



UPPSALA
UNIVERSITET

IT Licentiate theses
2018-003

Modeling and assessment of human balance and movement disorders using inertial sensors

FREDRIK OLSSON

UPPSALA UNIVERSITY
Department of Information Technology





UPPSALA
UNIVERSITET

Modeling and assessment of human balance and movement disorders
using inertial sensors

Fredrik Olsson
fredrik.olsson@it.uu.se

May 2018

*Division of Systems and Control
Department of Information Technology
Uppsala University
Box 337
SE-751 05 Uppsala
Sweden*

<http://www.it.uu.se/>

Dissertation for the degree of Licentiate of Philosophy in Electrical Engineering with
specialization in Automatic Control

© Fredrik Olsson 2018
ISSN 1404-5117

Printed by the Department of Information Technology, Uppsala University, Sweden

Abstract

Inertial sensors and magnetometers are abundant in today's society, where they can be found in many of our everyday electronic devices, such as smart phones or smart watches. Their primary function is to measure the movement and orientation of the device and provide this information for the apps that request it.

This licentiate thesis explores the use of these types of sensors in biomedical applications. Specifically, how these sensors can be used to analyze human movement and work as a tool for assessment of human balance and movement disorders. The methods presented in this thesis deal with mathematical modeling of the sensors, their relationship to the biomechanical models that are used to describe the dynamics of human movement and how we can combine these models to describe the mechanisms behind human balance and quantify the symptoms of movement disorders.

The main contributions come in the form of four papers. A practical calibration method for accelerometers is presented in Paper I, that deals with compensation of intrinsic sensor errors that are common for relatively cheap sensors that are used in e.g. smart phones. In Paper II we present an experimental evaluation and minor extension of methods that are used to determine the position of the joints in a biomechanical model, using inertial sensor data alone. Paper III deals with system identification of nonlinear controllers operating in closed loop, which is a method that can be used to model the neuromuscular control mechanisms behind human balance. In Paper IV we propose a novel method for quantification of hand tremor, a primary symptom of neurological disorders such as Parkinson's disease (PD) or Essential tremor (ET), where we make use of data collected from sensors in a smart phone. The thesis also contains an introduction to the sensors, biomechanical modeling, neuromuscular control and the various estimation and modeling techniques that are used throughout the thesis.

Acknowledgments

My time as a PhD student here at the IT Department at Uppsala University have been both very rewarding and fun. I have learned so much and I have met so many amazing people.

First, I would like to thank my supervisor, Dr. Kjartan Halvorsen for giving me this opportunity and for providing continuous guidance and support over the years. Thanks for sharing your knowledge with me and for continuing to be a great source of inspiration. Even though we have been around 9500km apart for most of the time, I feel that our weekly Skype meetings and your ability to give reliable feedback whenever I email you, is proof for your dedication!

I would like to thank Prof. Alexander Medvedev for introducing me to his work on individualized treatment of neurological disorders and for sharing his knowledge and ideas. Our collaborations have been a great experience, especially since I have had the opportunity to work so close to real clinical practice.

Thanks to Dr. Manon Kok, who helped me out a great deal when I was getting started. You made it so much for me easier to start working with inertial sensors and your work since then has continued to be a great source of knowledge and inspiration for me!

I am very grateful to all of the people whom I have collaborated and/or co-authored with. Thanks to Prof. Thomas Schön, Prof. Torbjörn Wigren, Dr. Dave Zachariah, Dr. Per Mattsson and Dr. Anna Cristina Åberg for your support and feedback.

Writing this thesis was made a lot easier thanks to Dr. Kjartan Halvorsen, Lic. Andreas Svensson, Anna Wigren and Dr. Dave Zachariah. I am very grateful that you took your time to read and provide honest feedback, it has been extremely valuable for me! Special thanks also to Lic. Andreas Svensson, for acting as my personal L^AT_EX-support.

Working at the IT Department is never boring thanks to my awesome former and current colleagues. It feels great to be a

part of SysCon. Thanks to everyone for contributing to creating such an amazing atmosphere, both during and after work. The same goes for colleagues from other divisions, you are also pretty cool! I have also had a lot of fun meeting people at the many social events that have been organized during my years as a PhD student; the yearly ski trips to Åre, the Friday pubs and afterworks, the late afternoon board game sessions, our running group, playing Ultimate frisbee, Polhacks, the TNDR social events and of course all of the PhD defenses and parties I have been invited to. Thanks to all of the organizers and participants, you have made living in Uppsala so much more enjoyable!

Thanks to my flatmate, second cousin and friend Mark. It's truly an amazing fact that we have been sharing the same student apartment since I started my studies in Uppsala in 2009, that's close to one third of our lives now! It's been great sharing this time in Uppsala with you, especially all the crazy Valborgs!

Last but not least I would like to thank my family; my mother Susanne, my father Mikael and my sisters Viktoria and Sofia, for believing in me and supporting me throughout my entire life. If it weren't for you I would not be where I am today, and for that you will always have my utmost and sincerest gratitude.

Uppsala, May 2018

Fredrik

Contents

List of papers	ix
1 Introduction	1
1.1 Inertial sensors	2
1.2 Magnetometers	4
1.3 Outline of introductory chapters	4
1.4 Summary and contributions of the papers	5
1.4.1 Paper I	5
1.4.2 Paper II	5
1.4.3 Paper III	6
1.4.4 Paper IV	6
2 Biomechanical modeling and control	7
2.1 Lagrangian formalism	8
2.2 Inverted single pendulum	8
2.3 Inverted double pendulum	9
2.4 Neuromuscular control	11
2.4.1 State-space representation and linearization	13
2.4.2 Modeling the neuromuscular control law	14
2.4.3 Modeling the biological sensors	19
3 Estimation techniques using inertial sensors	21
3.1 Sensor models	21
3.1.1 Orientation representations	22
3.1.2 Measurement models	24
3.2 Orientation estimation	26
3.2.1 Extended Kalman filter	27
3.2.2 Using EKF for orientation estimation	29
3.3 Sensor calibration	29
3.3.1 Maximum likelihood estimation	30
3.3.2 Accelerometer and magnetometer calibration	31
3.4 Calibration of the biomechanical model	31

3.4.1	Linear least squares	33
3.4.2	Gradient descent	34
3.4.3	Hessian-based methods	34
3.5	Identification of neuromuscular control	35
3.5.1	Experiment design	37
3.5.2	Model selection	38
3.5.3	Estimation	41
3.5.4	Validation	42
4	Concluding remarks	45
4.1	Summary of contributions	45
4.2	Future work	45
	Glossary and notation	47
	References	49
	Paper I – Accelerometer calibration using sensor fusion with a gyroscope	53
5.1	Introduction	55
5.2	Model and Problem formulation	57
5.3	Calibration algorithm	59
5.4	Experimental results	60
5.4.1	Real sensor data	60
5.4.2	Orientation estimation	61
5.4.3	Synthetic data	62
5.4.4	Cramér-Rao lower bound	64
5.5	Conclusions	64
	References	66
	Paper II – Experimental evaluation of joint position estimation using inertial sensors	69
6.1	Introduction	71
6.2	Modeling	72
6.2.1	Relating the inertial measurements to the biomechanical model	72
6.2.2	Measurement models	73
6.3	Estimation	74
6.3.1	Least-squares estimate	74
6.3.2	Estimation using iterative optimization methods	76
6.4	Experiment	77
6.5	Results	79
6.5.1	Metrics for evaluation of estimators	79

6.5.2	Method verification using simulated data with ideal excitation	80
6.5.3	Estimation using the experimental data	80
6.5.4	Estimation with simulated soft tissue artifacts	82
6.5.5	Analysis of the cost functions of the iterative method	83
6.6	Discussion	84
6.6.1	Motion is important	84
6.6.2	The main difference of the estimators lies in the cost functions	84
6.6.3	The least-squares estimator relies on accurate orientation estimates	87
6.6.4	The SA estimator is least sensitive to outliers	89
6.7	Conclusion	89
	References	91

Paper III – Identification of nonlinear feedback mechanisms operating in closed loop using inertial sensors **93**

7.1	Introduction	95
7.2	Modeling and methods	96
7.2.1	Identification of a controller operating in closed loop	96
7.2.2	Inertial sensors	97
7.2.3	Method for nonlinear system identification	98
7.2.4	Evaluation metrics	99
7.3	Simulated standing human balance	99
7.3.1	Results and discussion	105
7.4	Position servo	106
7.4.1	Results and discussion	107
7.5	Conclusion and future work	108
	References	111

Paper IV – Non-parametric time-domain tremor quantification with smart phone for therapy individualization **113**

8.1	Introduction	115
8.2	Data acquisition	118
8.3	Spectral analysis methods	123
8.4	Extracting the tremor signal in time domain	125
8.4.1	Acceleration measurement model	126
8.4.2	Step 1. Orientation estimation	127
8.4.3	Step 2. Position estimation	128
8.4.4	Step 3. Separation of tremor and voluntary movement	128
8.4.5	Step 4. Planar projection and rotation	129
8.5	Frequency estimation using extreme points of curvature	130

8.6	Markov chain model of tremor severity	133
8.6.1	Tremor quantification parameters	135
8.6.2	Results	136
8.7	Discussion	137
8.8	Conclusion	139
	References	141

List of Papers

This thesis is based on the following papers

- Paper I** F. Olsson, M. Kok, K. Halvorsen and T. B. Schön (2016). ‘Accelerometer calibration using sensor fusion with a gyroscope’. In: *Statistical Signal Processing Workshop (SSP), 2016 IEEE*. (Palma de Mallorca, Spain). IEEE, pp. 1–5
- Paper II** F. Olsson and K. Halvorsen (2017). ‘Experimental evaluation of joint position estimation using inertial sensors’. In: *Information Fusion (Fusion), 2017 20th International Conference on*. (Xi’an, China). IEEE, pp. 1–8
- Paper III** F. Olsson, K. Halvorsen, D. Zachariah and P. Mattsson (2018). *Identification of nonlinear feedback mechanisms operating in closed loop using inertial sensors*. To be presented at the 18th IFAC symposium on system identification (SYSID). Stockholm, Sweden
- Paper IV** F. Olsson and A. Medvedev (2018a). *Non-parametric Time-domain Tremor Quantification with Smart Phone for Therapy Individualization*. Submitted for publication

The following papers are of relevance to the thesis, but not included

- Paper A** K. Halvorsen and F. Olsson (2016). ‘Pose estimation of cyclic movement using inertial sensor data’. In: *Statistical Signal Processing Workshop (SSP), 2016 IEEE*. (Palma de Mallorca, Spain). IEEE, pp. 1–5
- Paper B** K. Halvorsen and F. Olsson (2017). ‘Robust tracking of periodic motion in the plane using inertial sensor data’. In: *IEEE Sensors 2017*. (Glasgow, Scotland)
- Paper C** A. Medvedev, F. Olsson and T. Wigren (2017). ‘Tremor quantification through data-driven nonlinear system modeling’. In: *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*. (Melbourne, Australia). IEEE, pp. 5943–5948

Paper D F. Olsson and A. Medvedev (2018b). *Tremor severity rating by Markov chains*. To be presented at the 18th IFAC symposium on system identification (SYSID). Stockholm, Sweden

Chapter 1

Introduction

The main focus of this thesis is the analysis of human movement using inertial sensors (accelerometers and gyroscopes) and magnetometers, see Figure 1.1. The use of these types of sensors is widespread in society today. They are used in our smart phones and smart watches, to perform basic functions, such as setting the correct screen orientation, and mobile apps that track activity (Kranz et al. 2013). Ever since the Nintendo Wii brought motion control to our home video game consoles, most modern game controllers have adopted the use of inertial sensors (LaViola Jr 2008). Inertial sensors are also used in motion capture, to create life-like animations that are used in movies and video games (Roetenberg et al. 2009). Our interest is in the biomedical area, and the applications discussed within this thesis is about using the measurements provided by inertial sensors to construct mathematical models that can be used to assess human balance and movement disorders.

