,(k)}(\mathbf{q}, \dot{\mathbf{q}}) \right\rangle_{F_2(s)} \right),$$

with $k = 1 \dots 2$. The filtering operations are interpreted to apply to the individual columns of the matrices. Similarly, we have

$$\tilde{\boldsymbol{\tau}}'_{f,(k)} = \left\langle \tilde{\boldsymbol{\tau}}_{(k)} \right\rangle_{F(s)}.$$

Parameter	Muscle number				Units
	1	2	3	4	
a_0	8.16906	62.7708	28.2623	8.15986	N/bar
a_1	295.069	-671.791	-16.2374	174.709	N/bar
a_2	-1586.59	5792.52	678.514	-490.279	N/bar
a_3	3623.1	-20984.8	-3595.7	-1182.38	N/bar
a_4	-5728.79	24787.8	3216.48	1905.53	N/bar

Table B.1: Estimated parameters of muscle model (3.10). 1 bar equals 100 *kPa*.

Parameter	Valve number				Units
	1	2	3	4	
v_1	0.0447	0.0097	0.0314	0.0078	bar
v_2	0.9933	0.9939	0.9975	0.9736	(dimensionless)

Table B.2: Estimated parameters of static valve model (3.12).

Since the filtering operations cause transients (which we don't want to include for the identification), we discard the first R rows of $\tilde{K}'_{f,(k)}$ and $\tilde{\tau}'_{f,(k)}$. The resulting matrices and vectors are given the same name, but without the accent (e.g. after dropping the first R rows of $\tilde{K}'_{f,(1)}$, we call it $\tilde{K}_{f,(1)}$). Finally, we can set

$$\tilde{K}_f = \begin{bmatrix} \tilde{K}_{f,(1)} \\ \tilde{K}_{f,(2)} \end{bmatrix}, \quad (\text{B.7})$$

$$\tilde{\tau}_f = \begin{bmatrix} \tilde{\tau}_{f,(1)} \\ \tilde{\tau}_{f,(2)} \end{bmatrix}$$

and write the system to be solved as

$$\tilde{K}_f \boldsymbol{\theta} = \tilde{\tau}_f.$$

\tilde{K}_f and $\tilde{\tau}_f$ thus have $2(N - R)$ rows. The value of R was chosen to be 300.

B.6 Results of the static estimation

Table 3.1 lists the parameter values of the staical mechanical model that were estimated using the static estimation procedures. The other parameters that were estimated with data from the experiment described in section 3.5.1.4 (page 47) are shown in tables B.1 and B.2.

Appendix C

Feedback Linearization

C.1 Matrices

Expressions for $H(\mathbf{q})$, $C(\mathbf{q}, \dot{\mathbf{q}})$ and $\mathbf{G}(\mathbf{q})$ as they appear in eq. (3.1) are given in appendix B. Eq. (4.21) in chapter 4, however, contains the matrices $A = H^{-1}C$ and H^{-1} , as well as vector $\mathbf{B} = H^{-1}\mathbf{G}$. Expressions for A , \mathbf{B} and H^{-1} are given in this section.

For the inverse of the inertia matrix H (given in eq. (B.3)) we have

$$H^{-1} = \begin{bmatrix} h_{1,1}^{-1} & h_{1,2}^{-1} \\ h_{2,1}^{-1} & h_{2,2}^{-1} \end{bmatrix}$$

with

$$h_{1,1}^{-1} = \frac{1}{|H|} (I_{zG_2} + (d_{G_2}^2 + L_{G_2}^2) m_2)$$

$$h_{1,2}^{-1} = h_{2,1}^{-1}$$

$$= \frac{1}{|H|} (-I_{zG_2} - \sin(q_2) d_{G_2} L_1 m_2 - \cos(q_2) L_1 L_{G_2} m_2 - (d_{G_2}^2 + L_{G_2}^2) m_2)$$

$$h_{2,2}^{-1} = \frac{1}{|H|} (I_{zG_1} + I_{zG_2} + (d_{G_1}^2 + L_{G_1}^2) m_1 + 2 L_1 (\sin(q_2) d_{G_2} + \cos(q_2) L_{G_2}) m_2 + (d_{G_2}^2 + L_1^2 + L_{G_2}^2) m_2).$$

The determinant $|H|$ used in these expressions is given by

$$|H| = - (L_1^2 (\sin(q_2) d_{G_2} + \cos(q_2) L_{G_2})^2 m_2^2) + (I_{zG_1} + (d_{G_1}^2 + L_{G_1}^2) m_1 + L_1^2 m_2) (I_{zG_2} + (d_{G_2}^2 + L_{G_2}^2) m_2).$$

Matrix $A = H^{-1}C$ is given by (using $\omega_i = \dot{q}_i$)

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix}$$

where

$$a_{1,1} = \frac{1}{|H|} (L_1 (\cos(q_2) d_{G_2} - \sin(q_2) L_{G_2}) m_2 (\sin(q_2) d_{G_2} L_1 m_2 \omega_1 + \cos(q_2) L_1 L_{G_2} m_2 \omega_1 + (I_{zG_2} + (d_{G_2}^2 + L_{G_2}^2) m_2) (\omega_1 + \omega_2)))$$

$$a_{1,2} = \frac{1}{|H|} (L_1 (\cos(q_2) d_{G_2} - \sin(q_2) L_{G_2}) m_2 (I_{zG_2} + (d_{G_2}^2 + L_{G_2}^2) m_2) (\omega_1 + \omega_2))$$

$$a_{2,1} = \frac{1}{|H|} (- (L_1 (\cos(q_2) d_{G_2} - \sin(q_2) L_{G_2}) m_2 ((I_{zG_1} + I_{zG_2} + (d_{G_1}^2 + L_{G_1}^2) m_1 + 2 L_1 (\sin(q_2) d_{G_2} + \cos(q_2) L_{G_2}) m_2 + (d_{G_2}^2 + L_1^2 + L_{G_2}^2) m_2) \omega_1 + (I_{zG_2} + \sin(q_2) d_{G_2} L_1 m_2 + \cos(q_2) L_1 L_{G_2} m_2 + (d_{G_2}^2 + L_{G_2}^2) m_2) \omega_2)))$$

$$a_{2,2} = \frac{1}{|H|} (- (L_1 (\cos(q_2) d_{G_2} - \sin(q_2) L_{G_2}) m_2 (I_{zG_2} + \sin(q_2) d_{G_2} L_1 m_2 + \cos(q_2) L_1 L_{G_2} m_2 + (d_{G_2}^2 + L_{G_2}^2) m_2) (\omega_1 + \omega_2))),$$

and for $\mathbf{B} = H^{-1}\mathbf{G}$ we have

$$\mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$$

with

$$b_1 = \frac{1}{|H|} (g (- ((\sin(q_1 + q_2) d_{G_2} + \cos(q_1 + q_2) L_{G_2}) m_2 (I_{zG_2} + \sin(q_2) d_{G_2} L_1 m_2 + \cos(q_2) L_1 L_{G_2} m_2 + (d_{G_2}^2 + L_{G_2}^2) m_2)) + (I_{zG_2} + (d_{G_2}^2 + L_{G_2}^2) m_2) (\sin(q_1) d_{G_1} m_1 + \sin(q_1 + q_2) d_{G_2} m_2 + \cos(q_1 + q_2) L_{G_2} m_2 + \cos(q_1) (L_{G_1} m_1 + L_1 m_2))))$$

$$b_2 = \frac{1}{|H|} (g ((\sin(q_1 + q_2) d_{G_2} + \cos(q_1 + q_2) L_{G_2}) m_2 (I_{zG_1} + I_{zG_2} + (d_{G_1}^2 + L_{G_1}^2) m_1 + 2 L_1 (\sin(q_2) d_{G_2} + \cos(q_2) L_{G_2}) m_2 + (d_{G_2}^2 + L_1^2 + L_{G_2}^2) m_2) - (I_{zG_2} + \sin(q_2) d_{G_2} L_1 m_2 + \cos(q_2) L_1 L_{G_2} m_2 + (d_{G_2}^2 + L_{G_2}^2) m_2) \cdot (\sin(q_1) d_{G_1} m_1 + \sin(q_1 + q_2) d_{G_2} m_2 + \cos(q_1 + q_2) L_{G_2} m_2 + \cos(q_1) (L_{G_1} m_1 + L_1 m_2))))).$$

C.2 Coordinate transformation

This appendix provides the details about the coordinate transformations (4.27)-(4.30). With $x_1 = [q_1 \ \omega_1 \ p_1 \ p_2]^T$ and $x_2 = [q_2 \ \omega_2 \ p_3 \ p_4]^T$, we have

$$\begin{aligned}
 \xi_{11} &= h_1(x_1) = q_1 \\
 \xi_{12} &= L_{f_1} h_1(x_1) = \omega_1 \\
 \xi_{13} &= L_{f_1}^2 h_1(x_1) \\
 &= \left(- (L_1^2 (\sin(q_2) d_{G_2} + \cos(q_2) L_{G_2})^2 m_2^2) + (I_{z_{G_1}} + (d_{G_1}^2 + L_{G_1}^2) m_1 + L_1^2 m_2) \right. \\
 &\quad \cdot (I_{z_{G_2}} + (d_{G_2}^2 + L_{G_2}^2) m_2) \left. \right)^{-1} \cdot \left(- ((I_{z_{G_2}} + (d_{G_2}^2 + L_{G_2}^2) m_2) \cdot \right. \\
 &\quad (g \sin(q_1 + q_2) d_{G_2} m_2 + g (\sin(q_1) d_{G_1} m_1 + \cos(q_1) L_{G_1} m_1 + \cos(q_1) L_1 m_2 \\
 &\quad + \cos(q_1 + q_2) L_{G_2} m_2) + L_1 (\cos(q_2) d_{G_2} - \sin(q_2) L_{G_2}) m_2 \omega_2 (2\omega_1 + \omega_2) \\
 &\quad - p_1 m_{\tau_1}(q_1) - p_2 m_{\tau_2}(q_1)) - (I_{z_{G_2}} + \sin(q_2) d_{G_2} L_1 m_2 + \cos(q_2) L_1 L_{G_2} m_2 \\
 &\quad + (d_{G_2}^2 + L_{G_2}^2) m_2) (- (m_2 (g \sin(q_1 + q_2) d_{G_2} + g \cos(q_1 + q_2) L_{G_2} \\
 &\quad + L_1 (-\cos(q_2) d_{G_2}) + \sin(q_2) L_{G_2}) \omega_1^2)) + p_3 m_{\tau_3}(q_2) + p_4 m_{\tau_4}(q_2) \left. \right) \\
 \eta_1 &= p_1 T_1 + p_2 T_2
 \end{aligned}$$

and

$$\begin{aligned}
 \xi_{21} &= h_2(x_2) = q_2 \\
 \xi_{22} &= L_{f_2} h_2(x_2) = \omega_2 \\
 \xi_{23} &= L_{f_2}^2 h_2(x_2) \\
 &= \left(- (L_1^2 (\sin(q_2) d_{G_2} + \cos(q_2) L_{G_2})^2 m_2^2) + (I_{z_{G_1}} + (d_{G_1}^2 + L_{G_1}^2) m_1 + \right. \\
 &\quad L_1^2 m_2) (I_{z_{G_2}} + (d_{G_2}^2 + L_{G_2}^2) m_2) \left. \right)^{-1} \cdot ((I_{z_{G_2}} + \sin(q_2) d_{G_2} L_1 m_2 + \\
 &\quad \cos(q_2) L_1 L_{G_2} m_2 + (d_{G_2}^2 + L_{G_2}^2) m_2) (g \sin(q_1 + q_2) d_{G_2} m_2 + \\
 &\quad g (\sin(q_1) d_{G_1} m_1 + \cos(q_1) L_{G_1} m_1 + \cos(q_1) L_1 m_2 + \cos(q_1 + q_2) L_{G_2} m_2) + \\
 &\quad L_1 (\cos(q_2) d_{G_2} - \sin(q_2) L_{G_2}) m_2 \omega_2 (2\omega_1 + \omega_2) - p_1 m_{\tau_1}(q_1) - p_2 m_{\tau_2}(q_1)) + \\
 &\quad (I_{z_{G_1}} + I_{z_{G_2}} + (d_{G_1}^2 + L_{G_1}^2) m_1 + 2 L_1 (\sin(q_2) d_{G_2} + \cos(q_2) L_{G_2}) m_2 + \\
 &\quad (d_{G_2}^2 + L_1^2 + L_{G_2}^2) m_2) (- (m_2 (g \sin(q_1 + q_2) d_{G_2} + g \cos(q_1 + q_2) L_{G_2} + \\
 &\quad L_1 (-\cos(q_2) d_{G_2}) + \sin(q_2) L_{G_2}) \omega_1^2)) + p_3 m_{\tau_3}(q_2) + p_4 m_{\tau_4}(q_2) \left. \right)
 \end{aligned}$$

$$\eta_2 = p_3 T_3 + p_4 T_4.$$

With $\phi_i = \left[h_i(x_i) \ L_{f_i} h_i(x_i) \ L_{f_i}^2 h_i(x_i) \ \eta_i \right]^T = [\xi_{i1} \ \xi_{i2} \ \xi_{i3} \ \eta_i]^T$ the determinant of the Jacobian matrix $\frac{\partial \phi_i}{\partial x_i}$ of both transformations becomes

$$\begin{aligned} \left| \frac{\partial \phi_1}{\partial x_1} \right| &= (-I_{zG_2} - (d_{G_2}^2 + L_{G_2}^2) m_2) \cdot (- (T_2 m_{\tau_1}(q_1)) + T_1 m_{\tau_2}(q_1)) \\ &\quad \cdot (- (L_1^2 (\sin(q_2) d_{G_2} + \cos(q_2) L_{G_2})^2 m_2^2) + \\ &\quad (I_{zG_1} + (d_{G_1}^2 + L_{G_1}^2) m_1 + L_1^2 m_2) (I_{zG_2} + (d_{G_2}^2 + L_{G_2}^2) m_2))^{-1} \\ \left| \frac{\partial \phi_2}{\partial x_2} \right| &= (I_{zG_1} + I_{zG_2} + (d_{G_1}^2 + L_{G_1}^2) m_1 + 2 L_1 (\sin(q_2) d_{G_2} + \cos(q_2) L_{G_2}) m_2 \\ &\quad + (d_{G_2}^2 + L_1^2 + L_{G_2}^2) m_2) \cdot (T_4 m_{\tau_3}(q_2) - T_3 m_{\tau_4}(q_2)) \\ &\quad \cdot (- (L_1^2 (\sin(q_2) d_{G_2} + \cos(q_2) L_{G_2})^2 m_2^2) + (I_{zG_1} + \\ &\quad (d_{G_1}^2 + L_{G_1}^2) m_1 + L_1^2 m_2) (I_{zG_2} + (d_{G_2}^2 + L_{G_2}^2) m_2))^{-1}. \end{aligned}$$