The main motivation for working on the problem of balance assessment is the large increase in fall-related injuries due to loss of balance that we have seen over the past several years (WHO, Department of Ageing and Life Course (ALC) 2008). As the average age of the population increases, so do the problems caused by lowered mobility related to aging. Individually designed training and rehabilitation programs can help with preventing injuries, but must be designed and followed up by assessments of the motor control of the individual.

Today, assessment of human balance and movement disorders is done predominantly in movement laboratories or physiotherapy clinics. These assessments often rely on a physician's expert opinion after making multiple observations. High fidelity motion capture systems are sometimes used to construct mathematical models that facilitates more in-depth and objective assessment, but are often confined to the lab environment. Furthermore,



Figure 1.1: Two sensor platforms which contain inertial and magnetic sensors. The Xsens MTw Awinda wireless motion tracker (left) and a modern smart phone (right).

this type of assessment is expensive since it uses work hours of the physician and the limited lab resources. The same reasons lead to another problem, that assessment is performed infrequently and is therefore vulnerable to short-term fluctuations.

Inertial sensors has the potential to solve both of these issues due to their relatively low cost and small size. Allowing the assessment to be performed outside of the lab, preferably even in the patient's home, would ease the workload of the physicians and use less lab resources. The sensors may be worn throughout the daily life of the patient, which allows for more frequent assessment. We are specifically interested in using as few sensors as possible, to obtain an assessment technique that is unobtrusive and easy to maintain.

1.1 Inertial sensors

Throughout this thesis, the term *inertial sensors* will be used to denote both three-axis accelerometers and three-axis gyroscopes. Accelerometers measure the linear acceleration of the sensor, which includes the gravitational acceleration component, $g \in \mathbb{R}^3$, in addition to the acceleration caused

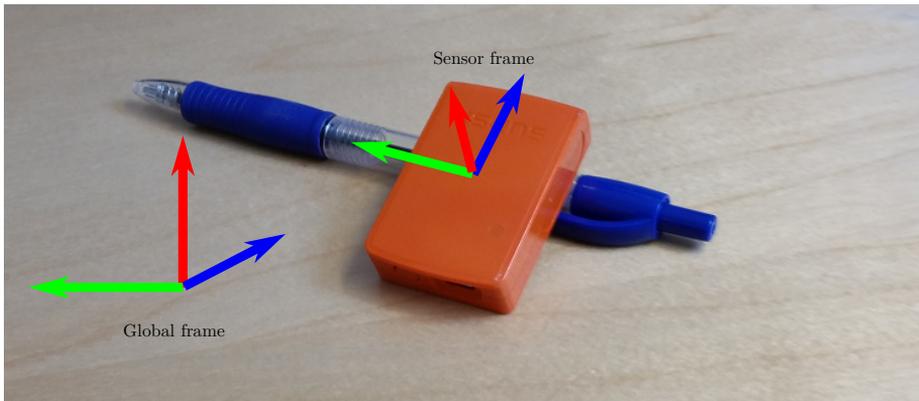


Figure 1.2: A global Earth-fixed reference frame and a sensor-fixed reference frame. The rotation from the global frame to the sensor frame defines the orientation of the sensor.

by the movement of the sensor itself. The gravitational acceleration varies depending on where on Earth you are. Specifically, how close the poles and how far above the center of the earth you are, as well as the density of the bedrock beneath your feet, will have an effect on the magnitude of g , which is measured in the Euclidean norm, $\|g\|_2$. For instance, in Smygehuk (southernmost point in Sweden) $\|g\|_2 = 9.815220\text{m s}^{-2}$ has been measured, which is slightly lower than at Treriksörset (northernmost point in Sweden), where $\|g\|_2 = 9.823944\text{m s}^{-2}$ has been measured (Lantmäteriet 2018). Typically, $\|g\|_2$ is approximated to be constant in the vicinity of a specific location, and this constant value can be used to calibrate the sensor.

Gyroscopes measure angular velocity, the rate of change of the sensor's orientation. The sensor's orientation is typically defined as the rotation between two Cartesian reference frames, the (global) navigation frame and the sensor frame, see Figure 1.2. In most applications we are interested in how the sensor moves in the navigation frame, and therefore we need to know its orientation. Integrating the gyroscope signal is a method for obtaining an estimate of the sensor's orientation relative to its initial orientation. However, as the measurements are typically biased and noisy, the integration will cause the estimates to drift and become more erroneous with time. This phenomenon will be referred to as *integration drift*.

Luckily, we can often use information from other sensors to obtain absolute observations of the orientation. This additional information can be used to correct for the integration drift of the gyroscope. Combining the information from multiple different sensors like this is referred to as *sensor fusion* (Gustafsson 2010). The g vector is typically aligned with the ver-

tical direction in the navigation frame, pointing radially outwards from the center of the Earth. Therefore, an accelerometer that is stationary in the navigation frame can be used to estimate the inclination of the sensor, which is the angle between the vertical direction in the navigation frame and the axis that has been defined as the vertical in the sensor frame. Note that the inclination provides no information about the heading, which is defined as the rotation angle around the vertical in the navigation frame. Therefore, inertial sensors alone can provide accurate estimates of the inclination but will yield erroneous estimates of the heading due to the integration drift.

Today, inertial sensors see abundant use and play an important role in navigation (Woodman 2007), motion capture (Hol 2011; Kok 2016) and healthcare and sports monitoring (Avci et al. 2010).

1.2 Magnetometers

Magnetometers are sensors that measure the local magnetic field, and are also frequently used in sensor platforms together with inertial sensors. Their most common use is similar to that of a regular compass; to provide an absolute observation of the heading. In the most commonly used reference frames in navigation, the horizontal axes will point in the longitudinal and latitudinal directions. In such a reference frame, the horizontal components of the Earth magnetic field will point to the magnetic north pole and a heading of 0° typically aligns with this direction. However, magnetometers used for this purpose are also susceptible to magnetic disturbances and will function poorly when they are surrounded by metallic or electrically conducting objects that have their own magnetic fields superimposed to the Earth magnetic field. This is important to be aware of when using magnetometer measurements to observe the heading. Having a prior model of the local Earth magnetic field can help to decide which measurements that may be affected by magnetic disturbances. In other applications, local fluctuations in the magnetic field can be exploited for benefit instead. It is possible to construct a map of the magnetic environment in buildings for example, which can be used in e.g. indoor navigation (Kok et al. 2013; Solin et al. 2015).

1.3 Outline of introductory chapters

The introductory chapters of this thesis consists of Chapter 1–4, and will serve to provide background and context to Paper I–IV, that make up the latter part of the thesis. Chapter 2 introduces biomechanical models that we use to describe the dynamics of a standing human. In the same chapter

we also introduce some of the most common mathematical models used to describe the mechanisms behind human balance. Chapter 3 introduces the various estimation problems that we face when working with inertial sensors. Measurement models for the inertial and magnetic sensors are introduced, as are the estimation problems related to orientation estimation, calibration of the sensors, calibration of the biomechanical model and identification of neuromuscular control. The methods that are used to solve these various problems are presented in the context of the problems themselves, rather than a standalone method summary, to emphasize the applications considered in this thesis. Finally, Chapter 4 concludes by summarizing the contributions of this thesis and potential future work is discussed.

1.4 Summary and contributions of the papers

The four papers contain the main contributions of this thesis. The focus of these papers concerns the practical use of inertial and magnetic sensors for analyzing human movement. Each paper contains examples where data from real sensors are used to illustrate the functionality of the proposed methods. Here follows a short summary of the papers.

1.4.1 Paper I

In this paper we propose a practical calibration method for accelerometers. The method only requires a gyroscope as additional equipment, which is commonly included in inertial sensor platforms. Lower quality accelerometers often have significant errors that have to be compensated for, such as misaligned sensor axes, varying gains and biases. Through sensor fusion of the accelerometer and the gyroscope we estimate the orientation of the sensor platform. By observing the accelerometer measurements for different orientations, we are able to identify a set of model parameters that can be applied to the raw accelerometer measurements to compensate for these types of errors.

1.4.2 Paper II

Here we consider a different type of calibration problem, which is to identify the position of a joint in a biomechanical model, with respect to two inertial sensors attached to the two segments adjacent to the joint. This can be thought of as calibrating the biomechanical model. The problem is formulated as an optimization problem, to find the joint positions that minimize a cost function, that is based on the kinematic constraints

of the biomechanical model. The contribution of this paper is the experimental evaluation and minor extension of three methods designed to solve this problem, where each method uses a differently formulated cost function. The evaluation is based on real data from sensors attached to two rigid cylindrical segments joined together by a spherical joint, where the true joint center with respect to the sensors could be measured.

1.4.3 Paper III

This paper concerns identification of the dynamics in unknown feedback controllers that operate in closed loop systems. The human balance system can be seen as one such closed loop system, where the controller consists of the central nervous system (CNS), which senses the state of the body and activates muscles to prevent us from falling. A recently proposed nonlinear system identification method (Mattsson et al. 2018) is applied to identify two different types of controllers; a simulated neuromuscular controller balancing a standing human and a real-world controller for a position servo that uses a DC-motor. Simulated and real inertial sensor data is used for identification. Thus, the contribution of this paper is not the identification method itself but rather it's application to closed loop identification and making use of inertial sensor data to do so.

1.4.4 Paper IV

In this paper we make use of inertial and magnetic sensor measurements from a smart phone to quantify the severity of hand tremor, which is a movement disorder and a primary symptom of neurological disorders such as Parkinson's disease (PD) and Essential tremor (ET). The contribution of this paper is the full description of a proposed method for tremor quantification. A tremor signal is obtained from smart phone measurements, which considers the involuntary movement caused by tremor as the deviation in position from an estimated voluntary movement. The amplitudes of this tremor signal is then modeled as a stochastic process in the form of a Markov chain. An invariant distribution is extracted from the Markov chain, which can then be interpreted as a probability distribution of the tremor severity. The method is then illustrated on real data from a PD patient with deep brain stimulation (DBS) implanted, which is a therapy used to treat the symptoms, such as tremor, of PD. Using the proposed method it is possible to distinguish between different DBS settings, which indicates that it could be used for therapy individualization.

Chapter 2

Biomechanical modeling and control

In this chapter we derive biomechanical models that can be used to analyze the balance mechanisms in standing humans. A very general model of the musculoskeletal system in humans is a kinematic chain, where each body segment is approximated as a rigid body link, which are connected by joints that constrain the relative movement of all links. The degrees of freedom in such a model is $6n - m$, where n is the number of individual links and m is the number of constraint equations due to the joints.

When studying standing human balance however, it is common to approximate the musculoskeletal system using a much simpler model, the *inverted pendulum*. Numerous studies have used this model to study standing human balance (Gage et al. 2004; Winter 1995). The motivation for this approximation is that there are two main strategies that humans use to maintain upright standing; the *ankle strategy* and the *hip strategy*. These strategies essentially mean that the central nervous system activates muscles, which generates torques around the ankle joint and the hip joint, respectively, hence the names. The inverted *single* pendulum, is a kinematic chain with one joint, the ankle joint, and all body segments above the feet are lumped together as one link. By adding one joint we get the inverted *double* pendulum, where the added joint represent the hip joint, the lower link represent the legs and the upper link represent the head, arms and trunk segments.

By only considering one or two links, we have reduced the degrees of freedom considerably, compared to a complete kinematic chain model with all body segments modeled by separate links. We can reduce the degrees of freedom further by constraining the movement of the inverted pendulum to a plane. This movement constraint is motivated by the fact that, given

a sufficient stance width, a standing human moves predominantly in the anterior-posterior directions. This results in only one rotation angle for each joint instead of three. Finally, if we fix the location of the feet, the ankle joint becomes stationary, and the position of the center of mass (CoM) of each link can be determined by only knowing the joint angles. Thus, the degrees of freedom of the model are reduced to one for the planar inverted single pendulum and two for the planar inverted double pendulum.

2.1 Lagrangian formalism

We will make use of the Lagrangian formalism to derive the equations of motion (EoM) that describe the rigid body dynamics of the biomechanical models. First we form the Lagrangian

$$L(q, \dot{q}) = E_k(q, \dot{q}) - E_p(q), \quad (2.1)$$

where $E_k(q, \dot{q})$ is the kinetic energy and $E_p(q)$ is the potential energy of the system. The Lagrangian depends on the generalized coordinates $q \in \mathbb{R}^n$ and their time derivatives (generalized velocities) $\dot{q} = \frac{dq}{dt}$, where n is the number of degrees of freedom of the system. For each generalized coordinate we get one EoM, which is computed as

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial q_i} = 0, \quad i = 1, \dots, n. \quad (2.2)$$

For the case of the inverted pendulum the generalized coordinates will be the angles ϕ between each link and the global vertical direction and the generalized velocities will be the angular velocities $\dot{\phi}$.

2.2 Inverted single pendulum

Starting with the simplest case, the inverted single pendulum, we first note that the Cartesian coordinates (x, y) , of the CoM are uniquely determined by the joint angle ϕ

$$x = c \sin \phi \quad (2.3)$$

$$y = c \cos \phi, \quad (2.4)$$

where c is the distance from the joint to the CoM. The Lagrangian is

$$L = \frac{mv^2}{2} + \frac{I \dot{\phi}^2}{2} - mgy, \quad (2.5)$$

where m is the mass of the link, v is the linear velocity of the CoM, I is the moment of inertia w.r.t. the rotation axis and g is the gravitational acceleration. Using (2.3)-(2.4) we can express the Lagrangian in terms of ϕ and $\dot{\phi}$

$$\begin{aligned} L &= \frac{m}{2}(\dot{x}^2 + \dot{y}^2) + \frac{I}{2}\dot{\phi}^2 - mgy \\ &= \frac{mc^2}{2}(\cos^2 \phi + \sin^2 \phi) + \frac{I}{2}\dot{\phi}^2 - mgc \cos \phi \\ &= \frac{mc^2 + I}{2}\dot{\phi}^2 - mgc \cos \phi. \end{aligned} \quad (2.6)$$

Using (2.2) we find that the EoM is

$$(mc^2 + I)\ddot{\phi} - mgc \sin \phi = 0, \quad (2.7)$$

which is also straightforward to derive using Newton's second law.

2.3 Inverted double pendulum

In an inverted double pendulum, see Figure 2.1, the CoM coordinates for each separate link is given by

$$x_1 = c_1 \sin \phi_1 \quad (2.8)$$

$$y_1 = c_1 \cos \phi_1 \quad (2.9)$$

$$x_2 = l_1 \sin \phi_1 + c_2 \sin \phi_2 \quad (2.10)$$

$$y_2 = l_1 \cos \phi_1 + c_2 \cos \phi_2, \quad (2.11)$$

where l is the length of the link and subscripts now indicate which link a certain quantity belong to. The Lagrangian then becomes

$$\begin{aligned} L &= \frac{m_1 v_1^2}{2} + \frac{m_2 v_2^2}{2} + \frac{I_1 \dot{\phi}_1^2}{2} + \frac{I_2 \dot{\phi}_2^2}{2} - m_1 g y_1 - m_2 g y_2 \\ &= \frac{m_1}{2}(\dot{x}_1^2 + \dot{y}_1^2) + \frac{m_2}{2}(\dot{x}_2^2 + \dot{y}_2^2) + \frac{I_1 \dot{\phi}_1^2}{2} + \frac{I_2 \dot{\phi}_2^2}{2} - m_1 g y_1 - m_2 g y_2 \\ &= \frac{m_1 c_1^2}{2} \dot{\phi}_1^2 (\cos^2 \phi_1 + \sin^2 \phi_1) + \frac{m_2}{2} (l_1^2 \dot{\phi}_1^2 \cos^2 \phi_1 + c_2^2 \dot{\phi}_2^2 \cos^2 \phi_2 \\ &\quad + 2l_1 c_2 \dot{\phi}_1 \dot{\phi}_2 \cos \phi_1 \cos \phi_2 + l_1^2 \dot{\phi}_1^2 \sin^2 \phi_1 + c_2^2 \dot{\phi}_2^2 \sin^2 \phi_2 \\ &\quad + 2l_1 c_2 \dot{\phi}_1 \dot{\phi}_2 \sin \phi_1 \sin \phi_2) + \frac{I_1 \dot{\phi}_1^2}{2} + \frac{I_2 \dot{\phi}_2^2}{2} - m_1 g y_1 - m_2 g y_2 \\ &= \frac{m_1}{2} c_1^2 \dot{\phi}_1^2 + \frac{m_2}{2} (l_1^2 \dot{\phi}_1^2 + c_2^2 \dot{\phi}_2^2 + 2l_1 c_2 \dot{\phi}_1 \dot{\phi}_2 \cos(\phi_1 - \phi_2)) \end{aligned}$$

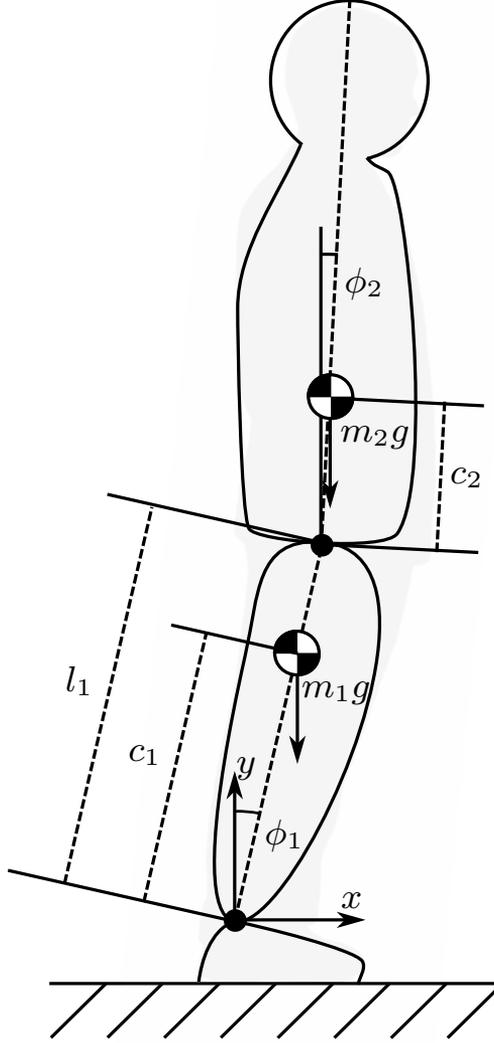


Figure 2.1: The inverted double pendulum model.

$$+ \frac{I_1 \dot{\phi}_1^2}{2} + \frac{I_2 \dot{\phi}_2^2}{2} - m_1 g c_1 \cos \phi_1 - m_2 g (l_1 \cos \phi_1 + c_2 \cos \phi_2), \quad (2.12)$$

and the two equations of motion are then given by

$$m_1 c_1^2 \ddot{\phi}_1 + m_2 \left(l_1^2 \ddot{\phi}_1 + l_1 c_2 \ddot{\phi}_2 \cos(\phi_1 - \phi_2) + l_1 c_2 \dot{\phi}_2^2 \sin(\phi_1 - \phi_2) \right) + I_1 \ddot{\phi}_1 - m_1 g c_1 \sin \phi_1 - m_2 g l_1 \sin \phi_1 = 0, \quad (2.13)$$

and

$$m_2 \left(c_2^2 \ddot{\phi}_2 + l_1 c_2 \ddot{\phi}_1 \cos(\phi_1 - \phi_2) - l_1 c_2 \dot{\phi}_1^2 \sin(\phi_1 - \phi_2) \right) + I_2 \ddot{\phi}_2 - m_2 g c_2 \sin \phi_2 = 0. \quad (2.14)$$