As can be seen in fig. 2.11, we have $m_{\tau_1} > 0$, $m_{\tau_2} < 0$, $m_{\tau_3} > 0$ and $m_{\tau_4} < 4$ in the entire manipulator working area ($\frac{\pi}{9} \leq q_1 \leq \frac{7\pi}{12}$, $-\frac{13\pi}{18} \leq q_2 \leq -\frac{\pi}{6}$). Since the time constants T_i are always positive, this means that both Jacobian matrices are nonsingular, and the coordinate transformations ϕ_1 and ϕ_2 are diffeomorphic within the working area. The inverse transformations ϕ_i^{-1} are given by

$$q_1 = \xi_{11}$$

$$\omega_1 = \xi_{12}$$

$$\begin{aligned} p_1 &= (- (T_2 ((I_{zG_1} + I_{zG_2} + (d_{G_1}^2 + L_{G_1}^2) m_1 + (d_{G_2}^2 + L_1^2 + L_{G_2}^2) m_2) \xi_{13} + \\ &\quad (I_{zG_2} + (d_{G_2}^2 + L_{G_2}^2) m_2) \xi_{23})) + T_2 (- (g \sin(\xi_{11}) d_{G_1} m_1) - \\ &\quad g \cos(\xi_{11}) (L_{G_1} m_1 + \sin(\xi_{21}) d_{G_2} m_2 + L_1 m_2 + \cos(\xi_{21}) L_{G_2} m_2) + \\ &\quad \sin(\xi_{21}) m_2 (g \sin(\xi_{11}) L_{G_2} + L_1 (L_{G_2} \xi_{22} (2 \xi_{12} + \xi_{22}) + d_{G_2} (-2 \xi_{13} - \xi_{23}))) - \\ &\quad \cos(\xi_{21}) m_2 (g \sin(\xi_{11}) d_{G_2} + L_1 (d_{G_2} \xi_{22} (2 \xi_{12} + \xi_{22}) + L_{G_2} (2 \xi_{13} + \xi_{23})))) + \\ &\quad \eta_1 m_{\tau_2}(\xi_{11})) \cdot (- (T_2 m_{\tau_1}(\xi_{11})) + T_1 m_{\tau_2}(\xi_{11}))^{-1} \end{aligned}$$

$$p_2 = \frac{-(p_1 T_1) + \eta_1}{T_2}$$

and

$$q_2 = \xi_{21}$$

$$\begin{aligned}
 \omega_2 &= \xi_{22} \\
 p_3 &= \left(- (m_2 T_4 (g \sin(\xi_{11} + \xi_{21}) d_{G_2} + g \cos(\xi_{11} + \xi_{21}) L_{G_2} + \sin(\xi_{21}) L_1 (L_{G_2} \xi_{12}^2 + d_{G_2} \xi_{13}) + \cos(\xi_{21}) L_1 (- (d_{G_2} \xi_{12}^2) + L_{G_2} \xi_{13}))) - \right. \\
 &\quad \left. (I_{z_{G_2}} + (d_{G_2}^2 + L_{G_2}^2) m_2) T_4 (\xi_{13} + \xi_{23}) + \eta_2 m_{\tau_4}(\xi_{21}) \right) \\
 &\quad \cdot (- (T_4 m_{\tau_3}(\xi_{21})) + T_3 m_{\tau_4}(\xi_{21}))^{-1} \\
 p_4 &= \frac{-(p_3 T_3) + \eta_2}{T_4}.
 \end{aligned}$$

The coefficients a_i , b_i and r_i (see eq. (4.33)-(4.34)) are given by

$$\begin{aligned}
 a_1 &= L_{g_1} L_{f_1}^2 h_1(x_1) \\
 &= \left(- \left((I_{z_{G_2}} + (d_{G_2}^2 + L_{G_2}^2) m_2) (- (T_2 m_{\tau_1}(q_1)) + T_1 m_{\tau_2}(q_1)) \right) \right) \\
 &\quad \cdot \left(- (L_1^2 (\sin(q_2) d_{G_2} + \cos(q_2) L_{G_2})^2 m_2^2) + \right. \\
 &\quad \left. (I_{z_{G_1}} + (d_{G_1}^2 + L_{G_1}^2) m_1 + L_1^2 m_2) (I_{z_{G_2}} + (d_{G_2}^2 + L_{G_2}^2) m_2) \right) T_1 T_2^{-1} \\
 b_1 &= L_{f_1}^3 h_1(x_1) \\
 &= (2 L_1 (\cos(q_2) d_{G_2} - \sin(q_2) L_{G_2}) m_2 (L_1 (\sin(q_2) d_{G_2} + \cos(q_2) L_{G_2}) m_2 \omega_1 + \\
 &\quad (I_{z_{G_2}} + (d_{G_2}^2 + L_{G_2}^2) m_2) (\omega_1 + \omega_2)) ((I_{z_{G_2}} + (d_{G_2}^2 + L_{G_2}^2) m_2) \\
 &\quad \cdot (g \sin(q_1 + q_2) d_{G_2} m_2 + g (\sin(q_1) d_{G_1} m_1 + \cos(q_1 + q_2) L_{G_2} m_2 + \\
 &\quad \cos(q_1) (L_{G_1} m_1 + L_1 m_2)) + L_1 (\cos(q_2) d_{G_2} - \sin(q_2) L_{G_2}) m_2 \omega_2 (2 \omega_1 + \omega_2) - \\
 &\quad p_1 m_{\tau_1}(q_1) - p_2 m_{\tau_2}(q_1)) + (I_{z_{G_2}} + \sin(q_2) d_{G_2} L_1 m_2 + \cos(q_2) L_1 L_{G_2} m_2 + \\
 &\quad (d_{G_2}^2 + L_{G_2}^2) m_2) (- (m_2 (g \sin(q_1 + q_2) d_{G_2} + g \cos(q_1 + q_2) L_{G_2} + \\
 &\quad L_1 (- (\cos(q_2) d_{G_2}) + \sin(q_2) L_{G_2}) \omega_1^2)) + p_3 m_{\tau_3}(q_2) + p_4 m_{\tau_4}(q_2))) - \\
 &\quad (- (L_1^2 (\sin(q_2) d_{G_2} + \cos(q_2) L_{G_2})^2 m_2^2) + (I_{z_{G_1}} + (d_{G_1}^2 + L_{G_1}^2) m_1 + \\
 &\quad L_1^2 m_2) (I_{z_{G_2}} + (d_{G_2}^2 + L_{G_2}^2) m_2)) \omega_1 (- (g (\cos(q_1 + q_2) d_{G_2} - \\
 &\quad \sin(q_1 + q_2) L_{G_2}) m_2 (I_{z_{G_2}} + \sin(q_2) d_{G_2} L_1 m_2 + \cos(q_2) L_1 L_{G_2} m_2 + \\
 &\quad (d_{G_2}^2 + L_{G_2}^2) m_2)) - (I_{z_{G_2}} + (d_{G_2}^2 + L_{G_2}^2) m_2) (g (- (\cos(q_1) d_{G_1} m_1) - \\
 &\quad \cos(q_1 + q_2) d_{G_2} m_2 + \sin(q_1 + q_2) L_{G_2} m_2 + \sin(q_1) (L_{G_1} m_1 + L_1 m_2)) + \\
 &\quad p_1 (m_{\tau_1})'(q_1) + p_2 (m_{\tau_2})'(q_1))) + (I_{z_{G_2}} + (d_{G_2}^2 + L_{G_2}^2) m_2) \\
 &\quad \cdot (- (L_1^2 (\sin(q_2) d_{G_2} + \cos(q_2) L_{G_2})^2 m_2^2) + (I_{z_{G_1}} + (d_{G_1}^2 + L_{G_1}^2) m_1 +
 \end{aligned}$$

$$\begin{aligned}
& L_1^2 m_2 \left(I_{zG_2} + (d_{G_2}^2 + L_{G_2}^2) m_2 \right) m_{\tau_1}(q_1) \\
& \cdot \left(\frac{-p_1 + p_m}{T_1} - \frac{n (patm + p_1) \omega_1 (V_1)'(q_1)}{V_1(q_1)} \right) + (I_{zG_2} + (d_{G_2}^2 + L_{G_2}^2) m_2) \\
& \cdot \left(- (L_1^2 (\sin(q_2) d_{G_2} + \cos(q_2) L_{G_2})^2 m_2^2) + (I_{zG_1} + (d_{G_1}^2 + L_{G_1}^2) m_1 + \right. \\
& \left. L_1^2 m_2) \left(I_{zG_2} + (d_{G_2}^2 + L_{G_2}^2) m_2 \right) \right) m_{\tau_2}(q_1) \\
& \cdot \left(\frac{-p_2 + p_m}{T_2} - \frac{n (patm + p_2) \omega_1 (V_2)'(q_1)}{V_2(q_1)} \right) \\
& \cdot \left(- (L_1^2 (\sin(q_2) d_{G_2} + \cos(q_2) L_{G_2})^2 m_2^2) + \right. \\
& \left. (I_{zG_1} + (d_{G_1}^2 + L_{G_1}^2) m_1 + L_1^2 m_2) \left(I_{zG_2} + (d_{G_2}^2 + L_{G_2}^2) m_2 \right) \right)^{-1} \\
r_1 &= L_{f_1} \eta_1(x_1) \\
&= - \left(-2 p_m V_1(q_1) V_2(q_1) + n P_{atm} T_1 \omega_1 V_2(q_1) (V_1)'(q_1) + \right. \\
& p_1 V_2(q_1) (V_1(q_1) + n T_1 \omega_1 (V_1)'(q_1)) + n P_{atm} T_2 \omega_1 V_1(q_1) (V_2)'(q_1) + \\
& \left. p_2 V_1(q_1) (V_2(q_1) + n T_2 \omega_1 (V_2)'(q_1)) \right) \cdot (V_1(q_1) V_2(q_1))^{-1}
\end{aligned}$$

and

$$\begin{aligned}
a_2 &= L_{g_2} L_{f_2}^2 h_2(x_2) \\
&= \left((I_{zG_1} + I_{zG_2} + (d_{G_1}^2 + L_{G_1}^2) m_1 + 2 L_1 (\sin(q_2) d_{G_2} + \cos(q_2) L_{G_2}) m_2 + \right. \\
& \left. (d_{G_2}^2 + L_1^2 + L_{G_2}^2) m_2) (T_4 m_{\tau_3}(q_2) - T_3 m_{\tau_4}(q_2)) \right) \\
& \cdot \left(- (L_1^2 (\sin(q_2) d_{G_2} + \cos(q_2) L_{G_2})^2 m_2^2) + \right. \\
& \left. (I_{zG_1} + (d_{G_1}^2 + L_{G_1}^2) m_1 + L_1^2 m_2) \left(I_{zG_2} + (d_{G_2}^2 + L_{G_2}^2) m_2 \right) \right) T_3 T_4^{-1} \\
b_2 &= L_{f_2}^3 h_2(x_2) \\
&= (2 L_1 (\cos(q_2) d_{G_2} - \sin(q_2) L_{G_2}) m_2 (I_{zG_2} + \sin(q_2) d_{G_2} L_1 m_2 + \\
& \cos(q_2) L_1 L_{G_2} m_2 + (d_{G_2}^2 + L_{G_2}^2) m_2) (\omega_1 + \omega_2) ((I_{zG_2} + \sin(q_2) d_{G_2} L_1 m_2 + \\
& \cos(q_2) L_1 L_{G_2} m_2 + (d_{G_2}^2 + L_{G_2}^2) m_2) (g \sin(q_1 + q_2) d_{G_2} m_2 + \\
& g (\sin(q_1) d_{G_1} m_1 + \cos(q_1 + q_2) L_{G_2} m_2 + \cos(q_1) (L_{G_1} m_1 + L_1 m_2)) + \\
& L_1 (\cos(q_2) d_{G_2} - \sin(q_2) L_{G_2}) m_2 \omega_2 (2 \omega_1 + \omega_2) - p_1 m_{\tau_1}(q_1) - p_2 m_{\tau_2}(q_1)) + \\
& \left. (I_{zG_1} + I_{zG_2} + (d_{G_1}^2 + L_{G_1}^2) m_1 + 2 L_1 (\sin(q_2) d_{G_2} + \cos(q_2) L_{G_2}) m_2 + \right.
\end{aligned}$$