For systems with multiple degrees of freedom it is often convenient to express the EoM on the form

$$M(\phi) \ddot{\phi} + C(\phi, \dot{\phi}) \dot{\phi} + G(\phi) = 0, \quad (2.15)$$

where we see that inserting (2.13)-(2.14) yields

$$\underbrace{\begin{bmatrix} m_1 c_1^2 + m_2 l_1^2 + I_1 & m_2 l_1 c_2 \cos(\phi_1 - \phi_2) \\ m_2 l_1 c_2 \cos(\phi_1 - \phi_2) & m_2 c_2^2 + I_2 \end{bmatrix}}_{=M(\phi)} \begin{bmatrix} \ddot{\phi}_1 \\ \ddot{\phi}_2 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & m_2 l_1 c_2 \dot{\phi}_2 \sin(\phi_1 - \phi_2) \\ -m_2 l_1 c_2 \dot{\phi}_1 \sin(\phi_1 - \phi_2) & 0 \end{bmatrix}}_{=C(\phi, \dot{\phi})} \begin{bmatrix} \dot{\phi}_1 \\ \dot{\phi}_2 \end{bmatrix} + \underbrace{\begin{bmatrix} -m_1 g c_1 \sin \phi_1 - m_2 g l_1 \sin \phi_1 \\ -m_2 g c_2 \sin \phi_2 \end{bmatrix}}_{=G(\phi)} = 0. \quad (2.16)$$

2.4 Neuromuscular control

So far, the EoM that we have derived for inverted single and double pendula only contain the forces of gravity. Inverted pendula are inherently unstable; for any deviation from the vertical equilibrium point $\phi = 0$, the pendulum will start to fall. Humans are similar, without the aid of our muscles to counter the force of gravity we would meet the same fate. When the muscles that connect to the joint tendons contract, the result can be seen as a torque that is generated and acts on the adjacent body segments. This torque will be referred to as the *joint torque*. The human musculoskeletal system is inherently redundant in the sense that multiple muscles act across each joint. We view a joint torque as the net torque from all contributing muscles.

The muscle activity is controlled by the CNS, which sends activation signals out to the muscles. To choose appropriate activation signals, the CNS receives information, y , from biological sensors in our body and chooses an appropriate response depending on which task that should be performed. This is similar to how controllers work in a feedback loop. The task can

be seen as a certain desired configuration or *state* of the body segments, and is the equivalent of a reference signal, r . The reference signal is compared to the information about the current state and the control action that is generated by the CNS are the joint torques that should move the body segments closer to the desired state. Therefore, a control mechanism in the CNS that takes information from our biological sensors as input and outputs a joint torque by activating muscles will be referred to as a *neuromuscular controller*. As $\phi = \dot{\phi} = 0$ corresponds to upright standing in our biomechanical model, the reference signal in that case will correspond to this equilibrium point. Figure 2.2 shows a block diagram, which illustrates the closed neuromuscular control loop.

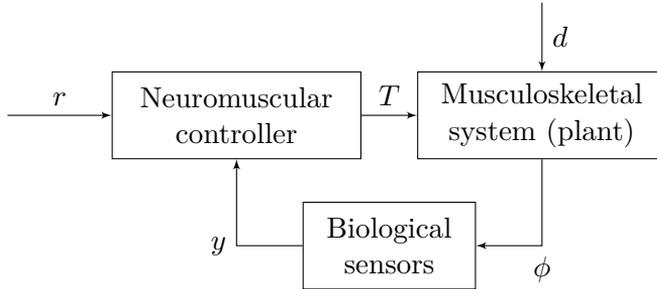


Figure 2.2: A closed neuromuscular control loop in humans.

We can expand the EoM to include the joint torques, T , as

$$M(\phi)\ddot{\phi} + C(\phi, \dot{\phi})\dot{\phi} + G(\phi) = DT, \quad (2.17)$$

where D is a matrix with elements $D_{ij} = \pm 1$, that describes how the joint torques affect the segments or $D_{ij} = 0$ if a joint torque does not affect some segments at all. For the inverted single pendulum we simply have $D = 1$ and in the inverted double pendulum we have

$$D = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}, \quad (2.18)$$

which follows from Newton's third law of equal and opposite forces.

Apart from the forces of gravity and the joint torques generated by the neuromuscular controller, the plant may also be affected by external disturbances, denoted by d in Figure 2.2. External disturbances play an important role in the identification of the neuromuscular controller, which is introduced in Chapter 3 and is the main topic of Paper III. The external disturbances can be designed as an exogenous input signal to the closed neuromuscular control loop, which allows information about the controller to be extracted via observations of the movement of the human.

2.4.1 State-space representation and linearization

We can formulate the extended EoM in (2.17) as a nonlinear state-space model

$$\begin{aligned} \begin{bmatrix} \dot{\phi} \\ \dot{\dot{\phi}} \end{bmatrix} &= \begin{bmatrix} 0 & I \\ 0 & -M(\phi)^{-1}C(\phi, \dot{\phi}) \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \end{bmatrix} \\ &+ \begin{bmatrix} 0 \\ -M(\phi)^{-1}G(\phi) \end{bmatrix} + \begin{bmatrix} 0 \\ M(\phi)^{-1}D \end{bmatrix} T \end{aligned}, \quad (2.19)$$

which is a system of nonlinear ordinary differential equations (ODE), where $x = \begin{bmatrix} \phi & \dot{\phi} \end{bmatrix}^\top$ is referred to as the state.

It is often convenient to approximate (2.19) as a linear state-space model, since control theory for linear systems is simpler and there exists many powerful mathematical methods for analysis or controller design (Kailath 1980; Rugh 1996). The EoM (2.17) can be expressed as the nonlinear function

$$f(\phi, \dot{\phi}, \ddot{\phi}, T) = M(\phi)\ddot{\phi} + C(\phi, \dot{\phi})\dot{\phi} + G(\phi) - DT = 0. \quad (2.20)$$

Let the variables be denoted by $z = \begin{bmatrix} \phi & \dot{\phi} & \ddot{\phi} & T \end{bmatrix}^\top$. We then select a linearization point $z_0 = \begin{bmatrix} \phi_0 & \dot{\phi}_0 & \ddot{\phi}_0 & T_0 \end{bmatrix}^\top$ and use a first-order Taylor expansion to approximate f as

$$\begin{aligned} \widehat{f}(\phi, \dot{\phi}, \ddot{\phi}, T) &= \left. \frac{\partial f}{\partial \phi} \right|_{z=z_0} (\phi - \phi_0) + \left. \frac{\partial f}{\partial \dot{\phi}} \right|_{z=z_0} (\dot{\phi} - \dot{\phi}_0) \\ &+ \left. \frac{\partial f}{\partial \ddot{\phi}} \right|_{z=z_0} (\ddot{\phi} - \ddot{\phi}_0) + \left. \frac{\partial f}{\partial T} \right|_{z=z_0} (T - T_0) \end{aligned} \quad (2.21)$$

$$= f_{\phi_0} \Delta_\phi + f_{\dot{\phi}_0} \Delta_{\dot{\phi}} + f_{\ddot{\phi}_0} \Delta_{\ddot{\phi}} + f_T \Delta_T, \quad (2.22)$$

where

$$f_{q_0} = \left. \frac{\partial f}{\partial q} \right|_{z=z_0}, \quad \Delta_q = q - q_0, \quad q = \{\phi, \dot{\phi}, \ddot{\phi}, T\}. \quad (2.23)$$

Note that the Taylor expansion (2.21) does not require the function value at the equilibrium point, since $f(z_0) = 0, \forall z_0$, per definition. Then we have obtained a linear state-space model for $\begin{bmatrix} \Delta_\phi & \Delta_{\dot{\phi}} \end{bmatrix}^\top$ according to

$$\begin{bmatrix} \Delta_{\dot{\phi}} \\ \Delta_{\ddot{\phi}} \end{bmatrix} = \begin{bmatrix} 0 & I \\ -f_{\ddot{\phi}_0}^{-1} f_{\phi_0} & -f_{\ddot{\phi}_0}^{-1} f_{\dot{\phi}_0} \end{bmatrix} \begin{bmatrix} \Delta_\phi \\ \Delta_{\dot{\phi}} \end{bmatrix} + \begin{bmatrix} 0 \\ -f_{\ddot{\phi}_0}^{-1} f_T \end{bmatrix} \Delta_T. \quad (2.24)$$

We now choose to linearize around the equilibrium point $z_0 = 0$, to obtain a state-space model on the form

$$\dot{x} = Ax + BT, \quad (2.25)$$

where

$$A = \begin{bmatrix} 0 & I \\ -f_{\ddot{\phi}_0}^{-1} f_{\phi_0} & -f_{\ddot{\phi}_0}^{-1} f_{\dot{\phi}_0} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ -f_{\ddot{\phi}_0}^{-1} f_T \end{bmatrix} \quad (2.26)$$

$$f_{\ddot{\phi}_0} = M(\phi_0) \quad (2.27)$$

$$f_{\dot{\phi}_0} = \left. \frac{\partial}{\partial \dot{\phi}} \left(C(\phi_0, \dot{\phi}) \dot{\phi} \right) \right|_{\dot{\phi}=\dot{\phi}_0} \quad (2.28)$$

$$f_{\phi_0} = \left. \frac{\partial}{\partial \phi} \left(M(\phi) \ddot{\phi}_0 + C(\phi, \dot{\phi}_0) \dot{\phi}_0 + G(\phi) \right) \right|_{\phi=\phi_0} \quad (2.29)$$

$$f_T = -D. \quad (2.30)$$

The linearized state-space representation of the EoM is a good approximation for sufficiently small deviations from the equilibrium point.

2.4.2 Modeling the neuromuscular control law

The neuromuscular controller is what we are primarily interested in modeling. A good model should not only be able to predict the response of the true controller, but also be easy to interpret so that it can provide meaningful insight about individual performance. Unlike the biomechanical model of the musculoskeletal system, there are no well defined laws of physics that can completely describe the dynamics of the neuromuscular controller. Even if we are able to describe the function of groups of several neurons we are still far from understanding the extremely complex and interconnected network, consisting of close to 100 billion neurons, which is the human brain (Herculano-Houzel 2009). However, many researchers have studied the neuromuscular control in human balance and have proposed different types of models as well as methods to identify these from empirical data. These are dynamical models that aim to describe the relationship between the inputs and outputs of the neuromuscular controller, which is significantly more simple than trying to explain the function of the brain itself. In this section, we will give a very brief introduction to some of the most common models that have been proposed.