$$\begin{aligned}
 & (d_{G_2}^2 + L_1^2 + L_{G_2}^2) m_2) (- (m_2 (g \sin(q_1 + q_2) d_{G_2} + g \cos(q_1 + q_2) L_{G_2} + \\
 & L_1 (- (\cos(q_2) d_{G_2}) + \sin(q_2) L_{G_2}) \omega_1^2)) + p_3 m_{\tau 3}(q_2) + p_4 m_{\tau 4}(q_2))) + \\
 & \omega_2 (2 L_1^2 (\sin(q_2) d_{G_2} + \cos(q_2) L_{G_2}) (\cos(q_2) d_{G_2} - \sin(q_2) L_{G_2}) \\
 & \cdot m_2^2 ((I_{zG_2} + \sin(q_2) d_{G_2} L_1 m_2 + \cos(q_2) L_1 L_{G_2} m_2 + (d_{G_2}^2 + L_{G_2}^2) m_2) \\
 & \cdot (g \sin(q_1 + q_2) d_{G_2} m_2 + g (\sin(q_1) d_{G_1} m_1 + \cos(q_1 + q_2) L_{G_2} m_2 + \\
 & \cos(q_1) (L_{G_1} m_1 + L_1 m_2)) + L_1 (\cos(q_2) d_{G_2} - \sin(q_2) L_{G_2}) m_2 \omega_2 (2 \omega_1 + \omega_2) - \\
 & p_1 m_{\tau 1}(q_1) - p_2 m_{\tau 2}(q_1)) + (I_{zG_1} + I_{zG_2} + (d_{G_1}^2 + L_{G_1}^2) m_1 + \\
 & 2 L_1 (\sin(q_2) d_{G_2} + \cos(q_2) L_{G_2}) m_2 + (d_{G_2}^2 + L_1^2 + L_{G_2}^2) m_2) \\
 & \cdot (- (m_2 (g \sin(q_1 + q_2) d_{G_2} + g \cos(q_1 + q_2) L_{G_2} + L_1 (- (\cos(q_2) d_{G_2}) + \\
 & \sin(q_2) L_{G_2}) \omega_1^2)) + p_3 m_{\tau 3}(q_2) + p_4 m_{\tau 4}(q_2))) + \\
 & (- (L_1^2 (\sin(q_2) d_{G_2} + \cos(q_2) L_{G_2})^2 m_2^2) + (I_{zG_1} + (d_{G_1}^2 + L_{G_1}^2) m_1 + \\
 & L_1^2 m_2) (I_{zG_2} + (d_{G_2}^2 + L_{G_2}^2) m_2)) (m_2 (I_{zG_2} + \sin(q_2) d_{G_2} L_1 m_2 + \\
 & \cos(q_2) L_1 L_{G_2} m_2 + (d_{G_2}^2 + L_{G_2}^2) m_2) (g \cos(q_1 + q_2) d_{G_2} - g \sin(q_1 + q_2) L_{G_2} - \\
 & L_1 (\sin(q_2) d_{G_2} + \cos(q_2) L_{G_2}) \omega_2 (2 \omega_1 + \omega_2)) + L_1 (\cos(q_2) d_{G_2} - \\
 & \sin(q_2) L_{G_2}) m_2 (g \sin(q_1 + q_2) d_{G_2} m_2 + g (\sin(q_1) d_{G_1} m_1 + \cos(q_1 + q_2) L_{G_2} m_2 + \\
 & \cos(q_1) (L_{G_1} m_1 + L_1 m_2)) + L_1 (\cos(q_2) d_{G_2} - \sin(q_2) L_{G_2}) m_2 \omega_2 (2 \omega_1 + \omega_2) - \\
 & p_1 m_{\tau 1}(q_1) - p_2 m_{\tau 2}(q_1)) + 2 L_1 (\cos(q_2) d_{G_2} - \sin(q_2) L_{G_2}) \\
 & \cdot m_2 (- (m_2 (g \sin(q_1 + q_2) d_{G_2} + g \cos(q_1 + q_2) L_{G_2} + \\
 & L_1 (- (\cos(q_2) d_{G_2}) + \sin(q_2) L_{G_2}) \omega_1^2)) + p_3 m_{\tau 3}(q_2) + p_4 m_{\tau 4}(q_2)) + \\
 & (I_{zG_1} + I_{zG_2} + (d_{G_1}^2 + L_{G_1}^2) m_1 + 2 L_1 (\sin(q_2) d_{G_2} + \cos(q_2) L_{G_2}) m_2 + \\
 & (d_{G_2}^2 + L_1^2 + L_{G_2}^2) m_2) (- (m_2 (g \cos(q_1 + q_2) d_{G_2} - g \sin(q_1 + q_2) L_{G_2} + \\
 & L_1 (\sin(q_2) d_{G_2} + \cos(q_2) L_{G_2}) \omega_1^2)) + p_3 (m_{\tau 3})'(q_2) + p_4 (m_{\tau 4})'(q_2))) - \\
 & ((I_{zG_1} + I_{zG_2} + (d_{G_1}^2 + L_{G_1}^2) m_1 + 2 L_1 (\sin(q_2) d_{G_2} + \cos(q_2) L_{G_2}) m_2 + \\
 & (d_{G_2}^2 + L_1^2 + L_{G_2}^2) m_2) (- (L_1^2 (\sin(q_2) d_{G_2} + \cos(q_2) L_{G_2})^2 m_2^2) + \\
 & (I_{zG_1} + (d_{G_1}^2 + L_{G_1}^2) m_1 + L_1^2 m_2) (I_{zG_2} + (d_{G_2}^2 + L_{G_2}^2) m_2)) m_{\tau 3}(q_2) \\
 & \cdot ((p_3 - p_m) V_3(q_2) + n (p_3 + P_{atm}) T_3 \omega_2 (V_3)'(q_2))) \cdot (T_3 V_3(q_2))^{-1} -
 \end{aligned}$$

$$\begin{aligned}
& ((I_{zG_1} + I_{zG_2} + (d_{G_1}^2 + L_{G_1}^2) m_1 + 2 L_1 (\sin(q_2) d_{G_2} + \cos(q_2) L_{G_2}) m_2 + \\
& (d_{G_2}^2 + L_1^2 + L_{G_2}^2) m_2) (- (L_1^2 (\sin(q_2) d_{G_2} + \cos(q_2) L_{G_2})^2 m_2^2) + \\
& (I_{zG_1} + (d_{G_1}^2 + L_{G_1}^2) m_1 + L_1^2 m_2) (I_{zG_2} + (d_{G_2}^2 + L_{G_2}^2) m_2)) m_{\tau_4}(q_2) \\
& \cdot ((p_4 - p_m) V_4(q_2) + n (p_4 + P_{atm}) T_4 \omega_2 (V_4)'(q_2))) \cdot (T_4 V_4(q_2))^{-1}) \\
& \cdot ((- (L_1^2 (\sin(q_2) d_{G_2} + \cos(q_2) L_{G_2})^2 m_2^2) + \\
& (I_{zG_1} + (d_{G_1}^2 + L_{G_1}^2) m_1 + L_1^2 m_2) (I_{zG_2} + (d_{G_2}^2 + L_{G_2}^2) m_2))^2)^{-1} \\
r_2 &= L_{f_2} \eta_2(x_2) \\
&= -(-2 p_m V_3(q_2) V_4(q_2) + n P_{atm} T_3 \omega_2 V_4(q_2) (V_3)'(q_2) + p_3 V_4(q_2) (V_3(q_2) + \\
& n T_3 \omega_2 (V_3)'(q_2)) + n P_{atm} T_4 \omega_2 V_3(q_2) (V_4)'(q_2) + p_4 V_3(q_2) (V_4(q_2) + \\
& n T_4 \omega_2 (V_4)'(q_2))) \cdot (V_3(q_2) V_4(q_2))^{-1}
\end{aligned}$$

where ' stands for derivative (for instance $V_1'(q_1) = \frac{dV_1(q_1)}{dq_1}$).

It was numerically verified that a_1 and a_2 stay far away from zero in the entire working area ($\frac{\pi}{9} \leq q_1 \leq \frac{7\pi}{12}$, $-\frac{13\pi}{18} \leq q_2 \leq -\frac{\pi}{6}$), so the strict relative degree (see eqs. (4.25)-(4.26)) of both SISO systems is always defined.

In the expressions above, a_i , b_i and r_i are expressed as functions of the state vectors x_i . If desired, they can be expressed in terms of the new coordinates ξ_{ij} , η_i by using the inverse transformations ϕ_i^{-1} . If we do this for r_1 and r_2 , while setting all ξ_{ij} to zero, (4.34) gives the zero dynamics (Sastry, 1999) of both systems:

$$\begin{aligned}
\dot{\eta}_1 &= r_1(0, \eta_1) \\
&= \frac{-2m_{\tau_2}(0)p_m T_1 + 2m_{\tau_1}(0)p_m T_2 + g(L_{G_1} m_1 + (L_1 + L_{G_2}) m_2) (T_1 - T_2)}{T_2 m_{\tau_1}(0) - T_1 m_{\tau_2}(0)} \\
&\quad - \frac{m_{\tau_1}(0) - m_{\tau_2}(0)}{T_2 m_{\tau_1}(0) - T_1 m_{\tau_2}(0)} \eta_1 \\
&= \alpha_1 - \beta_1 \eta_1 \\
\dot{\eta}_2 &= r_2(0, \eta_2) \\
&= \frac{-2m_{\tau_4}(0)p_m T_3 + 2m_{\tau_3}(0)p_m T_4 + g L_{G_2} m_2 (T_3 - T_4)}{T_4 m_{\tau_3}(0) - T_3 m_{\tau_4}(0)} \\
&\quad - \frac{m_{\tau_3}(0) - m_{\tau_4}(0)}{T_4 m_{\tau_3}(0) - T_3 m_{\tau_4}(0)} \eta_2
\end{aligned}$$

$$= \alpha_2 - \beta_2 \eta_2,$$

where α_i and β_i are defined by the above equations. We see that the zero dynamics is linear, and stable if $\beta_i > 0$, which is always the case since $m_{\tau_1}(0) > 0$, $m_{\tau_2}(0) < 0$, $m_{\tau_3}(0) > 0$, $m_{\tau_4}(0) < 0$ and all $T_i > 0$. Both SISO systems are thus minimum phase.

Appendix D

Proxy-Based Sliding Mode Control

D.1 Signum function and unit saturation function

In this section, theorem 1 from Kikuuwe and Fujimoto (2006) is reproduced. It is based on definition (5.1) of the $\text{sgn}(\cdot)$ function, which is repeated here for convenience:

$$\text{sgn}(x) \begin{cases} = 1 & \text{if } x > 0 \\ \in [-1, 1] & \text{if } x = 0 \\ = -1 & \text{if } x < 0. \end{cases} \quad (\text{D.1})$$

Theorem 1 (Kikuuwe and Fujimoto, 2006). *With two real numbers x and y , the following statement holds true:*

$$y = \text{sgn}(x - y) \iff y = \text{sat}(x), \quad (\text{D.2})$$

where $\text{sat}(\cdot)$ is the unit saturation function, which is defined as

$$\text{sat}(x) = \begin{cases} x & \text{if } |x| \leq 1 \\ \text{sgn}(x) & \text{if } |x| > 1. \end{cases} \quad (\text{D.3})$$

Proof. The statement $y = \text{sgn}(x - y)$ is equivalent to

$$(y = 1 \wedge x - y > 0) \vee (y \in [-1, 1] \wedge x - y = 0) \vee (y = -1 \wedge x - y < 0). \quad (\text{D.4})$$

Each term in (D.4) can be rewritten as follows:

$$\begin{aligned} (y = 1 \wedge x - y > 0) &\iff (y = 1 \wedge x > 1) \\ (y \in [-1, 1] \wedge x - y = 0) &\iff (y = x \wedge x \in [-1, 1]) \\ (y = -1 \wedge x - y < 0) &\iff (y = -1 \wedge x < -1). \end{aligned}$$

Therefore, (D.4) is equivalent to $y = \text{sat}(x)$. This means that $y = \text{sgn}(x - y)$ is equivalent to $y = \text{sat}(x)$. \square

In order to use the above in a multi-input, multi-output case, we need a vectorial version of (D.2), i.e.

$$\mathbf{y} = \text{sgn}(\mathbf{x} - \mathbf{y}) \iff \mathbf{y} = \text{sat}(\mathbf{x}). \quad (\text{D.5})$$

It is clear that the above doesn't hold when using the most straightforward vectorial definition of the signum function, $\text{sgn}(\mathbf{x}) = [\text{sgn}(x_1) \ \cdots \ \text{sgn}(x_n)]^T$ with $\mathbf{x} = [x_1 \ \cdots \ x_n]^T$ (which is used in for instance Slotine and Li (1991)), since it doesn't conserve direction. In order to see how $\text{sgn}(\cdot)$ and $\text{sat}(\cdot)$ can be generalized, we can rewrite (D.1) as follows:

$$\text{sgn}(x) = \begin{cases} = x/|x| & \text{if } x \neq 0 \\ \in [-1, 1] & \text{if } x = 0. \end{cases}$$

$\text{sat}(x)$ can then be written as

$$\text{sat}(x) = \begin{cases} x & \text{if } |x| \leq 1 \\ x/|x| & \text{if } |x| > 1. \end{cases}$$

From these definitions, it is more straightforward to generalize them as (with $\mathbf{x} \in \mathbb{R}^n$)

$$\text{sgn}(\mathbf{x}) = \begin{cases} = \mathbf{x}/\|\mathbf{x}\| & \text{if } \mathbf{x} \neq \mathbf{0} \\ \in \{\mathbf{e} \in \mathbb{R}^n \mid \|\mathbf{e}\| \leq 1\} & \text{if } \mathbf{x} = \mathbf{0} \end{cases} \quad (\text{D.6})$$

and

$$\text{sat}(\mathbf{x}) = \begin{cases} \mathbf{x} & \text{if } \|\mathbf{x}\| \leq 1 \\ \mathbf{x}/\|\mathbf{x}\| & \text{if } \|\mathbf{x}\| > 1. \end{cases} \quad (\text{D.7})$$

Using these definitions (D.5) holds, as is shown in Kikuuwe et al. (2006).

D.2 Massless proxy

In proxy-based sliding mode control, the proxy is assumed to be massless. In this section, we will briefly look into some of the effects of having a massless proxy.

Let's first assume that there is no integral component in the virtual coupling, and that the sliding mode controller is absent (i.e. $\mathbf{F}_a = \mathbf{0}$). The equations of motion of the proxy then simplify to (using (5.11) and (5.6))

$$m\ddot{\mathbf{r}}_p = -K_p(\mathbf{r}_p - \mathbf{r}) - K_d(\dot{\mathbf{r}}_p - \dot{\mathbf{r}}). \quad (\text{D.8})$$

If n is the dimension of the vectors used in the above equation (i.e. $n = 1, 2$ or 3), the order of the proxy's dynamics is $2n$. If we set the proxy mass m to zero, eq. (D.8) can be rewritten as

$$\dot{\mathbf{r}}_p = -\frac{K_p}{K_d}(\mathbf{r}_p - \mathbf{r}) + \dot{\mathbf{r}}. \quad (\text{D.9})$$

By assuming a massless proxy, the order of the system has been reduced by n and the velocity vector $\dot{\mathbf{r}}_p$ is no longer part of the proxy's state. Since there is no coupling between components, the above system is equivalent to n uncoupled first-order systems.

In the general case (i.e. with the sliding mode controller and with a PID-type virtual coupling) the virtual coupling has an integral part, which means it has its own state. This state was written in eq. (5.7) as

$$\mathbf{a} = \int (\mathbf{r}_p - \mathbf{r}) dt.$$

Both \mathbf{F}_a and \mathbf{F}_c can be expressed as a function of \mathbf{a} and its derivatives, as in eqs. (5.10) and (5.8), so the equations of motion of the proxy can be written as (see eq. (5.11))

$$m\ddot{\mathbf{r}}_p = \mathbf{F}_a(\mathbf{r}, \dot{\mathbf{r}}, \dot{\mathbf{a}}, \ddot{\mathbf{a}}) - \mathbf{F}_c(\mathbf{r}, \dot{\mathbf{r}}, \mathbf{a}, \dot{\mathbf{a}}, \ddot{\mathbf{a}}).$$

By setting

$$\begin{aligned} \mathbf{x}_1 &= \mathbf{a} \\ &= \int (\mathbf{r}_p - \mathbf{r}) dt \\ \mathbf{x}_2 &= \dot{\mathbf{a}} \\ &= \mathbf{r}_p - \mathbf{r} \\ \mathbf{x}_3 &= \ddot{\mathbf{a}} \\ &= \dot{\mathbf{r}}_p - \dot{\mathbf{r}} \end{aligned}$$

we can write the dynamic equations of the proxy-virtual coupling system in state space form:

$$\begin{aligned} \dot{\mathbf{x}}_1 &= \mathbf{x}_2 \\ \dot{\mathbf{x}}_2 &= \mathbf{x}_3 \\ m\dot{\mathbf{x}}_3 &= \mathbf{F}_a(\mathbf{r}, \dot{\mathbf{r}}, \mathbf{x}_2, \mathbf{x}_3) - \mathbf{F}_c(\mathbf{r}, \dot{\mathbf{r}}, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3) - m\ddot{\mathbf{r}}. \end{aligned}$$

The order of this system is $3n$ ($2n$ states belonging the proxy and n states to the virtual coupling). If we set the proxy mass m equal to zero, the system reduces to

$$\dot{\mathbf{x}}_1 = \mathbf{x}_2 \quad (\text{D.10})$$

$$\mathbf{F}_a(\mathbf{r}, \dot{\mathbf{r}}, \mathbf{x}_2, \dot{\mathbf{x}}_2) = \mathbf{F}_c(\mathbf{r}, \dot{\mathbf{r}}, \mathbf{x}_1, \mathbf{x}_2, \dot{\mathbf{x}}_2) \quad (\text{D.11})$$

Since eq. (D.11) implicitly defines $\dot{\mathbf{x}}_2$, it determines the dynamics of the proxy. If needed, $\dot{\mathbf{x}}_2$ can be isolated using the procedure used in section 5.2.5.

Eqs. (D.10)-(D.11) show some of the consequences of considering a massless proxy:

- The order of the proxy-virtual coupling system is reduced by n .
- The proxy is always in force-equilibrium, which means the sliding mode force \mathbf{F}_a is always balanced by the force \mathbf{F}_c resulting from the virtual coupling.
- The force-equilibrium constraint directly determines the dynamics of the proxy, since it (implicitly) defines $\dot{\mathbf{x}}_2$.
- The behavior of the massless proxy cannot easily be determined by numerical integration of eqs. (D.10)-(D.11), due to the difficulty in isolating $\dot{\mathbf{x}}_2$.

D.3 Discrete-time controller equations

This section presents the detailed calculations of how to solve eqs. (5.19)-(5.20) for $\mathbf{f}[k]$ and $\mathbf{a}[k]$. For easier reference, (5.19)-(5.20) are repeated here:

$$\mathbf{f}[k] = F \text{sgn} \left(\boldsymbol{\sigma}[k] - \frac{\nabla \mathbf{a}[k]}{T} - \lambda \frac{\nabla^2 \mathbf{a}[k]}{T^2} \right) \quad (\text{D.12})$$

$$\mathbf{f}[k] = K_p \frac{\nabla \mathbf{a}[k]}{T} + K_i \mathbf{a}[k] + K_d \frac{\nabla^2 \mathbf{a}[k]}{T^2}. \quad (\text{D.13})$$

We start by using eqs. (5.17) and (5.18) to rewrite (D.13) as

$$\begin{aligned} \mathbf{f}[k] &= \frac{K_p}{T} \mathbf{a}[k] - \frac{K_p}{T} \mathbf{a}[k-1] + K_i \mathbf{a}[k] + \frac{K_d}{T^2} (\nabla \mathbf{a}[k] - \nabla \mathbf{a}[k-1]) \\ &= \mathbf{a}[k] \left(\frac{K_p}{T} + K_i \right) - \frac{K_p}{T} \mathbf{a}[k-1] + \frac{K_d}{T^2} \mathbf{a}[k] - \frac{K_d}{T^2} \mathbf{a}[k-1] \\ &\quad - \frac{K_d}{T^2} \nabla \mathbf{a}[k-1] \\ &= \mathbf{a}[k] \left(\frac{K_p}{T} + K_i + \frac{K_d}{T^2} \right) - \mathbf{a}[k-1] \left(\frac{K_p}{T} + \frac{K_d}{T^2} \right) - \frac{K_d}{T^2} \nabla \mathbf{a}[k-1] \\ &= \mathbf{a}[k] \frac{K_i T^2 + K_p T + K_d}{T^2} - \mathbf{a}[k-1] \frac{K_p T + K_d}{T^2} - \frac{K_d}{T^2} \nabla \mathbf{a}[k-1]. \end{aligned}$$

$\mathbf{a}[k]$ can now be written as a function of $\mathbf{f}[k]$ and past values of \mathbf{a} :

$$\mathbf{a}[k] = \frac{T^2}{K_i T^2 + K_p T + K_d} \left(\mathbf{f}[k] + \mathbf{a}[k-1] \frac{K_p T + K_d}{T^2} + \frac{K_d}{T^2} \nabla \mathbf{a}[k-1] \right)$$

$$= \frac{\mathbf{a}[k-1](K_p T + K_d) + K_d \nabla \mathbf{a}[k-1] + T^2 \mathbf{f}[k]}{K_i T^2 + K_p T + K_d} \quad (\text{D.14})$$

By rewriting (D.12) using (5.17) and (5.18) we get

$$\begin{aligned} \mathbf{f}[k] &= F \text{sgn} \left(\boldsymbol{\sigma}[k] - \frac{1}{T} \mathbf{a}[k] + \frac{1}{T} \mathbf{a}[k-1] - \frac{\lambda}{T^2} (\nabla \mathbf{a}[k] - \nabla \mathbf{a}[k-1]) \right) \\ &= F \text{sgn} \left(\boldsymbol{\sigma}[k] - \frac{1}{T} \mathbf{a}[k] + \frac{1}{T} \mathbf{a}[k-1] - \frac{\lambda}{T^2} (\mathbf{a}[k] - \mathbf{a}[k-1] - \nabla \mathbf{a}[k-1]) \right) \\ &= F \text{sgn} \left(\boldsymbol{\sigma}[k] - \frac{T+\lambda}{T^2} \mathbf{a}[k] + \frac{T+\lambda}{T^2} \mathbf{a}[k-1] + \frac{\lambda}{T^2} \nabla \mathbf{a}[k-1] \right). \end{aligned}$$

Substitution of (D.14) gives

$$\begin{aligned} \mathbf{f}[k] &= F \text{sgn} \left(\boldsymbol{\sigma}[k] + \frac{T+\lambda}{T^2} \mathbf{a}[k-1] + \frac{\lambda}{T^2} \nabla \mathbf{a}[k-1] \right. \\ &\quad \left. - \frac{T+\lambda}{T^2} \cdot \frac{\mathbf{a}[k-1](K_p T + K_d) + K_d \nabla \mathbf{a}[k-1] + T^2 \mathbf{f}[k]}{K_i T^2 + K_p T + K_d} \right) \quad (\text{D.15}) \end{aligned}$$

We see that $\mathbf{f}[k]$ appears on both sides in this equation. In order to solve for $\mathbf{f}[k]$, (D.5) can be used if we rewrite (D.15) as $\mathbf{f}[k] = F \text{sgn}(\mathbf{f}^*[k] - \mathbf{f}[k])$:

$$\begin{aligned} \mathbf{f}[k] &= F \text{sgn} \left(\boldsymbol{\sigma}[k] + \frac{T+\lambda}{T^2} \mathbf{a}[k-1] + \frac{\lambda}{T^2} \nabla \mathbf{a}[k-1] \right. \\ &\quad \left. - \frac{(T+\lambda)(K_p T + K_d)}{T^2(K_i T^2 + K_p T + K_d)} \mathbf{a}[k-1] - \frac{(T+\lambda)K_d}{T^2(K_i T^2 + K_p T + K_d)} \nabla \mathbf{a}[k-1] \right. \\ &\quad \left. - \frac{T+\lambda}{K_i T^2 + K_p T + K_d} \mathbf{f}[k] \right) \\ &= F \text{sgn} \left(\frac{T+\lambda}{K_i T^2 + K_p T + K_d} \left(\frac{K_i T^2 + K_p T + K_d}{T+\lambda} \boldsymbol{\sigma}[k] \right. \right. \\ &\quad \left. \left. + \frac{K_i T^2 + K_p T + K_d}{T^2} \mathbf{a}[k-1] + \frac{\lambda(K_i T^2 + K_p T + K_d)}{T^2(T+\lambda)} \nabla \mathbf{a}[k-1] \right. \right. \\ &\quad \left. \left. - \frac{K_p T + K_d}{T^2} \mathbf{a}[k-1] - \frac{K_d}{T^2} \nabla \mathbf{a}[k-1] - \mathbf{f}[k] \right) \right) \\ &= F \text{sgn} \left(\frac{T+\lambda}{K_i T^2 + K_p T + K_d} \left(\frac{K_i T^2 + K_p T + K_d}{T+\lambda} \boldsymbol{\sigma}[k] + K_i \mathbf{a}[k-1] \right. \right. \\ &\quad \left. \left. + \frac{\lambda(K_i T^2 + K_p T + K_d) - K_d(T+\lambda)}{T^2(T+\lambda)} \nabla \mathbf{a}[k-1] - \mathbf{f}[k] \right) \right) \end{aligned}$$

$$\begin{aligned}
&= F \mathbf{sgn} \left(\frac{T + \lambda}{K_i T^2 + K_p T + K_d} \left(\frac{K_i T^2 + K_p T + K_d}{T + \lambda} \boldsymbol{\sigma}[k] + K_i \mathbf{a}[k-1] \right. \right. \\
&\quad \left. \left. + \frac{\lambda(K_i T + K_p) - K_d}{T(T + \lambda)} \nabla \mathbf{a}[k-1] - \mathbf{f}[k] \right) \right) \\
&= F \mathbf{sgn} \left(\frac{T + \lambda}{K_i T^2 + K_p T + K_d} (\mathbf{f}^*[k] - \mathbf{f}[k]) \right), \tag{D.16}
\end{aligned}$$

with

$$\mathbf{f}^*[k] = \frac{K_i T^2 + K_p T + K_d}{T + \lambda} \boldsymbol{\sigma}[k] + K_i \mathbf{a}[k-1] + \frac{\lambda(K_i T + K_p) - K_d}{T(T + \lambda)} \nabla \mathbf{a}[k-1]. \tag{D.17}$$

Since

$$\frac{T + \lambda}{K_i T^2 + K_p T + K_d} > 0$$

(D.16) is equivalent to

$$\mathbf{f}[k] = F \mathbf{sgn}(\mathbf{f}^*[k] - \mathbf{f}[k]), \tag{D.18}$$

where $\mathbf{f}^*[k]$ only depends on past values of the state \mathbf{a} and on measured and known quantities (through $\boldsymbol{\sigma}[k]$, see eq. (5.21)), and can thus easily be calculated.

Since $F > 0$, (D.18) implies

$$\begin{aligned}
\frac{\mathbf{f}[k]}{F} &= \mathbf{sgn}(\mathbf{f}^*[k] - \mathbf{f}[k]) \\
&= \mathbf{sgn} \left(\frac{\mathbf{f}^*[k]}{F} - \frac{\mathbf{f}[k]}{F} \right).
\end{aligned}$$

Because of (D.5) and (D.7) this becomes

$$\begin{aligned}
\mathbf{f}[k] &= F \mathbf{sat} \left(\frac{\mathbf{f}^*[k]}{F} \right) \\
&= \begin{cases} \mathbf{f}^*[k] & \text{if } \|\mathbf{f}^*[k]\| \leq F \\ F \frac{\mathbf{f}^*[k]}{\|\mathbf{f}^*[k]\|} & \text{if } \|\mathbf{f}^*[k]\| > F. \end{cases} \tag{D.19}
\end{aligned}$$

In order to calculate the PSMC controller's output $\mathbf{f}[k]$ at timestep k , we first calculate $\boldsymbol{\sigma}[k]$ from (5.21). Knowing $\boldsymbol{\sigma}[k]$ and the state variable \mathbf{a} at the previous two timesteps (i.e. $\mathbf{a}[k-1]$ and $\mathbf{a}[k-2]$), we can determine $\mathbf{f}^*[k]$ from (D.17). Output $\mathbf{f}[k]$ is then calculated from (D.19). Once $\mathbf{f}[k]$ is known, state $\mathbf{a}[k]$ (which will be necessary in the next timestep) can be found from eq. (D.14).

By comparing (D.16) with (5.4), we see that $\mathbf{s}[k]$ is given by

$$\mathbf{s}[k] = \frac{T + \lambda}{K_i T^2 + K_p T + K_d} (\mathbf{f}^*[k] - \mathbf{f}[k]). \tag{D.20}$$

If $s[k] = 0$ the proxy is said to be “on its sliding surface” (see also the next section). According to (D.20) this is the case if $\mathbf{f}^*[k] = \mathbf{f}[k]$. From (D.19), we see that the proxy is on the sliding surface if

$$\|\mathbf{f}^*[k]\| \leq F.$$

D.4 The proxy in state-space

The goal of sliding mode control is to impose a certain dynamics (as described by the sliding surface) onto the controlled system. If we restrict ourselves to the single-input single-output (SISO) case, then the imposed dynamics usually has order $n-1$ if the system has order n (Slotine and Li, 1991). This was the case in the example of section 4.2.1, for instance. It was also the case for the “heavy”¹ proxy of section 5.2.3.