In the most general sense, the neuromuscular control law can be formulated as

$$T(t) = h(y(\tau), r(\tau), \tau, \theta(\tau)), \quad -\infty < \tau \leq t, \quad (2.31)$$

where we state that the joint torque T is a function of the sensory information y and the reference signal r . The function $h(\cdot)$ can be any nonlinear function, and may be parametrized by possibly a limitless number of parameters, denoted by θ , which may also be time-varying. The control law is non-static, and could in general depend on the infinite history of its arguments up to time t , denoted by $-\infty < \tau \leq t$. A general model like this is not specific enough to be of practical value, and the measurements available to us also place restrictions on how accurately we can actually describe the causal relationship in (2.31). Therefore, we assume certain model structures for the function $h(\cdot)$ and the parameters $\theta(t)$. A common assumption is that the neuromuscular controller is linear and time-invariant (LTI) and that the reference signal is constant, corresponding to the desired equilibrium point (Park et al. 2004; Peterka 2002). These assumptions reduces (2.31) to

$$T(t) = \int_0^\infty h(t, \theta) y(t - \tau) d\tau, \quad (2.32)$$

where $h(t, \theta)$ is a function of time, also known as the *impulse response* of the controller, which depends on a finite set of constant parameters θ . An equivalent formulation of (2.32) in the frequency domain is

$$T(s) = H(s, \theta) y(s), \quad (2.33)$$

where $H(s, \theta)$ is the *transfer function* of the controller for the Laplace variable s , and $T(s)$ and $y(s)$ are the Laplace transforms of the time-domain signals $T(t)$ and $y(t)$, respectively. Below we describe controllers that assume the LTI form in (2.32).

Proportional derivative (PD) controller

One of the simplest controllers that one can think of is a proportional derivative (PD) controller. The output of the controller is then chosen as a linear combination of the observed joint angle and its first time derivative

$$T(t) = \begin{bmatrix} K_p & D_p \end{bmatrix} \begin{bmatrix} \phi(t) \\ \dot{\phi}(t) \end{bmatrix} = T_p(t), \quad (2.34)$$

where the model parameters correspond to $\theta = \{K_p, D_p\}$. The type of feedback that $T_p(t)$ represents is often referred to as *passive* or *intrinsic feedback* as it can be thought of as describing the passive dynamics of the muscles, tendons and soft tissue, which gives the standing human properties similar to that of a damped spring (Winter et al. 1998). The parameters K_p and D_p correspond to the stiffness and damping coefficients. While a PD controller certainly can be used in practice to stabilize inverted single and double pendulua it is not a very realistic model of neuromuscular control, as it does not take muscle activation and neural time delay into account.

Time delayed PD controller

A natural extension of the PD controller in (2.34) is to add feedback from time delayed states, which takes into account the time it takes for neuronal signals to travel from the biological sensors to the CNS and back to the muscles (Peterka 2002). This extended model can be formulized as

$$T(t) = \begin{bmatrix} K_p & D_p & K_a & D_a \end{bmatrix} \begin{bmatrix} \phi(t) \\ \dot{\phi}(t) \\ \phi(t - \tau) \\ \dot{\phi}(t - \tau) \end{bmatrix} = T_p(t) + T_a(t), \quad (2.35)$$

where the model parameters now correspond to $\theta = \{K_p, D_p, K_a, D_a, \tau\}$ and τ is the neural time-delay. Note that in a biomechanical model with multiple joints, different delays will be used for every joint since they involve different muscle groups and neural pathways. The term $T_a(t)$, which has been added to (2.34) to create (2.35), is often referred to as *active* or *reflexive feedback* as it takes into account the delays in neuronal signals that activates specific muscle groups.

In some cases the muscle activation dynamics are also described, which can be modeled in the frequency domain as

$$T_a(s) = (K_a\phi(s) + D_a s\phi(s))e^{-\tau s}H_{act}(s), \quad (2.36)$$

where $H_{act}(s)$ is the transfer function that models the muscle activation dynamics. In Boonstra et al. (2013) and Engelhart et al. (2016), $H_{act}(s)$ is modeled as a second-order system

$$H_{act}(s) = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2}, \quad (2.37)$$

where ω_0 is the natural frequency and ζ is the damping of the muscle activation dynamics, both of which are then included in the set of model parameters θ . This type of model has been used frequently to simulate a more realistic neuromuscular controller. We also utilize a similar simulation model in Paper III.

Linear quadratic (LQ) controller

Another way to view neuromuscular control is that the CNS in some way generates controller outputs which minimize an internal cost function, which is known as *optimal control* see Kuo (1995). A special case is the linear quadratic (LQ) controller, where the cost function is quadratic, which is the

optimal controller for LTI plant models such as (2.25). The LQ controller finds the control law such that the cost function

$$J = \int \left(\begin{bmatrix} \phi^\top(t) & \dot{\phi}^\top(t) \end{bmatrix} Q \begin{bmatrix} \phi(t) \\ \dot{\phi}(t) \end{bmatrix} + T^\top(t) R T(t) \right) dt, \quad (2.38)$$

is minimized, where Q and R are positive semi-definite matrices that determine how much weight the states and the controller output are given, respectively. The proportions of Q relative to R will determine the controller properties and therefore, these are the model parameters for the LQ controller i.e. $\theta = \{Q, R\}$. For LTI systems (2.38) has a closed form solution, which yields a control law on the form

$$T(t) = H \begin{bmatrix} \phi(t) \\ \dot{\phi}(t) \end{bmatrix} \quad (2.39)$$

$$H = -R^{-1} B^\top P, \quad (2.40)$$

where P is found by solving the continuous time algebraic Riccati equation

$$A^\top P + PA - PBR^{-1}B^\top P + Q = 0, \quad (2.41)$$

and A and B come from the linearized plant model (2.25).

Frequency response function (FRF)

It is also possible to characterize the neuromuscular controller in the frequency domain as

$$T(i\omega) = H(i\omega)\phi(i\omega), \quad (2.42)$$

for the angular frequency ω . The frequency response function (FRF) of the controller $H(i\omega)$ is a complex-valued function that can be described on polar form by its magnitude and phase

$$H(i\omega) = |H(i\omega)|e^{i\text{Arg}H(i\omega)}. \quad (2.43)$$

The FRF is often considered a *non-parametric* model since it cannot be described by a finite set of parameters θ , unlike the the previous (parametric) controller models we have studied. A rational transfer function model described by a finite set of parameters may be used to approximate the FRF, in which case the parameters are estimated to fit the empirical FRF. To construct an FRF model for the neuromuscular controller, one makes use of the LTI assumption and a known external disturbance signal d , which has been designed specifically to excite the system at certain frequencies. In this

case, the LTI assumption implies that if d has a periodic component for the angular frequency ω_0 , the output of the neuromuscular controller will also have periodic components at ω_0 , but with a different amplitude and phase. Designing the external disturbance d is therefore important, as it decides at which frequencies it is possible to determine the FRF. In theory a *white noise* signal can be used as it contains all frequencies, but in practice the frequency band will be limited by the experimental setup.

The FRF models can be constructed by *opening* the closed control loop seen in Figure 2.2,

$$H(i\omega) = -H_{dT}(i\omega)H_{d\phi}^{-1}(i\omega), \quad (2.44)$$

where $H_{dT}(i\omega)$ and $H_{d\phi}(i\omega)$ are the cross-spectral density (Stoica, Moses et al. 2005) matrices for the external disturbance d and the controller output T and the plant output ϕ , respectively. This is known as the joint input-output method (Kooij et al. 2005; Ljung 1999).

Nonlinear controllers

Thus far we have only considered models of the neuromuscular controller that rely on the LTI assumption. In reality however, the neuromuscular controller contains nonlinearities and expresses a time-varying behaviour. Therefore, we would like to find more flexible model structures that can describe these properties. However, a more flexible model structure generally means that the number of model parameters increases, which makes it harder to find good parameter values. Flexible nonlinear models can also be harder to interpret compared to e.g. the simple PD controller, where the parameters can be interpreted as the stiffness and damping coefficients in an oscillator. In Paper III we consider a flexible model structure for the controller, which combines a nominal linear model with an overparametrized error model to describe the nonlinearities according to

$$T(t) = \Theta\varphi(t) + Z\gamma(t). \quad (2.45)$$

The linear model is described by $\Theta\varphi(t)$, where $\varphi(t) \in \mathbb{R}^{n_\varphi \times 1}$ is a vector containing the controller inputs and $\Theta \in \mathbb{R}^{n_T \times n_\varphi}$ is a matrix that contains the linear parameters. Similarly the nonlinear model is described by $Z\gamma(t)$, but here $\gamma(t) \in \mathbb{R}^{n_\gamma \times 1}$ is a vector containing a set of nonlinear basis functions applied to the controller inputs and $Z \in \mathbb{R}^{n_T \times n_\gamma}$ is a matrix containing the nonlinear parameters. The nonlinear parameters are the coefficients which are multiplied with nonlinear basis functions of the information given as input to the controller. The number of controller outputs, n_T , is determined by the biomechanical model, while the dimensions of the parameter matrices,

n_ϕ and n_γ , are determined by the model structure, the assumptions about $\varphi(t)$ and $\gamma(t)$. To try to minimize the model complexity, the model can be constructed using a sparse set of nonlinear parameters, such that the nominal linear model describes as much of the dynamics as possible and a minimum number of nonlinear basis functions are used (Mattsson et al. 2018).

2.4.3 Modeling the biological sensors

The biological sensors that contribute with information to the CNS in human balance are primarily the visual, vestibular and proprioceptive sensory systems. We have used $y(t)$ to denote the information that is received by the CNS, and for the previously discussed controller models, we have only assumed that $y(t)$ consists of the plant outputs $\phi(t)$, $\dot{\phi}(t)$ and possibly time delayed versions of these signals. A more realistic model should include a noise term, since the information that the CNS receives about the current state of the plant is not accurate. The inaccuracies in the sensory information can be seen as a natural cause of the involuntary body sway that can be observed in quiet (meaning unperturbed) upright standing.