In the one-dimensional case (when the system to be controlled is a 1-DOF SISO system, for instance), the equation of motion of the proxy (5.11) can be written as

$$m\ddot{x}_p = F_a - F_c.$$

The (heavy) proxy dynamics is thus described by a second order system. The state vector can be chosen as $\mathbf{x} = [x_p \ \dot{x}_p]$. The dynamics imposed by the sliding mode controller, $s(\mathbf{x}, t) = 0$ with $s(\mathbf{x}, t)$ given by

$$s(\mathbf{x}, t) = (x_d - x_p) + \lambda(\dot{x}_d - \dot{x}_p),$$

is first order. $s(\mathbf{x}, t) = 0$ defines a (one-dimensional) line in the (two-dimensional) state-space.

In section D.2, we saw that the order of the proxy’s dynamics decreases when its mass is set to zero. For the one-dimensional case considered here, this means that the proxy’s dynamics become first order², defined by $F_a(\dot{x}_p) = F_c(\dot{x}_p)$. Since the velocity \dot{x}_p is no longer part of the state, the proxy position x_p is the only state variable left, and the proxy’s state-space has become one-dimensional. Strictly speaking, $s(\mathbf{x}, t) = 0$ can no longer be considered to be a sliding surface for the massless proxy. It still describes the desired dynamics for the proxy, though. Saying that the proxy is “on the sliding surface” can be interpreted as it accomplishing its desired dynamics (i.e. the control goal). Saying that it is “off the sliding surface” then means that its dynamics don’t match the desired dynamics. In the case of PSMC, this happens when a control input higher than the imposed limit F is required to achieve the desired dynamics.

¹Before it was assumed to have zero mass.

²If the virtual coupling has an integral component, it has its own state and the dynamics of the proxy-virtual coupling system become second order (with one state belonging to the coupling and one to the proxy).

Since the calculations for the real PSMC controller are quite involved, we'll illustrate this with a much simpler example. Assume we have the following first-order system,

$$\dot{x} = u, \tag{D.21}$$

with u the control input. Suppose that the desired dynamics are given by $s(x, t) = 0$ with

$$s(x, t) = (x_d - x) + \lambda(\dot{x}_d - \dot{x}).$$

A control law that achieves this can simply be found by substituting the dynamic equation $\dot{x} = u$ into $s(x, t) = 0$ and solving for u :

$$(x_d - x) + \lambda(\dot{x}_d - u) = 0$$

or

$$u = \frac{1}{\lambda}(x_d - x) + \dot{x}_d. \tag{D.22}$$

Now assume that, as in PSMC, we want to impose the desired dynamics $s(x, t) = 0$ by a controller of the form

$$u = B \operatorname{sgn}(s). \tag{D.23}$$

(with $B > 0$). Note that (strictly speaking) we can't call this a sliding mode controller, since there is no "sliding mode" when $s(x, t) = 0$ (it can be called a variable structure controller, though). Substitution of (D.23) in (D.21) results in

$$\begin{aligned} \dot{x} &= B \operatorname{sgn}((x_d - x) + \lambda\dot{x}_d - \lambda\dot{x}) \\ &= B \operatorname{sgn}\left(\lambda\left(\frac{1}{\lambda}(x_d - x) + \dot{x}_d - \dot{x}\right)\right). \end{aligned}$$

By calling

$$u^* = \frac{1}{\lambda}(x_d - x) + \dot{x}_d \tag{D.24}$$

and knowing that $\lambda > 0$ and $B > 0$ we can write this as

$$\begin{aligned} \frac{\dot{x}}{B} &= \operatorname{sgn}(u^* - \dot{x}) \\ &= \operatorname{sgn}\left(\frac{u^*}{B} - \frac{\dot{x}}{B}\right). \end{aligned}$$

We can now apply (D.2), which results in

$$\begin{aligned} \dot{x} &= B \operatorname{sat}\left(\frac{u^*}{B}\right) \\ &= \begin{cases} u^* & \text{if } |u^*| \leq B \\ B \operatorname{sgn}(u^*) & \text{if } |u^*| > B. \end{cases} \end{aligned}$$

Comparing this with (D.21) shows that control law (D.23) is equivalent to

$$u = \begin{cases} u^* & \text{if } |u^*| \leq B \\ B \operatorname{sgn}(u^*) & \text{if } |u^*| > B. \end{cases} \quad (\text{D.25})$$

Since u^* (see eq. (D.24)) is equal to the previous control law (D.22), we see that (D.25) is equivalent to (D.22), except that (D.25) limits the absolute value of the control input to B . If the control input needed to achieve the desired dynamics $s(x, t) = 0$ is higher (in absolute value) than B , the control goal will not be achieved. Since $s(x, t)$ is not zero, we can interpret this as the system state not being “on the sliding surface”.

The example uses equivalence (D.2), on which proxy-based sliding mode control is based, but without discretization. Let us now make the example slightly more complex, and assume the system is given by

$$\dot{x} = -x + u.$$

Substitution of control law (D.23) gives

$$\dot{x} + x = B \operatorname{sgn}((x_d - x) + \lambda(\dot{x}_d - \dot{x})).$$

Due to the presence of x on the left-hand side, we can't immediately use eq. (D.2) to isolate \dot{x} . One possibility is to use the procedure described in section 5.2.5. However, we can also set

$$\begin{aligned} u &= \dot{x} + x \\ u &= B \operatorname{sgn}((x_d - x) + \lambda(\dot{x}_d - \dot{x})) \end{aligned}$$

and discretize to get (using (5.16))

$$u[k] = \frac{\nabla x[k]}{T} + x[k] \quad (\text{D.26})$$

$$u[k] = B \operatorname{sgn}\left(x_d[k] - x[k] + \frac{\lambda}{T}(\nabla x_d - \nabla x)\right). \quad (\text{D.27})$$

From eq. (D.26) we have (cf. (5.17))

$$\begin{aligned} u[k] &= \frac{1}{T}x[k] - \frac{1}{T}x[k-1] + x[k] \\ &= \frac{1+T}{T}x[k] - \frac{1}{T}x[k-1] \end{aligned}$$

or

$$x[k] = \frac{1}{1+T}x[k-1] + \frac{T}{1+T}u[k]. \quad (\text{D.28})$$

Eq. (D.27) can be written as

$$\begin{aligned} u[k] &= B \operatorname{sgn} \left(x_d[k] - x[k] + \frac{\lambda}{T} x_d[k] - \frac{\lambda}{T} x_d[k-1] - \frac{\lambda}{T} x[k] + \frac{\lambda}{T} x[k-1] \right) \\ &= B \operatorname{sgn} \left(\frac{T+\lambda}{T} x_d[k] - \frac{\lambda}{T} (x_d[k-1] - x[k-1]) - \frac{T+\lambda}{T} x[k] \right). \end{aligned}$$

Substitution of (D.28) gives

$$\begin{aligned} u[k] &= B \operatorname{sgn} \left(\frac{T+\lambda}{T} x_d[k] - \frac{\lambda}{T} (x_d[k-1] - x[k-1]) - \frac{T+\lambda}{T(1+T)} x[k-1] \right. \\ &\quad \left. - \frac{T+\lambda}{1+T} u[k] \right) \\ &= B \operatorname{sgn} \left(\frac{T+\lambda}{1+T} \left(\frac{1+T}{T} x_d[k] - \frac{\lambda(1+T)}{T(T+\lambda)} (x_d[k-1] - x[k-1]) \right. \right. \\ &\quad \left. \left. - \frac{1}{T} x[k-1] - u[k] \right) \right) \\ &= B \operatorname{sgn} \left(\frac{T+\lambda}{1+T} (u^*[k] - u[k]) \right) \end{aligned}$$

with

$$u^*[k] = \frac{1+T}{T} x_d[k] - \frac{\lambda(1+T)}{T(T+\lambda)} (x_d[k-1] - x[k-1]) - \frac{1}{T} x[k-1].$$

As before, this implies (using (5.2))

$$u[k] = \begin{cases} u^*[k] & \text{if } |u^*[k]| \leq B \\ B \operatorname{sgn}(u^*[k]) & \text{if } |u^*[k]| > B. \end{cases}$$

Again, if the control input $u^*[k]$ necessary to achieve $s(x, t) = 0$ exceeds B , the desired dynamics can't be achieved and the system can be considered to be "off the sliding surface".

Appendix E

Publication list

Peer-reviewed journal papers

- *Proxy-Based Sliding Mode Control of a Planar Pneumatic Manipulator*, M. Van Damme, B. Vanderborght, B. Verrelst, R. Van Ham, F. Daerden, D. Lefeber. *International Journal of Robotics Research*, Vol. 28, Issue 2, pp. 266-284 (2009).
- *Sliding Mode Control of a 2-DOF Planar Pneumatic Manipulator*, M. Van Damme, B. Vanderborght, R. Van Ham, B. Verrelst, F. Daerden, D. Lefeber. *Transactions of the ASME - Journal of Dynamic Systems, Measurement, and Control*. Vol. 131, Issue 2, 021013 (12 pages), (2009).
- *Sliding Mode Control of a “Soft” 2-DOF Planar Pneumatic Manipulator*, M. Van Damme, B. Vanderborght, R. Van Ham, B. Verrelst, F. Daerden, D. Lefeber. *International Applied Mechanics*. Vol. 44, Issue 10, pp. 1191-1199 (2008).
- *Design and Control of a Lower Limb Exoskeleton for Robot-assisted Gait Training*, P. Beyl, M. Van Damme, R. Van Ham, B. Vanderborght, D. Lefeber, *Applied Bionics and Biomechanics*, accepted for publication.
- *Successful preliminary walking experiments on a trans-tibial amputee fitted with a powered prosthesis*, R. Versluys, G. Lenaerts, M. Van Damme, I. Jonkers, A. Desomer, L. Peeraer, G. Van der Perre, D. Lefeber, *Prosthetics & Orthotics International*, submitted.
- *Prosthetic Feet: State-of-the-art Review and the Importance of Mimicking Human Ankle-Foot Biomechanics*, R. Versluys, P. Beyl, M. Van Damme, A. Desomer, R. Van Ham, D. Lefeber, *Disability and Rehabilitation: Assistive Technology*, Vol. 4, Issue 2, pp. 65-75 (2009).

- *A study on the bandwidth characteristics of pleated pneumatic artificial muscles*, R. Versluys, K. Deckers, M. Van Damme, R. Van Ham, G. Steenackers, P. Guillaume, D. Lefeber. *Applied Bionics and Biomechanics*. DOI: 10.1080/11762320902738647 (2009).
- *Objective locomotion parameters based inverted pendulum trajectory generator*, B. Vanderborght, B. Verrelst, R. Van Ham, M. Van Damme, D. Lefeber, *Robotics and Autonomous Systems*, Vol. 56, Issue 9, pp. 738-750 (2008).
- *Development of a compliance controller to reduce energy consumption for bipedal robots*, B. Vanderborght, B. Verrelst, R. Van Ham, M. Van Damme, P. Beyl, D. Lefeber, *Autonomous Robots*, Vol. 15, Issue 4, pp. 419-434 (2008).
- *Overview of the Lucy Project: Dynamic Stabilization of a Biped Powered by Pneumatic Artificial Muscles*, B. Vanderborght, R. Van Ham, B. Verrelst, M. Van Damme, D. Lefeber, *Advanced Robotics*, Vol.22, Issue 10, 1027-1051 (2008).
- *Treadmill Walking of the Pneumatic Biped Lucy: Walking at Different Speeds and Step-lengths*, B. Vanderborght, B. Verrelst, R. Van Ham, M. Van Damme, R. Versluys, D. Lefeber, *International Applied Mechanics*, Vol. 44, Issue 7, pp. 830-837 (2008).
- *MACCEPA, the mechanically adjustable compliance and controllable equilibrium position actuator: Design and implementation in a biped robot*, R. Van Ham, B. Vanderborght, M. Van Damme, B. Verrelst, D. Lefeber, *Robotics and Autonomous Systems*, Vol. 55, Issue 10, pp. 761-768 (2007)
- *MACCEPA, the Mechanically Adjustable Compliance and Controllable Equilibrium Position Actuator: A 3DOF Joint with 2 independent Compliances*, R. Van Ham, M. Van Damme, B. Verrelst, B. Vanderborght, D. Lefeber, *International Applied Mechanics*, Vol. 43, Issue 4, pp 467-474 (2007).
- *Exploiting Natural Dynamics to Reduce Energy Consumption by Controlling the Compliance of Soft Actuators*, B. Vanderborght, B. Verrelst, R. Van Ham, M. Van Damme, D. Lefeber, B. Meira y Duran, P. Beyl, *International Journal of Robotic Research*, Vol. 25, Issue 4, April 2006, pp. 343-358 (2006).
- *Second Generation Pleated Pneumatic Artificial Muscle and Its Robotic Applications*, B. Verrelst, R. Van Ham, B. Vanderborght, D. Lefeber, F. Daerden, M. Van Damme, *Advanced Robotics*, Vol. 20, Issue 7, pp. 783-805 (2006).
- *Motion Generation and Control for the Pneumatic Biped "Lucy"*, B. Verrelst, J. Vermeulen, B. Vanderborght, R. Van Ham, J. Naudet, D. Lefeber, F. Daerden, M. Van Damme, *International Journal of Humanoid Robotics (IJHR)*, Vol. 3, Issue 1, pp. 1-35 (2006).

Conference papers with full paper review

- *Modeling Hysteresis in Pleated Pneumatic Artificial Muscles*, VAN DAMME Michael, BEYL Pieter, VANDERBORGHT Bram, VAN HAM Ronald, VANDERNIEPEN Innes, VERSLUYS Rino, DAERDEN FRANK, LEFEBER DIRK, 2008 IEEE International Conference on Robotics, Automation & Mechatronics (RAM 2008), September 22-24, Chengdu, China, pp. 471 - 476.
Best Paper Award.
- *Design and control concepts of an exoskeleton for gait rehabilitation*, BEYL Pieter, VAN DAMME Michael, VAN HAM Ronald, LEFEBER DIRK, Proceedings of the 2nd Biennial IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, Scottsdale, USA, October 19-22, 2008, pp 103 - 108.
- *Design of a powered elbow orthosis for orthopaedic rehabilitation using compliant actuation*, VANDERNIEPEN Innes, VAN HAM Ronald, VAN DAMME Michael, LEFEBER DIRK, Proceedings of the 2nd Biennial IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, Scottsdale, USA, October 19-22, 2008, pp. 813 - 819.
- *From Conventional Prosthetic Feet to Bionic Feet: A Review Study*, VERSLUYS Rino, DESOMER Anja, LENAERTS Gerlinde, BEYL Pieter, VAN DAMME Michael, VANDERBORGHT Bram, VANDERNIEPEN Innes, Van der Perre Georges, LEFEBER DIRK, Proceedings of the 2nd Biennial IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, Scottsdale, USA, October 19-22, 2008, pp. 49-54.
- *A Pneumatically Powered Below-Knee Prosthesis: Design Specifications and First Experiments with an Amputee*, VERSLUYS Rino, DESOMER Anja, LENAERTS Gerlinde, VAN DAMME Michael, BEYL Pieter, Van der Perre Georges, Peeraer Louis, LEFEBER DIRK, Proceedings of the 2nd Biennial IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics, Scottsdale, USA, October 19-22, 2008, pp. 372 - 377.
- *An Exoskeleton for Gait Rehabilitation: Prototype Design and Control Principle*, BEYL Pieter, VAN DAMME Michael, VAN HAM Ronald, VERSLUYS Rino, VANDERBORGHT Bram, LEFEBER DIRK, Proceedings of the 2008 IEEE International Conference on Robotics and Automation (ICRA2008), May 19-23, Pasadena, California, 2008, pp. 2037-2042.
- *Proxy-Based Sliding Mode Control of a Manipulator Actuated by Pleated Pneumatic Artificial Muscles*, M. Van Damme, B. Vanderborght, R. Van Ham, B. Verrelst, F. Daerden, D. Lefeber. ICRA07 – 2007 IEEE International Conference on Robotics and Automation, 2007, pp. 4355-4360.

- *Novel compliant actuator for safe and ergonomic rehabilitation robots - Design of a powered elbow orthosis*, VANDERNIEPEN Innes, VAN HAM Ronald, NAUDET Joris, VAN DAMME Michael, VANDERBORGHT Bram, VERSLUYS Rino, LEFEBER DIRK, ICORR 2007 – IEEE 10th International Conference on Rehabilitation Robotics, 13-15 June 2007, pp. 790-797.
- *A strategy to combine active trajectory control with the exploitation of the natural dynamics to reduce energy consumption for bipedal robots*, VANDERBORGHT Bram, VAN DAMME Michael, VAN HAM Ronald, BEYL Pieter, LEFEBER DIRK, IEEE-RAS International Conference on Humanoids Robots - Humanoids 2007, November 29-December 1, 2007 Pittsburgh, USA.
- *Controlled Passive Walker Veronica Powered by Actuators with Independent Control of Equilibrium Position and Compliance*, VAN HAM Ronald, VANDERBORGHT Bram, VERRELST Bjorn, VAN DAMME Michael, LEFEBER DIRK, IEEE-RAS International Conference on Humanoids Robots - Humanoids 2006, December 4-6, 2006 Genova, Italy, pp. 234-239.
- *Locomotion Control Architecture for the Pneumatic Biped Lucy Consisting of a Trajectory Generator and Joint Trajectory Tracking Controller*, VANDERBORGHT Bram, VERRELST BJORN, VAN DAMME Michael, VAN HAM Ronald, BEYL Pieter, LEFEBER DIRK, IEEE-RAS International Conference on Humanoids Robots - Humanoids 2006, December 4-6, 2006 Genova, Italy, pp. 240-245.
- *MACCEPA: the Mechanically Adjustable Compliance and Controllable Equilibrium Position Actuator for ‘Controlled Passive Walking’*, VAN HAM Ronald, VANDERBORGHT Bram, VAN DAMME Michael, VERRELST Bjorn, LEFEBER Dirk. ICRA06 – 2006 IEEE International Conference on Robotics and Automation, 15-19 May 2006, Orlando Florida, USA, pp 2195-2200
- *Torque and Compliance Control of the Pneumatic Artificial Muscles in the Biped “Lucy”*, VANDERBORGHT Bram, VERRELST BJORN, VAN HAM Ronald, VAN DAMME Michael, BEYL Pieter, LEFEBER DIRK, ICRA06 – 2006 IEEE International Conference on Robotics and Automation, 15-19 May 2006, Orlando Florida, USA, pp 842-847
- *Pleated Pneumatic Artificial Muscles for Robotic Applications* (poster + video), VANDERBORGHT Bram, VAN DAMME Michael, VAN HAM Ronald, VERRELST Bjorn, LEFEBER Dirk. ICRA06 – 2006 IEEE International Conference on Robotics and Automation, 15-19 May 2006, Orlando Florida, USA, pp 4324-4326
- *Maccepa, the mechanically adjustable compliance and controllable equilibrium position actuator*, VAN HAM Ronald, VAN DAMME Michael, VANDERBORGHT Bram, VERRELST Bjorn, LEFEBER Dirk. Actuator 2006 - 10th

International Conference on New Actuators, 14-16 June 2006, Bremen, Germany, pp 1017-1020

- *A Pneumatic Biped: Experimental Walking Results and Compliance Adaptation Experiments*, VANDERBORGHT Bram, VERRELST Bjorn, VAN HAM Ronald, VAN DAMME Michael, LEFEBER Dirk. IEEE-RAS International Conference on Humanoids Robots - Humanoids 2005, December 5-7, 2005 Tsukuba International Congress Center (EPOCHAL TSUKUBA), Tsukuba, Japan, p 44-49
- *A Pneumatic Manipulator used in Direct Contact with an Operator*, VAN DAMME Michael, DAERDEN Frank, LEFEBER Dirk. Proceedings of the 2005 IEEE International Conference on Robotics and Automation, Barcelona, Spain, April 2005, p. 4505-4510.

Bibliography

1. Albu-Schäffer, A., Eiberger, O., Grebenstein, M., Haddadin, S., Ott, C., Wimböck, T., Wolf, S., and Hirzinger, G. Soft Robotics. *IEEE Robotics & Automation Magazine* 15(3):20–30, 2008.
2. Albu-Schäffer, A., Ott, C., and Hirzinger, G. A Unified Passivity-based Control Framework for Position, Torque and Impedance Control of Flexible Joint Robots. *The International Journal of Robotics Research* 26(1):23–39, 2007.
3. An, C. H., Atkeson, C. G., and Hollerbach, J. M. Estimation of inertial parameters of rigid body links of manipulators. In *Proceedings of the 24th IEEE Conference on Decision and Control*, pages 990–995. Fort Lauderdale, Florida, 1985.
4. An, C. H., Atkeson, C. G., and Hollerbach, J. M. *Model-Based Control of a Robot Manipulator*. MIT Press, Cambridge, Massachusetts, 1988.
5. Aschemann, H. and Hofer, E. P. Nonlinear Trajectory Control of a High-Speed Linear Axis Driven by Pneumatic Muscle Actuators. In *Proceedings of the 32nd Annual Conference on IEEE Industrial Electronics*, pages 3857–3862. 2006.
6. Avizienis, A., Laprie, J.-C., Randell, B., and Landwehr, C. Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE Transactions on Dependable and Secure Computing* 1(1):11–33, 2004.
7. Balasubramanian, K. and Rattan, K. Trajectory tracking control of a pneumatic muscle system using fuzzy logic. In *Proceedings of NAFIPS 2005 - Annual Meeting of the North American Fuzzy Information Processing Society*, pages 472–477. 2005.
8. Balasubramanian, K. and Rattan, K. S. Feedforward control of a non-linear pneumatic muscle system using fuzzy logic. In *Proceedings of The 12th IEEE International Conference on Fuzzy Systems*, volume 1, pages 272–277. 2003a.

9. Balasubramanian, K. and Rattan, K. S. Fuzzy logic control of a pneumatic muscle system using a linearizing control scheme. In *Proceedings of NAFIPS 2003 - 22nd International Conference of the North American Fuzzy Information Processing Society*, pages 432–436. 2003b.
10. Bélanger, P. R. Estimation of angular velocity and acceleration from shaft encoder measurements. In *Proceedings of the 1992 IEEE International Conference on Robotics and Automation*, pages 585–592. 1992.
11. Bemporad, A. and Mosca, E. Constraint fulfilment in feedback control via predictive reference management. In *Proceedings of the Third IEEE Conference on Control Applications*, volume 3, pages 1909–1914. 1994.
12. Beyl, P., Van Damme, M., Van Ham, R., Vanderborght, B., and Lefeber, D. Design and Control of a Lower Limb Exoskeleton for Robot-assisted Gait Training. *Applied Bionics and Biomechanics (accepted for publication)* 2009.
13. Bicchi, A., Peshkin, M. A., and Colgate, J. E. *Springer Handbook of Robotics*, chapter Safety for Physical Human-Robot Interaction, pages 1335–1348. Springer, Berlin, 2008.
14. Bicchi, A., Rizzini, S. L., and Tonietti, G. Compliant design for intrinsic safety: general issues and preliminary design. In *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 4, pages 1864–1869. Maui, Hawaii, 2001.
15. Bicchi, A. and Tonietti, G. Fast and Soft Arm Tactics: Dealing with the Safety-Performance Trade-Off in Robot Arms Design and Control. *IEEE Robotics and Automation Magazine* 11(2):22–33, 2004.
16. Bicchi, A., Tonietti, G., Bavaro, M., and Piccigallo, M. Variable Stiffness Actuators for Fast and Safe Motion Control. In B. Siciliano, O. Khatib, and F. Groen, editors, *Proceedings of the International Symposium on Robotics Research*, Springer Tracts in Advanced Robotics (STAR). Springer Verlag, 2003.
17. Brokate, M. Some Mathematical Properties of the Preisach Model for Hysteresis. *IEEE Transactions on Magnetics* 25(4):2922–2924, 1989.
18. Brokate, M. and Visintin, A. Properties of the Preisach model for hysteresis. *Journal fur die reine und angewandte Mathematik* 402:1–40, 1989.
19. Brown, R. H., Schneider, S. C., and Mulligan, M. G. Analysis of Algorithms for Velocity Estimation from Discrete Position versus Time Data. *IEEE Transactions on Industrial Electronics* 39(1):11–19, 1992.
20. Buerger, S. P. and Hogan, N. Complementary Stability and Loop Shaping for Improved Human-Robot Interaction. *IEEE Transactions on Robotics* 23(2):232–244, 2007.

21. Cai, D. and Dai, Y. A sliding mode controller for manipulator driven by artificial muscle actuator. In *Proceedings of the 2000 IEEE International Conference on Control Applications*, pages 668–673. 2000.
22. Cai, D. and Dai, Y. A Sliding Mode Controller for Manipulator Driven by Artificial Muscle Actuator. *Electronics and Communications in Japan* 86(11):57–64, 2003.
23. Cai, D. and Yamaura, H. A robust controller for manipulator driven by artificial muscle actuator. In *Proceedings of the 1996 IEEE International Conference on Control Applications*, pages 540–545. 1996.
24. Calafiore, G., Indri, M., and Bona, B. Robot Dynamic Calibration: Optimal Excitation Trajectories and Experimental Parameter Estimation. *Journal of Robotic Systems* 18(2):55–68, 2001.
25. Caldwell, D. G., Medrano-Cerda, G. A., and Goodwin, M. J. Braided Pneumatic Actuator Control of a Multi-Jointed Manipulator. In *International Conference on Systems, Man and Cybernetics*, pages 423–428. 1993.
26. Caldwell, D. G., Medrano-Cerda, G. A., and Goodwin, M. J. Control of pneumatic muscle actuators. *IEEE Control Systems Journal* 15(1):40–48, 1995.
27. Caldwell, D. G. and Tsagarakis, N. Biomimetic actuators in prosthetic and rehabilitation applications. *Technology and Health Care* 10:107–120, 2002.
28. Caldwell, D. G., Tsagarakis, N., Medrano-Cerda, G. A., Schofield, J., and Brown, S. A pneumatic muscle actuator driven manipulator for nuclear waste retrieval. *Control Engineering Practice* 9(1):23–36, 2001.
29. Carbonell, P., Jiang, Z. P., and Repperger, D. W. A Fuzzy Backstepping Controller for a Pneumatic Muscle Actuator System. In *Proceedings of the 2001 IEEE International Symposium on Intelligent Control*, pages 353–358. 2001a.
30. Carbonell, P., Jiang, Z. P., and Repperger, D. W. Nonlinear control of a pneumatic muscle actuator: backstepping vs.sliding-mode. In *Proceedings of the 2001 IEEE International Conference on Control Applications*, pages 167–172. 2001b.
31. Chang, M.-K., Yen, P.-L., and Yuan, T.-H. Angle Control of a one-Dimension Pneumatic Muscle Arm using Self-Organizing Fuzzy Control. In *Proceedings of the 2006 IEEE International Conference on Systems, Man, and Cybernetics*, pages 3834–3838. 2006.
32. Chang, X. and Lilly, J. H. Fuzzy Control for Pneumatic Muscle Tracking via Evolutionary Tuning. *Intelligent Automation and Soft Computing* 9(4):227–244, 2003.