A model that has been widely used is the sensory weighting model (Peterka 2002; Peterka and Loughlin 2004)

$$y(t) = \begin{bmatrix} w_{\text{vis}} & w_{\text{ves}} & w_{\text{pro}} \end{bmatrix} \begin{bmatrix} y_{\text{vis}}(t) \\ y_{\text{ves}}(t) \\ y_{\text{pro}}(t) \end{bmatrix}, \quad (2.46)$$

where w_{vis} , w_{ves} and w_{pro} are weights and $y_{\text{vis}}(t)$, $y_{\text{ves}}(t)$ and $y_{\text{pro}}(t)$ represent the information from the visual, vestibular and proprioceptive sensors, respectively. Evidence from previous studies (Mahboobin et al. 2005; Peterka and Loughlin 2004) suggests that humans use an adaptive *sensory reweighting* scheme to modulate the information used by the neuromuscular controller. This means that the weights adapt to physiological or environmental conditions. For example, closing your eyes will remove the visual information, so other sensory systems have to compensate.

Chapter 3

Estimation techniques using inertial sensors

Regardless if we want to learn something about an unknown quantity or describe physical phenomena, we compute estimates based on what we observe. In this thesis, the observations are measurements from inertial and magnetic sensors. Here, we provide a suitable background to the various estimation problems that arise in the papers.

3.1 Sensor models

The inertial and magnetic sensors that we make use of are microelectromechanical systems (MEMS), embedded in microchips that are soldered to a circuit board, forming a *sensor platform*. Each sensor has three individual, orthogonal sensor axes, so all measurements are in 3D. The measurements of linear acceleration, angular velocity and the local magnetic field are all with respect to a reference frame, fixed in the sensor platform. We refer to this reference frame as the *sensor frame* (S), and it is defined to be fixed with its axes and origin aligned with the accelerometer axes and origin.

In many applications we are interested in relating the measured quantities to a global Earth-fixed reference frame, often referred to as the *navigation* or *global frame* (G). On Earth it is logical for the vertical axis in the global frame to be chosen to be parallel to the gravitational acceleration vector, which points toward the center of the Earth. The horizontal axes are often chosen differently depending on the situation. In navigation situations (e.g. aviation), the two horizontal axes typically point to the north, or latitudinal direction, and the east, or longitudinal direction. However, when analyzing human movement it often makes more sense to define the horizontal axes to

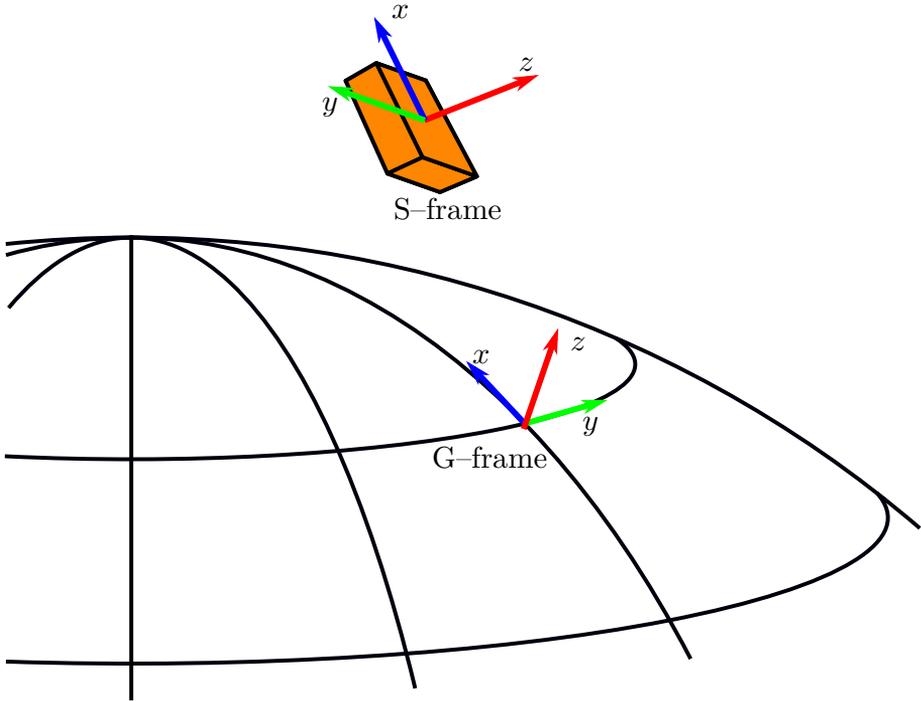


Figure 3.1: An illustration of the difference between the sensor frame (S), which has its axes fixed with respect to the sensor platform and the global frame (G), which has its axes fixed with respect to Earth.

suit the experimental setup. For example, when studying human balance, the human subject is often facing the same direction during a movement task. Then it is reasonable to have one axis pointing forward in the anterior/posterior direction and one axis pointing sideways in the medial/lateral direction. Figure 3.1 illustrates the difference between the S-frame and the G-frame.

3.1.1 Orientation representations

To transform the measured quantities from one reference frame to another is simply a change of orthonormal basis vectors in \mathbb{R}^3 . We make use of rotation matrices, which are part of the special orthogonal group $R \in \text{SO}(3)$. For example if v^G is a vector in the global frame and v^S is the same vector, but expressed in the sensor frame, then we have that

$$v^S = R^{S^G} v^G, \quad (3.1)$$

where R^{S^G} is the rotation matrix that when multiplied with any vector in the G-frame, maps that vector into the S-frame. The inverse rotation is the

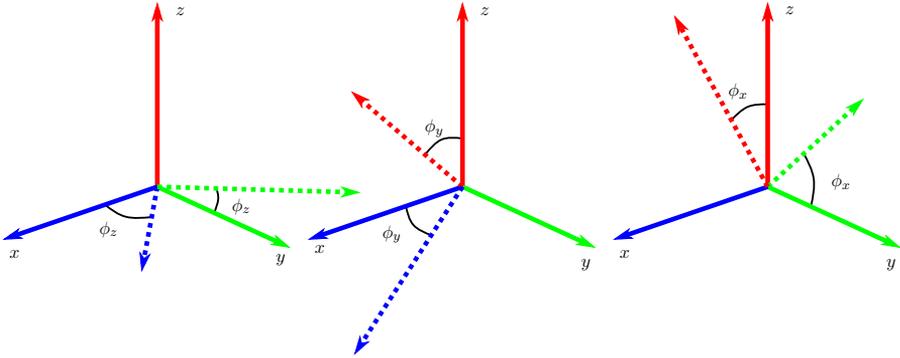


Figure 3.2: From left to right; rotations around the z -, y - and x -axes using the corresponding Euler angles ϕ_z , ϕ_y and ϕ_x .

opposite, i.e. $(R^{\text{SG}})^{-1} = (R^{\text{SG}})^{\top} = R^{\text{GS}}$. The columns of the rotation matrix R^{GS} give the coordinates of the three unit vector axes of the S-frame expressed in the G-frame. Therefore, a rotation matrix represents the relative orientation of one reference frame with respect to another.

A rotation can be represented in a number of ways. The rotation matrix is a non-minimal representation of a rotation. It has 9 elements, but each row and column is a unit vector, which gives 6 constraints. Hence a rotation can be described by 3 free variables. One possible parametrization uses Euler angles. If a reference frame is defined by the orthonormal basis vectors $\begin{bmatrix} x & y & z \end{bmatrix}$ where $x, y, z \in \mathbb{R}^3$, the Euler angles $\phi = \begin{bmatrix} \phi_x & \phi_y & \phi_z \end{bmatrix}^{\top}$ represent rotation angles around the x, y, z axes as indicated by the subscripts. The angles ϕ_x , ϕ_y and ϕ_z are commonly referred to as the roll, pitch and yaw angles, respectively. The complete rotation matrix can be constructed from a sequence of rotations around each individual axis, for example

$$\begin{aligned}
 R &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi_x & -\sin \phi_x \\ 0 & \sin \phi_x & \cos \phi_x \end{bmatrix} \begin{bmatrix} \cos \phi_y & 0 & \sin \phi_y \\ 0 & 1 & 0 \\ -\sin \phi_y & 0 & \cos \phi_y \end{bmatrix} \begin{bmatrix} \cos \phi_z & -\sin \phi_z & 0 \\ \sin \phi_z & \cos \phi_z & 0 \\ 0 & 0 & 1 \end{bmatrix}, \\
 &= R_x R_y R_z
 \end{aligned} \tag{3.2}$$

which represents a z - y - x rotation sequence. The rotations that correspond to R_z , R_y and R_x are shown in Figure 3.2. Note that the sequence of rotations is important; even if the Euler angles are the same, a different rotation sequence will generally yield a different rotation matrix. Another way to represent rotations is using unit quaternions, q , which are unit vectors in \mathbb{R}^4 . A unit quaternion that represent the same rotation as in (3.2) can

be constructed from the same Euler angles

$$q = \begin{bmatrix} \cos \frac{\phi_x}{2} \\ \sin \frac{\phi_x}{2} \\ 0 \\ 0 \end{bmatrix} \odot \begin{bmatrix} \cos \frac{\phi_y}{2} \\ 0 \\ \sin \frac{\phi_y}{2} \\ 0 \end{bmatrix} \odot \begin{bmatrix} \cos \frac{\phi_z}{2} \\ 0 \\ 0 \\ \sin \frac{\phi_z}{2} \end{bmatrix}, \quad (3.3)$$

$$= q_x \odot q_y \odot q_z$$

where \odot represents the quaternion product, which is a special algebraic operation for quaternions. More about quaternion algebra can be found in e.g. Särkkä (2007). Just like (3.2), we have that (3.3) represents the z - y - x rotation sequence, and changing the order of quaternion multiplications changes the rotation sequence accordingly. Unit quaternions can also be used to parametrize rotation matrices

$$R(q) = \begin{bmatrix} 2q_1^2 + 2q_2^2 - 1 & 2(q_2q_3 - q_1q_4) & 2(q_2q_4 + q_1q_3) \\ 2(q_2q_3 + q_1q_4) & 2q_1^2 + 2q_3^2 - 1 & 2(q_3q_4 - q_1q_2) \\ 2(q_2q_4 - q_1q_3) & 2(q_3q_4 + q_1q_2) & 2q_1^2 + 2q_4^2 - 1 \end{bmatrix} \quad (3.4)$$

$$q = [q_1 \quad q_2 \quad q_3 \quad q_4]^\top, \quad (3.5)$$

and provided that q represents the same rotation sequence as in (3.2), we have that (3.4) will produce the exact same rotation matrix.

3.1.2 Measurement models

In this section we introduce the models that are used to describe the measurements obtained from inertial sensors and magnetometers. Since the sensors are sampled in discrete time we will let, $t = 1, 2, \dots$, denote a discrete time index. The rotation matrix $R^{\text{SG}}(t)$ describes the relative orientation of the sensor frame with respect to the global frame. Furthermore, these models assume that the sensors are calibrated. Sensor calibration is discussed further in section 3.3 and is the main topic of Paper I.

Accelerometer

Accelerometer measurements are modelled as

$$y_a^{\text{S}}(t) = R^{\text{SG}}(t)(a^{\text{G}}(t) + g^{\text{G}}) + e_{a,t}, \quad (3.6)$$

where $a^{\text{G}}(t)$ is the linear acceleration of the sensor in the G-frame, g^{G} is the gravitational acceleration, which typically has one nonzero component along the vertical axis of the G-frame, and $e_{a,t}$ is an additive measurement noise term used to describe the uncertainties of the measurements. The most

common unit used for accelerometer measurements is $[\text{m/s}^2]$, but sometimes it is convenient to normalize the units by the magnitude of g , which is often referred to as a g -force.

Gyroscope

Gyroscope measurements are modelled as

$$y_g^S(t) = R^{SG}(t)\omega^G(t) + e_{g,t}, \quad (3.7)$$

where $\omega^G(t)$ is the angular velocity of the sensor in the G-frame and $e_{g,t}$ is measurement noise. The most common units used for gyroscope measurements are $[\text{rad/s}]$ or $[\text{°/s}]$.

Magnetometer

Magnetometer measurements are modelled as

$$y_m^S(t) = R^{SG}(t)m_0^G + e_{m,t}, \quad (3.8)$$

where m_0^G is the local Earth magnetic field, which has horizontal components pointing toward the magnetic north pole, and $e_{m,t}$ is measurement noise. Although the S.I. unit of the magnetic field strength is Tesla $[\text{T}]$, magnetometers often output measurements in arbitrary units $[\text{a.u.}]$ that are determined by their factory calibration. The reason for this is that in most applications it is the orientation of the magnetic field and not its strength that is of interest.

Measurement noise

The measurement noise terms in (3.6)–(3.8) are used to model the uncertainty in the measurements. This means that the measurements can be seen as the realizations of random variables, drawn from some probability distribution. The most common probabilistic model of these measurements is the multivariate Gaussian distribution, which has its probability density function (PDF) defined by

$$p(y(t)|\mu, \Sigma) = \mathcal{N}(y(t)|\mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^3|\Sigma|}} \exp\left(-\frac{1}{2}\|y(t) - \mu\|_{\Sigma^{-1}}^2\right), \quad (3.9)$$

where $\mu \in \mathbb{R}^{3 \times 1}$ is the mean, and $\Sigma \in \mathbb{R}^{3 \times 3}$ is a positive semi-definite covariance matrix. The norm inside the exponential is defined as $\|y(t) - \mu\|_{\Sigma^{-1}}^2 = (y(t) - \mu)^\top \Sigma^{-1} (y(t) - \mu)$, which is also known as the squared Mahalanobis distance. To examine whether the multivariate Gaussian distribution is a

good model for our sensors or not, we collected measurements from an Xsens MTw wireless motion tracker (Xsens 2017) that was at rest on a flat surface for one minute. We then compare the collected measurements with estimated Gaussian distributions. We use N measurements to approximate the mean using the sample mean

$$\mu \approx \bar{\mu} = \frac{1}{N} \sum_{t=1}^N y(t). \quad (3.10)$$

We assume that the elements corresponding to different sensor axes are independent, i.e. that the covariance matrix is diagonal

$$\Sigma = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_z^2 \end{bmatrix}, \quad (3.11)$$

where each σ_i^2 , $i = \{x, y, z\}$, is the scalar variance of the corresponding axis. The variance is then approximated by the sample variance

$$\sigma_i^2 \approx \bar{\sigma}_i^2 = \frac{1}{N-1} \sum_{t=1}^N (y_i(t) - \bar{\mu}_i)^2, \quad (3.12)$$

where we have used the sample mean from (3.10). We divide by $N - 1$ instead of N when the sample mean is used to obtain an unbiased estimator. Division by N will only yield an unbiased estimator of the variance if the true mean is known a priori or if the sample mean is computed using a different set of independent measurements. Figure 3.3 shows the normalized histograms of $N = 6000$ measurements compared to the estimated Gaussian PDF:s with means and variances corresponding to the sample means and sample variances. The histograms and the estimated Gaussian PDF:s appear to coincide for the most part. Note that since the sensors were at rest the measurements should contain zero acceleration and angular velocity, i.e. $a^G(t) = 0$ and $\omega^G(t) = 0$. If the measurement noise terms are modeled as zero mean Gaussian, then the sample means should correspond to $R^{SG}g^G$ for the accelerometer, $R^{SG}m_0^G$ for the magnetometer and 0 for the gyroscope. We can see that the sample means of the gyroscope appear to be nonzero. This can be accounted for in the gyroscope measurement model by considering an additional bias term, which we can estimate to calibrate the gyroscope. Sensor calibration is discussed later in Section 3.3.

3.2 Orientation estimation

As explained in section 3.1.1, the rotation matrix R^{GS} can be used to express quantities measured in the S-frame in the G-frame. This is certainly

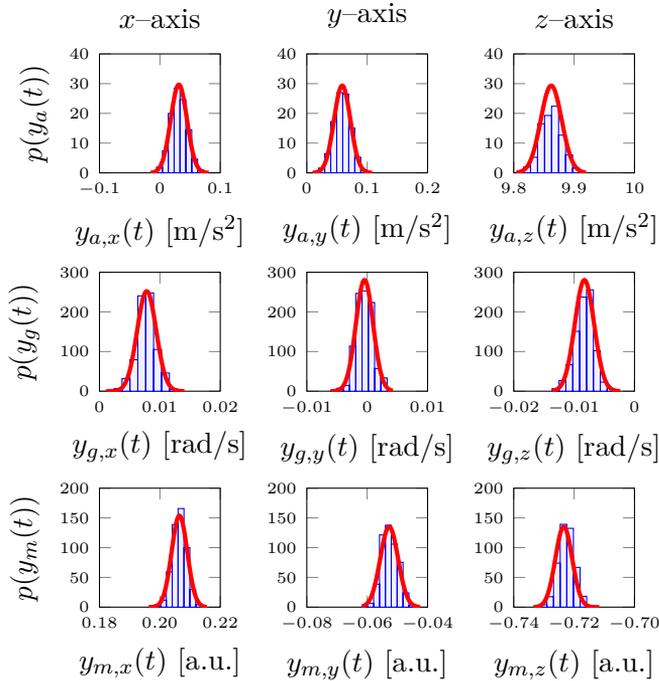


Figure 3.3: The normalized histograms of $N = 6000$ measurements collected from an Xsens MTw sensor that was at rest on a flat surface for one minute compared to the PDF:s of Gaussian distributions (red curves) with means and variances corresponding to the sample means and sample variances .

necessary to obtain measurements of quantities of interest, for example, the generalized coordinates in the biomechanical model presented in Chapter 2 are expressed in the G-frame. To estimate the orientation of a sensor platform we can combine the information from the gyroscope, accelerometer and magnetometer.

3.2.1 Extended Kalman filter

The Extended Kalman filter (EKF) is a well known adaption for non-linear state-space models of the popular Kalman filter proposed by Kalman (1960). We consider a state-space model on the form

$$x(t+1) = f(x(t), u(t), v_t) \quad (3.13)$$

$$y(t) = h(x(t)) + e_t, \quad (3.14)$$

where $f(\cdot)$ and $h(\cdot)$ are nonlinear differentiable functions. The state, $x(t)$, is the quantity that we are interested in estimating. Equation (3.13) is

the *process model* which describes how the state evolves in time, where $u(t)$ is a known input signal and $v_t \sim \mathcal{N}(0, \Sigma_v)$ is the *process noise*, which accounts for the uncertainties in the process model. Here we assume the process noise to be zero-mean Gaussian with covariance matrix Σ_v . Equation (3.14) is the *measurement model*, that describes the relationship between the measurements $y(t)$ and the state, and $e_t \sim \mathcal{N}(0, \Sigma_e)$ is the *measurement noise*, assumed to be additive zero-mean Gaussian with covariance matrix Σ_e .

The EKF is first initialized at the *state estimate* $\hat{x}(0|0)$ with the *state covariance* $P(0|0)$, which represents the prior knowledge of the state. After initialization, the EKF is run by performing a *time update* followed by a *measurement update* sequentially for as long as there is data available, i.e. $t = 1, \dots, N$. The time update performs a one-step ahead prediction using the process model (3.13) according to

$$\hat{x}(t|t-1) = f(\hat{x}(t-1|t-1), u(t), 0) \quad (3.15)$$

$$P(t|t-1) = F_t P(t-1|t-1) F_t^\top + G_t \Sigma_v G_t^\top, \quad (3.16)$$

where

$$F_t = \left. \frac{\partial}{\partial x(t)} f(x(t), u(t), v_t) \right|_{x(t)=\hat{x}(t-1|t-1), v_t=0} \quad (3.17)$$

$$G_t = \left. \frac{\partial}{\partial v_t} f(x(t), u(t), v_t) \right|_{x(t)=\hat{x}(t-1|t-1), v_t=0}. \quad (3.18)$$

The measurement update uses available measurements to update the predicted state estimates using the measurement model (3.14) according to

$$\tilde{y}(t) = y(t) - h(\hat{x}(t|t-1)) \quad (3.19)$$

$$S_t = H_t P(t|t-1) H_t^\top + \Sigma_e \quad (3.20)$$

$$K_t = P(t|t-1) H_t^\top S_t^{-1} \quad (3.21)$$

$$\hat{x}(t|t) = \hat{x}(t|t-1) + K_t \tilde{y}(t) \quad (3.22)$$

$$P(t|t) = (I - K_t H_t) P(t|t-1), \quad (3.23)$$

where

$$H_t = \left. \frac{\partial}{\partial x(t)} h(x(t)) \right|_{x(t)=\hat{x}(t|t-1)}, \quad (3.24)$$

$\tilde{y}(t)$ is often referred to as the *residual*, S_t is the *residual covariance* and K_t is the *Kalman gain*. The classic Kalman filter is the optimal state estimator for linear state-space models with additive Gaussian noise. The EKF replicates the Kalman filter by linearizing the state-space models with respect to the most recent state estimate as in (3.17), (3.18) and (3.24). Therefore, the EKF is not an optimal estimator in general, and its performance will depend on how close the state-space model is to a linear Gaussian model.