33. Chettouh, M., Toumi, R., and Hamerlain, M. Derivative control in sliding mode of a robot driven by artificial muscles. In *Proceedings of the 1st IEEE International Conference on E-Learning in Industrial Electronics*. 2006.
34. Chettouh, M., Toumi, R., and Hamerlain, M. A 2-sliding control for pneumatic artificial muscles. In *Proceedings of the 5th International Multi-Conference on Systems, Signals and Devices*. 2008a.
35. Chettouh, M., Toumi, R., and Hamerlain, M. Chatter Reduction in an Artificial Muscles Robot Application. *International Journal of Robotics and Automation* 23(2):88–97, 2008b.
36. Chou, C. P. and Hannaford, B. Measurement and modelling of McKibben Pneumatic Artificial Muscles. *IEEE Transactions on Robotics and Automation* 12(1):90–102, 1996.
37. Clapa, D. J., Croft, E. A., and Hodgson, A. J. Equilibrium Point Control of a 2-DOF Manipulator. *Transactions of the ASME - G - Journal of Dynamic Systems Measurement Control* 128:134–141, 2006.
38. Colgate, J. E., Stanley, M. C., and Brown, J. M. Issues in the haptic display of tool use. In *Proceedings. 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 140–145. 1995.
39. Costa, N. and Caldwell, D. G. Control of a Biomimetic "Soft-actuated" 10DoF Lower Body Exoskeleton. In *Proceedings of the First IEEE/RAS-EMBS International Conference on Biomedical Robotics and Biomechatronics*, pages 495–501. 2006.
40. Daerden, F. *Conception and Realization of Pleated Pneumatic Artificial Muscles and their Use as Compliant Actuation Elements*. Ph.D. thesis, Vrije Universiteit Brussel, 1999.
41. Daerden, F. and Lefeber, D. The concept and design of pleated pneumatic artificial muscles. *International Journal of Fluid Power* 2(3):41–50, 2001.
42. Daerden, F. and Lefeber, D. Pneumatic artificial muscles: actuators for robotics and automation. *European Journal of Mechanical and Environmental Engineering* 47(1):10–21, 2002.
43. De Luca, A. Feedforward/feedback laws for the control of flexible robots. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 233 – 240. 2000.
44. De Luca, A., Albu-Schäffer, A., Haddadin, S., and Hirzinger, G. Collision Detection and Safe Reaction with the DLR-III Lightweight Manipulator Arm. In *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1623–1630. 2006.

45. De Luca, A. and Book, W. *Springer Handbook of Robotics*, chapter Robots with Flexible Elements, pages 287–319. Springer, Berlin, 2008.
46. De Luca, A. and Mattone, R. Actuator Failure Detection and Isolation using Generalized Momenta. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation*, pages 634–639. Taipei, Taiwan, 2003.
47. De Luca, A. and Mattone, R. Sensorless Robot Collision Detection and Hybrid Force-Motion Control. In *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pages 1011–1016. Barcelona, Spain, 2005.
48. De Santis, A., Siciliano, B., De Luca, A., and Bicchi, A. An atlas of physical human-robot interaction. *Mechanism and Machine Theory* 43(3):253–270, 2008.
49. Dhillon, B. S., Fashandi, A. R. M., and Liu, K. L. Robot systems reliability and safety: a review. *Journal of Quality in Maintenance Engineering* 8(3):170–212, 2002.
50. Diolaiti, N., Melchiorri, C., and Stramigioli, S. Contact Impedance Estimation for Robotic Systems. *IEEE Transactions on Robotics* 21(5):925–935, 2005.
51. Draženović, B. The Invariance Conditions in Variable Structure Systems. *Automatica* 5(3):287–295, 1969.
52. Eskiizmirli, S., Tondu, B., and Darlot, C. Motor control of a limb segment actuated by artificial muscles. In *Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, volume 1, pages 865–868. 2001.
53. EuroNCAP. *European New Car Assessment Programme - Assessment Protocol and Biomechanical Limits*. 2008.
54. Eykhoff, P. *System Identification: Parameter and State Estimation*. John Wiley & Sons, 1979.
55. Falcão Carneiro, J. and Gomes de Almeida, F. Modeling Pneumatic Servovalves using Neural Networks. In *Proceedings of the 2006 IEEE Conference on Computer Aided Control Systems Design*, pages 790–795. Munich, Germany, 2006.
56. Filippini, R., Sen, S., and Bicchi, A. Toward Soft Robots You Can Depend On. *IEEE Robotics & Automation Magazine* 15(3):31–41, 2008.
57. Fisette, P., Raucourt, B., and Samin, J.-C. Minimal Dynamic Characterization of Tree-Like Multibody Systems. *Nonlinear Dynamics* 9(1-2):165–184, 1996.
58. Formica, D., Zollo, L., and Guglielmelli, E. Torque-Dependent Compliance Control in the Joint Space for Robot-Mediated Motor Therapy. *Journal of Dynamic Systems, Measurement, and Control* 128:152–158, 2006.

59. Frankowicz, M. and Chrenowski, M. Application of Preisach model to adsorption-desorption hysteresis. *Physica B* 372:219–221, 2006.
60. Friedland, B. *Advanced Control System Design*. Prentice Hall, Englewood Cliffs, New Jersey, 1996.
61. Gautier, M. Dynamic identification of robots with power model. In *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, pages 1922–1927. Albuquerque, New Mexico, 1997.
62. Gautier, M. and Khalil, W. Direct Calculation of Minimum Set of Inertial Parameters of Serial Robots. *IEEE Transactions on Robotics and Automation* 6(3):368–373, 1990.
63. Gilardi, G. and Sharf, I. Literature survey of contact dynamics modelling. *Mechanism and Machine Theory* 37:1213–1239, 2002.
64. Goodman, L. E. and Warner, W. H. *Dynamics*. Dover Publications, Mineola, New York, 2001.
65. Gorbet, R. B. *Control of Hysteretic Systems with Preisach Representations*. Ph.D. thesis, University of Waterloo, 1997.
66. Haddadin, S., Albu-Schäffer, A., De Luca, A., and Hirzinger, G. Collision Detection & Reaction: A Contribution to Safe Physical Human-Robot Interaction. In *Proceedings of the 2008 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pages 3356–3363. Nice, France, 2008a.
67. Haddadin, S., Albu-Schäffer, A., and Hirzinger, G. Approaching Asimov’s 1st law: The role of the robot’s weight class. In *Robotics: Science and Systems Conference (RSS2007)*. Atlanta, USA, 2007a.
68. Haddadin, S., Albu-Schäffer, A., and Hirzinger, G. Safety Evaluation of Physical Human-Robot Interaction via Crash-Testing. In *Robotics: Science and Systems Conference (RSS2007)*. Atlanta, USA, 2007b.
69. Haddadin, S., Albu-Schäffer, A., and Hirzinger, G. The Role of the Robot Mass and Velocity in Physical Human-Robot Interaction - Part I: Non-constrained Blunt Impacts. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation*, pages 1331–1338. Pasadena, CA, 2008b.
70. Haddadin, S., Albu-Schäffer, A., and Hirzinger, G. The Role of the Robot Mass and Velocity in Physical Human-Robot Interaction - Part II: Constrained Blunt Impacts. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation*, pages 1339–1345. Pasadena, CA, 2008c.
71. Hägele, M., Nilsson, K., and Pires, J. N. *Springer Handbook of Robotics*, chapter Industrial Robotics, pages 963–986. Springer, Berlin, 2008.

72. Hamerlain, M. An anthropomorphic robot arm driven by artificial muscles using a variable structure control. In *Proceedings of the 1995 IEEE/RJS International Conference of Intelligent Robots and Systems*, volume 1, pages 550–555. 1995.
73. Heinrichs, B., Sepehri, N., and Thornton-Trump, A. B. Position Based Impedance Control of an Industrial Hydraulic Manipulator. *IEEE Control Systems Magazine* 17(1):46–52, 1997.
74. Heinzmann, J. and Zelinsky, A. Quantitative Safety Guarantees for Physical Human-Robot Interaction. *The International Journal of Robotics Research* 22(7-8):479–504, 2003.
75. Hesselroth, T., Sarkar, K., van der Smagt, P. P., and Schulten, K. Neural Network Control of a Pneumatic Robot Arm. *IEEE Transactions on Systems, Man, and Cybernetics* 24(1):28–38, 1994.
76. Hildebrandt, A., Sawodny, O., Neumann, R., and Hartmann, A. A Flatness Based Design for Tracking Control of Pneumatic Muscle Actuators. In *Proceedings of the 7th International Conference on Control, Automation, Robotics and Vision.*, pages 1156–1161. 2002.
77. Hildebrandt, A., Sawodny, O., Neumann, R., and Hartmann, A. Cascaded control concept of a robot with two degrees of freedom driven by four artificial pneumatic muscle actuators. In *Proceedings of the 2005 American Control Conference*, pages 680–685. Portland, OR, USA, 2005.
78. Hirzinger, G., Sporer, N., Albu-Shäffer, A., Hähle, M., Krenn, R., Pascucci, A., and Schedl, M. DLR’s torque-controlled light weight robot III—are we reaching the technological limits now? In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation*, pages 1710–1716. Washington, DC, 2002.
79. Hogan, N. Impedance Control: An approach to Manipulation - parts I - III. *Journal of Dynamic Systems, Measurement, and Control* 107:1–24, 1985.
80. Hollerbach, J., Khalil, W., and Gautier, M. *Springer Handbook of Robotics*, chapter Model Identification, pages 321–344. Springer, Berlin, 2008.
81. Hsu, P., Bodson, M., Sastry, S., and Paden, B. Adaptive Identification and Control for Manipulators without Using Joint Accelerations. In *Proceedings of the 1987 IEEE International Conference on Robotics and Automation*, pages 1210–1215. 1987.
82. Hu, H. and Ben Mrad, R. On the classical Preisach model for hysteresis in piezoceramic actuators. *Mechatronics* 13:85–94, 2003.
83. Hung, J. Y., Gao, W., and Hung, J. C. Variable Structure Control: A Survey. *IEEE Transactions on Industrial Electronics* 40(1):2–22, 1993.

84. Hunt, K. H. and Crossley, F. R. E. Coefficient of Restitution Interpreted as Damping in Vibroimpact. *ASME Journal of Applied Mechanics* pages 440–445, 1975.
85. Ikuta, K., Ishii, H., and Nokata, M. Safety Evaluation Method of Design and Control for Human-Care Robots. *The International Journal of Robotics Research* 22(5):281–297, 2003.
86. Inoue, K. Rubbertuators and Applications for Robots. In *Proceedings of the 4th International Symposium on Robotic Research*, pages 57–63. 1987.
87. Janabi-Sharifi, F., Hayward, V., and Chen, C.-S. J. Discrete-Time Adaptive Windowing for Velocity Estimation. *IEEE Transactions on Control Systems Technology* 8(6):1003–1009, 2000.
88. Kapasouris, P., Athans, M., and Stein, G. Design of feedback control systems for stable plants with saturating actuators. In *Proceedings of the 27th IEEE Conference on Decision and Control*, volume 1, pages 469–479. 1988.
89. Khalil, H. K. *Nonlinear Systems*. Prentice Hall, 3 edition, 2002.
90. Khosla, P. K. and Kanade, T. Parameter Identification of Robot Dynamics. In *Proceedings of the 24th IEEE Conference on Decision and Control*, pages 1754–1760. Fort Lauderdale, Florida, 1985.
91. Kikuuwe, R. and Fujimoto, H. Proxy-based sliding mode control for accurate and safe position control. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, pages 25–30. 2006.
92. Kikuuwe, R., Yamamoto, T., and Fujimoto, H. Velocity-Bounding Stiff Position Controller. *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems* pages 3050–3055, 2006.
93. Kimura, T., Hara, S., Fujita, T., and Kagawa, T. Feedback Linearization for Pneumatic Actuator Systems with Static Friction. *Control Eng. Practice* 5(10):1385–1394, 1997.
94. Kulić, D. and Croft, E. A. Real-time safety for human-robot interaction. *Robotics and Autonomous Systems* 54(1):1–12, 2006.
95. Laprie, J.-C. Dependable Computing and Fault Tolerance: Concepts and Terminology. In *Proceedings of the 15th IEEE International Symposium on Fault-Tolerant Computing*, pages 2–11. 1985.
96. Lawrence, D. A. Impedance control stability properties in common implementations. In *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, volume 2, pages 1185–1190. 1988.

97. Lee, S.-H. and Song, J.-B. Acceleration estimator for low-velocity and low-acceleration regions based on encoder position data. *IEEE/ASME Transactions on Mechatronics* 6(1):58–64, 2001.
98. Li, W. and Slotine, J.-J. Parameter Estimation Techniques for Robotic Applications. In *ASME Winter Annual Meeting*, pages 213–218. 1987.
99. Li, Y., Gu, F., Harris, G., Ball, A., Bennett, N., and Travis, K. The measurement of instantaneous angular speed. *Mechanical Systems and Signal Processing* 19:786–805, 2005.
100. Lilly, J. H. Adaptive Tracking for Pneumatic Muscle Actuators in Bicep and Tricep Configurations. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 11(3):333–339, 2003.
101. Lilly, J. H. and Quesada, P. M. A Two-Input Sliding-Mode Controller for a Planar Arm Actuated by Four Pneumatic Muscle Groups. *IEEE Transactions On Neural Systems and Rehabilitation Engineering* 12(3):349–359, 2004.
102. Lilly, J. H. and Yang, L. Sliding Mode Tracking for Pneumatic Muscle Actuators in Opposing Pair Configuration. *IEEE Transactions On Control Systems Technology* 13(4):550–558, 2005.
103. Lim, H.-o. and Tanie, K. Human Safety Mechanisms of Human-Friendly Robots: Passive Viscoelastic Trunk and Passively Movable Base. *The International Journal of Robotics Research* 19(4):307–335, 2000.
104. Liu, G., Goldenberg, A. A., and Zhang, Y. Precise slow motion control of a direct-drive robot arm with velocity estimation and friction compensation. *Mechatronics* 14(7):821–834, 2004.
105. Ljung, L. *System Identification - Theory for the User*. Prentice Hall, Upper Saddle River, NJ, 2 edition, 1999.
106. Mayergoyz, I. D. *Mathematical Models of Hysteresis*. Springer Verlag, New York, 1991.
107. Medrano-Cerda, G. A., Bowler, C. J., and Caldwell, D. G. Adaptive position control of antagonistic pneumatic muscle actuators. In *Proceedings of the International Conference on Intelligent Robots and Systems*, volume 1, pages 378–383. 1995.
108. Nicolò, F. and Katende, J. *Second IASTED International Symposium on Robotics and Automation*, chapter A Robust MRAC for Industrial Robots, pages 64–67. Acta Press, 1983.
109. Noritsugu, T. and Tanaka, T. Application of a Rubber Artificial Muscle Manipulator as a Rehabilitation Robot. *IEEE/ASME Transactions on Mechatronics* 2(4):259–267, 1997.

110. Nouri, A. S., Gauvert, C., Tondu, B., and Lopez, P. Generalized variable structure model reference adaptive control of one-link artificial muscle manipulator in two operating modes. In *Proceedings of the 1994 IEEE International Conference on Systems, Man, and Cybernetics*, volume 2, pages 1944–1950. 1994.
111. Oberer, S. and Schraft, R. D. Robot-Dummy Crash Tests for Robot Safety Assessment. In *2007 IEEE International Conference on Robotics and Automation*, pages 2934–2939. Rome, Italy, 2007.
112. Otani, K. and Kakizaki, T. Motion Planning and Modeling for Accurately Identifying Dynamic Parameters of an Industrial Robotic Manipulator. In *Proceedings of the 24th ISIR*, pages 743–748. Tokyo, Japan, 1993.
113. Park, K.-J. Fourier-based optimal excitation trajectories for the dynamic identification of robots. *Robotica* 24:625–633, 2006.
114. Perruquetti, W. and Barbot, J. P., editors. *Sliding Mode Control in Engineering*. Marcel Dekker, 2002.
115. Pervez, A. and Ryu, J. Safe physical human robot interaction - past, present and future. *Journal of Mechanical Science and Technology* 22:469–483, 2008.
116. Pham, M. T., Gautier, M., and Poignet, P. Identification of joint stiffness with bandpass filtering. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation*, pages 2867–2872. 2001.
117. Pratt, G. A. and Williamson, M. M. Series Elastic Actuators. In *IEEE International Workshop on Intelligent Robots and Systems (IROS 1995)*, pages 399–406. Pittsburgh, USA, 1995.
118. Preisach, F. Über die magnetische Nachwirkung. *Zeitschrift für Physik* 94:277–302, 1935.
119. Repperger, D. W., Johnson, K. R., and Philips, C. A. A VSC Position Tracking System Involving a large Scale Pneumatic Muscle Actuator. In *Proceedings of the 37th IEEE Conference on Decision & Control*, pages 4302–4307. 1998.
120. Reynolds, D. B., Repperger, D. W., Phillips, C. A., and Bandry, G. Modeling the Dynamic Characteristics of Pneumatic Muscle. *Annals of Biomedical Engineering* 31(3):310–317, 2003.
121. Richardson, R., Brown, M., Bhakta, B., and Levesley, M. C. Design and control of a three degree of freedom pneumatic physiotherapy robot. *Robotica* 21:589–604, 2003.
122. Ritter, H., Haschke, R., Koiva, R., Rothling, F., and Steil, J. A layered control architecture for imitation grasping with a 20-DOF pneumatic anthropomorphic hand. In *Internal report University of Bielefeld*. 2005.

123. Salisbury, K., Townsend, W., Eberman, B., and DiPietro, D. Preliminary design of a whole-arm manipulation system (WAMS). In *Proceedings of the 1988 IEEE International Conference on Robotics and Automation*, volume 1, pages 254–260. Philadelphia, PA, 1988.
124. Sardellitti, I., Park, J., Shin, D., and Khatib, O. Air muscle controller design in the distributed macro-mini (DM²) actuation approach. In *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1822–1827. 2007.
125. Sastry, S. *Nonlinear systems: Analysis, Stability and Control*. Springer Verlag, 1999.
126. Schindele, D. and Aschemann, H. Nonlinear Model Predictive Control of a High-Speed Linear Axis Driven by Pneumatic Muscles. In *Proceedings of the 2008 American Control Conference*, pages 3017–3022. 2008.
127. Schröder, J., Kawamura, K., Gockel, T., and Dillmann, R. Improved control of a humanoid arm driven by pneumatic actuators. In *Proceedings of Humanoids 2003*. 2003.
128. Schulte, H. F. The characteristics of the McKibben artificial muscle. In *The Application of External Power in Prosthetics and Orthotics*, Publication 874, pages 94–115. National Academy of Sciences - National Research Council, Washington DC, 1961.
129. Shin, D., Sardellitti, I., and Khatib, O. A Hybrid Actuation Approach for Human-Friendly Robot Design. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation*, pages 1747–1752. Pasadena, CA, 2008.
130. Sira-Ramírez, H., Lopez, P., and Tondu, B. On the robust stabilization and tracking for robotic manipulators with artificial muscles. *International Journal of Systems Science* 27(11):1067–1075, 1996.
131. Šitum, Ž. and Herceg, S. Design and Control of a Manipulator Arm Driven by Pneumatic Muscle Actuators. In *Proceedings of the 16th IEEE Mediterranean Conference on Control and Automation*, pages 926–932. 2008.
132. Slotine, J.-J. and Li, W. *Applied Nonlinear Control*. Prentice Hall, Upper Saddle River, New Jersey, 1991.
133. Sorli, M., Figliolini, G., and Pastorelli, S. Dynamic Model and Experimental Investigation of a Pneumatic Proportional Pressure Valve. *IEEE/ASME Transactions on Mechatronics* 9(1):78–86, 2004.
134. Spong, M. W. Modeling and control of elastic joint robots. *ASME Journal of Dynamic Systems, Measurement, and Control* 109:310–319, 1987.

135. Spong, M. W., Hutchinson, S., and Vidyasagar, M. *Robot Modeling and Control*. John Wiley & Sons, Inc., 2006.
136. Su, Y. X., Zheng, C. H., Mueller, P. C., and Duan, B. Y. A Simple Improved Velocity Estimation for Low-Speed Regions Based on Position Measurements Only. *IEEE Transactions on Control Systems Technology* 14(5):937–942, 2006.
137. Sugar, T. G., He, J., Koeneman, E. J., Koeneman, J. B., Herman, R., Huang, H., Schultz, R. S., Herring, D. E., Wanberg, J., Balasubramanian, S., Swenson, P., and Ward, J. A. Design and Control of RUPERT: A Device for Robotic Upper Extremity Repetitive Therapy. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 15(3):336–346, 2007.
138. Swevers, J., Ganseman, C., De Schutter, J., and Van Brussel, H. Experimental robot identification using optimised periodic trajectories. *Mechanical Systems and Signal Processing* 10(5):561–577, 1996.
139. Swevers, J., Ganseman, C., Tükel, D. B., De Schutter, J., and Van Brussel, H. Optimal Robot Excitation and Identification. *IEEE Transactions on Robotics and Automation* 13(5):730–740, 1997.
140. Swevers, J., Verdonck, W., and De Schutter, J. Dynamic Model Identification for Industrial Robots: Integrated Experiment Design and Parameter Estimation. *IEEE Control Systems Magazine* 27(5):58–71, 2007.
141. Thanh, T. D. C. and Ahn, K. K. Intelligent phase plane switching control of pneumatic artificial muscle manipulators with magneto-rheological brake. *Mechatronics* 16:85–95, 2006a.
142. Thanh, T. D. C. and Ahn, K. K. Nonlinear PID control to improve the control performance of 2 axes pneumatic artificial muscle manipulator using neural network. *Mechatronics* 16(9):577–587, 2006b.
143. Tilli, A. and Montanari, M. A low-noise estimator of angular speed and acceleration from shaft encoder measurements. In *9th Mediterranean Conference on Control and Automation (MED2001)*. 2001.
144. Tomei, P. A simple PD controller for robots with elastic joints. *IEEE Transactions on Automatic Control* 36(10):1208–1213, 1991.
145. Tondu, B., Boitier, V., and Lopez, P. Naturally compliant robot-arms actuated by McKibben artificial muscles. In *Proceedings of the 1994 IEEE International Conference on Systems, Man and Cybernetics*, volume 3, pages 2635–2640. 1994.
146. Tondu, B. and Lopez, P. Modeling and Control of McKibben Artificial Muscle Robot Actuators. *IEEE Control Systems Magazine* 20(2):15–38, 2000.

147. Tonietti, G. and Bicchi, A. Adaptive Simultaneous Position and Stiffness Control for a Soft Robot Arm. In *Proc. IEEE Int. Symp. Intelligent Robots and Systems*, pages 1992–1997. 2002.
148. Tsagarakis, N. G. and Caldwell, G. Development and Control of a "Soft-Actuated" Exoskeleton for Use in Physiotherapy and Training. *Autonomous Robots* 15:21–33, 2003.
149. Utkin, V. I. Variable Structure Systems with Sliding Modes. *IEEE Transactions On Automatic Control* 22:212–222, 1977.
150. Valency, T. and Zacksenhouse, M. Accuracy/Robustness Dilemma in Impedance Control. *Journal of Dynamic Systems, Measurement, and Control* 125:310–319, 2003.
151. Vallery, H., Veneman, J., Van Asseldonk, E., Ekkelenkamp, R., Buss, M., and van der Kooij, H. Compliant Actuation of Rehabilitation Robots. *IEEE Robotics & Automation Magazine* 15(3):60–69, 2008.
152. van der Smagt, P., Groen, F., and Schulten, K. Analysis and control of a rubber-tuator arm. *Biological Cybernetics* 75:433–440, 1996.
153. Van Ham, R., Sugar, T. G., Vanderborght, B., Hollander, K. W., and Lefeber, D. Review of Actuators with Passive Adjustable Compliance/Controllable Stiffness for Robotic Applications. *IEEE Robotics and Automation Magazine (accepted for publication)*. 2009.
154. Van Ham, R., Vanderborght, B., Van Damme, M., Verrelst, B., and Lefeber, D. MACCEPA, the Mechanically Adjustable Compliance and Controllable Equilibrium Position Actuator: Design and Implementation in a Biped Robot. *Robotics and Autonomous Systems* 55(10):761–768, 2007.
155. Vanderborght, B., Tsagarakis, N., Semini, C., Van Ham, R., and Caldwell, D. MACCEPA 2.0: Compliant Actuator used for Energy Efficient Hopping Robot CHOBINO. In *Proceedings of the 2009 IEEE International Conference on Robotics and Automation (accepted for publication)*. 2009.
156. Vanderborght, B., Van Ham, R., Verrelst, B., Van Damme, M., and Lefeber, D. Overview of the Lucy Project: Dynamic Stabilization of a Biped Powered by Pneumatic Artificial Muscles. *Advanced Robotics* 22(25):1027–1051, 2008a.
157. Vanderborght, B., Verrelst, B., Van Ham, R., and Lefeber, D. Controlling a Bipedal Walking Robot Actuated by Pleated Pneumatic Artificial Muscles. *Robotica* 24(4):401–410, 2006a.
158. Vanderborght, B., Verrelst, B., Van Ham, R., Van Damme, M., Beyl, P., and Lefeber, D. Development of a compliance controller to reduce energy consumption for bipedal robots. *Autonomous Robots* 24(4):419–434, 2008b.

159. Vanderborght, B., Verrelst, B., Van Ham, R., Van Damme, M., Lefeber, D., Meira Y Duran, B., and Beyl, P. Exploiting Natural Dynamics to Reduce Energy Consumption by Controlling the Compliance of Soft Actuators. *The International Journal of Robotics Research* 25(4):343–358, 2006b.
160. Verrelst, B., Van Ham, R., Vanderborght, B., Daerden, F., and Lefeber, D. The Pneumatic Biped "LUCY" Actuated with Pleated Pneumatic Artificial Muscles. *Autonomous Robots* 18:201–213, 2005.
161. Verrelst, B., Van Ham, R., Vanderborght, B., Lefeber, D., Daerden, F., and Van Damme, M. Second generation pleated pneumatic artificial muscle and its robotic applications. *Advanced Robotics* 20(7):783–805, 2006a.
162. Verrelst, B., Vermeulen, J., Vanderborght, B., Van Ham, R., Naudet, J., Lefeber, D., Daerden, F., and Van Damme, M. Motion Generation and Control for the Pneumatic Biped "Lucy". *International Journal of Humanoid Robotics* 3(1):1–35, 2006b.
163. Versace, J. A Review of the Severity Index. In *Proc. 15th Stapp Car Crash conference*. New York, 1971.
164. Villani, L. and De Schutter, J. *Springer Handbook of Robotics*, chapter Force Control, pages 161–185. Springer, Berlin, 2008.
165. Wassink, M. and Stramigioli, S. Towards a Novel Safety Norm for Domestic Robotics. In *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3354–3359. San Diego, CA, 2007.
166. Willinger, R., Kang, H.-S., and Diaw, B. Three-Dimensional Human Head Finite-Element Model Validation Against Two Experimental Impacts. *Annals of Biomedical Engineering* 27:403–410, 1999.
167. Wolf, S. and Hirzinger, G. A New Variable Stiffness Design: Matching Requirements of the Next Robot Generation. In *Proceedings of the 2008 IEEE International Conference on Robotics and Automation*, pages 1741–1746. Pasadena, CA, 2008.
168. Yamada, Y., Hirasawa, Y., Huang, S., Umetani, Y., and Suita, K. Human-Robot Contact in the Safeguarding Space. *IEEE/ASME Transactions on Mechatronics* 2(4):230–236, 1997.
169. Yamazaki, M. and Yasunobu, S. An Intelligent Control for State-Dependent Non-linear Actuator and Its Application to Pneumatic Servo System. In *Proceedings of the SICE Annual Conference 2007*, pages 2194–2199. 2007.
170. Yang, L. *Sliding Mode Control of Robotics Systems Actuated by Pneumatic Muscles*. Ph.D. thesis, University of Louisville, 2006.

171. Zhang, L., Xie, J., and Lu, D. Adaptive Robust Control of One-Link Joint Actuated by Pneumatic Artificial Muscles. In *Proceedings of the 1st IEEE International Conference on Bioinformatics and Biomedical Engineering*, pages 1185–1189. 2007.
172. Zilles, C. B. and Salisbury, J. K. A Constraint-based God-object Method for Haptic Display. In *Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems*, volume 3, pages 146–151. 1995.
173. Zinn, M., Khatib, O., Roth, B., and Salisbury, J. K. Playing It Safe. *IEEE Robotics & Automation Magazine* 11(2):12–21, 2004a.
174. Zinn, M., Khatib O., Roth, B., and Salisbury, J. K. A New Actuation Approach for Human Friendly Robot Design. In *Proceedings of International Symposium on Experimental Robotics*. Santa Angelo d’Ischia, Italy, 2002.
175. Zinn, M., Roth, B., Khatib O., and Salisbury, J. K. A New Actuation Approach for Human Friendly Robot Design. *The International Journal of Robotics Research* 23(4-5):379–398, 2004b.
176. Zollo, L., Siciliano, B., Laschi, C., Teti, G., and Dario, P. An experimental study on compliance control for a redundant personal robot arm. *Robotics and Autonomous Systems* 44:101–129, 2003.
177. Zsolt, S. *Hysteresis models from elementary operators and integral equations*. Ph.D. thesis, Budapest University of Technology and Economics, 2002.