3.2.2 Using EKF for orientation estimation

In this thesis we primarily use unit quaternions to represent the rotations as shown in (3.4). Even though quaternions are parametrized by four free variables, they are often preferred over the three Euler angles. The main reason for this is that the Euler angle representation is ambiguous for some orientations, which can cause problems in the orientation estimation algorithms. One of the main problems that can occur is the phenomenon known as *gimbal lock*, which causes the changes in the first and third Euler angle in the rotation sequence to become indistinguishable for some critical value of the second Euler angle (Diebel 2006).

Estimates of the quaternions that represent the orientation of the sensor platform are obtained using an EKF. In an orientation estimation EKF, the process model describes the dynamic rotation of the sensor platform and uses the gyroscope measurements integrated over the sample interval to represent the incremental orientation update. Accelerometer measurements for situations where the sensor platform is close to stationary measure the gravitational acceleration vector, which is used as an observation of the inclination, the angle between the vertical axes in the S-frame and the G-frame. Magnetometer measurements can be used as measurements of the heading angle. An orientation estimation EKF can be formulated in different ways, either by using the quaternions as the state variable, or by using the orientation deviation from a linearization point as the state variable. These orientation estimation EKF algorithms are derived and described in more detail in Kok (2016).

3.3 Sensor calibration

The sensor models described in section 3.1.2 assumed that the sensors were calibrated. Here, we consider the case where the sensor models contain unknown parameters, θ . These parameters model sensor errors, biases and gains, some of which may vary with time or depend on physical aspects, such as the temperature of the sensor. This creates a need for re-calibration, but for shorter experiments it is often enough to perform an initial calibration and then assume that θ remains constant throughout the experiment. The calibration problem then boils down to a parameter estimation problem, to find an estimate, $\hat{\theta}$, of θ , which describes the observed measurements as accurately as possible. We formulate the calibration problem as a maximum likelihood (ML) problem.

3.3.1 Maximum likelihood estimation

The ML problem is to find the parameter estimate $\hat{\theta}$, that maximizes the likelihood function

$$\hat{\theta} = \arg \max_{\theta} p(y(1:N)|\theta) = \arg \max_{\theta} \prod_{t=1}^N p(y(t)|\theta), \quad (3.25)$$

where $p(y(t)|\theta)$ denotes the conditional probability distribution of the measurement $y(t)$ given the parameters θ . An equivalent formulation of (3.25) is

$$\hat{\theta} = \arg \min_{\theta} - \sum_{t=1}^N \ln p(y(t)|\theta), \quad (3.26)$$

since \ln is a monotone function. This formulation is often computationally convenient when $p(y(t)|\theta)$ belongs to the exponential family of probability distributions.

Example: Calibration of a biased gyroscope

One of the most common parameters to include in sensor calibration is additive bias. We extend the gyroscope measurement model (3.7) as

$$y_g(t) = R(t)\omega(t) + b_g + e_{g,t}, \quad (3.27)$$

where $\theta = b_g \in \mathbb{R}^3$ is the unknown bias parameter to estimate. The likelihood function of (3.27) is a multivariate Gaussian PDF:

$$\begin{aligned} p(y_g(t)|\theta) &= \mathcal{N}(y_g(t)|R(t)\omega(t) + b_g, \Sigma_g) \\ &= \frac{1}{\sqrt{(2\pi)^3|\Sigma_g|}} \exp\left(-\frac{1}{2} \|y_g(t) - R(t)\omega(t) - b_g\|_{\Sigma_g^{-1}}^2\right). \end{aligned} \quad (3.28)$$

A problem that we are faced with when trying to maximize (3.28) with respect to θ is that we also need to know the rotation matrix $R(t)$ and the angular velocity $\omega(t)$, which we cannot know exactly in general. The easiest way to circumvent this problem is to let the sensor remain stationary, i.e. $\omega(t) = 0$. Then the likelihood function becomes

$$\begin{aligned} p(y_g(t)|\theta) &= \mathcal{N}(y_g(t)|b_g, \Sigma_g) \\ &= \frac{1}{\sqrt{(2\pi)^3|\Sigma_g|}} \exp\left(-\frac{1}{2} \|y_g(t) - b_g\|_{\Sigma_g^{-1}}^2\right). \end{aligned} \quad (3.29)$$

Another possibility is to subject the sensor to a specific, known movement, where $R(t)$ and $\omega(t)$ are known exactly. However, this requires additional

equipment, which allows exact control of the movement of the sensor. Using the likelihood function in (3.29) we can compute the ML estimate

$$\begin{aligned}
 \hat{\theta} &= \arg \min_{\theta} -\ln p(y_g(t)|\theta) \\
 &= \arg \min_{\theta} \sum_{t=1}^N \frac{1}{2} \|y_g(t) - b_g\|_{\Sigma_g^{-1}}^2 + \ln \left(\sqrt{(2\pi)^3 |\Sigma_g|} \right) \\
 &= \arg \min_{\theta} \sum_{t=1}^N \frac{1}{2} \|y_g(t) - b_g\|_{\Sigma_g^{-1}}^2 \\
 &= \frac{1}{N} \sum_{t=1}^N y_g(t), \tag{3.30}
 \end{aligned}$$

where the last step is found by differentiating w.r.t. θ and solving $\frac{d}{d\theta} -\ln p(y_g(t)|\theta) = 0$ for θ . In this case, the ML estimate corresponds exactly to the sample mean of $y_g(1 : N)$.

3.3.2 Accelerometer and magnetometer calibration

When calibrating accelerometers and magnetometers it is common to at least include an additive bias parameter as was done for the gyroscope described in section 3.3.1. However, cheap, low-quality sensors may require more sophisticated calibration. For example, accelerometers in smart phones are known to have errors that vary with the orientation of the sensor, which cannot be described by one additive bias term. This is illustrated in Figure 3.4, where the magnitude of accelerometer measurements, collected from a MPU-6500 inertial sensor platform (InvenSense 2017) in a smart phone is plotted. We let the sensor be at rest in six different (approximately orthogonal) orientations for approximately one minute per orientation. It is clear that one constant bias term, which is fixed for all orientations, cannot describe the behaviour illustrated in Figure 3.4. In Paper I we present a practical method for accelerometer calibration that considers these types of errors. There we extend the sensor model in (3.6) to

$$y_a(t) = DR(t)(a(t) + g) + b_a + e_{a,t}, \tag{3.31}$$

where the unknown parameters are $\theta = \{D, b_a\}$, $D \in \mathbb{R}^{3 \times 3}$ and $b_a \in \mathbb{R}^3$. A similarly formulated method for magnetometer calibration have also been proposed in Kok and Schön (2016).

3.4 Calibration of the biomechanical model

In the previous section we discussed the calibration of the sensors themselves, which allows one to compensate for the intrinsic sensor errors and

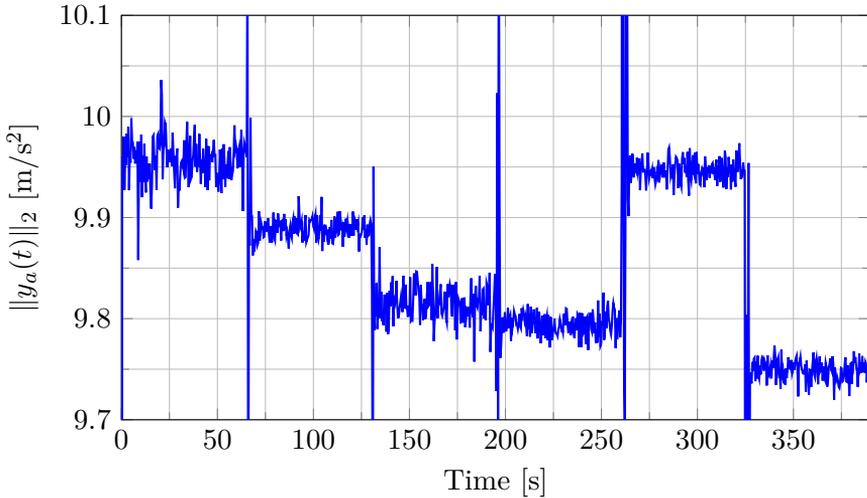


Figure 3.4: The magnitude of accelerometer measurements collected from a smart phone that was at rest in six different orientations. It can be seen that a change in orientation corresponds to a distinguishable change in the average magnitude.

biases that may exist. A different type of calibration problem is discussed in this section. It concerns how to relate the sensors to a biomechanical model, for example the inverted single or double pendulum that was presented in Chapter 2. It is important in motion capture to first determine the location of each sensor relative to the proximal and distal joints, and the orientation of each sensor with respect to the body segment to which the sensor is attached. Knowing where the sensors are located on the body makes it possible to track the relative movement of the body segments using the inertial and magnetic measurements. For example, the position of the joints in a kinematic chain model can be tracked by simply knowing the orientation of the sensor. The position of the joints can be found using inertial sensor data and by exploiting the kinematic constraints of the biomechanical model. The linear acceleration of a sensor attached to some body segment can be expressed in terms of its rotational movement around the center of rotation (the joint) and the linear, translational movement of the joint itself. This can be expressed as

$$a^S(t) = a_0^S(t) + \omega^S(t) \times (\omega^S(t) \times r^S) + \dot{\omega}^S(t) \times r^S, \quad (3.32)$$

where $a^S(t)$ is the linear acceleration of the sensor, $a_0^S(t)$ is the linear acceleration of the joint, $\omega^S(t)$ is the angular velocity and $\dot{\omega}^S(t)$ is the angular acceleration of the sensor. Assuming that the sensor is rigidly attached to

the body segment, the position of the joint, r^S , is fixed. By adding data from multiple sensors that all share a common center of rotation, we can identify r^S for each individual sensor by utilizing the fact that r^S and $a_0^S(t)$ all correspond to the same joint position r^G and acceleration $a_0^G(t)$ in the global frame. To be able to identify the joint position for a stationary joint, i.e. $a_0^G(t) = 0$, only one inertial sensor is required, while a moving joint requires at least two inertial sensors. The latter case is studied in more detail in Paper II where the identification of the joint position is formulated as an unconstrained optimization problem

$$\hat{\theta} = \arg \min_{\theta} V(\theta, y(1 : N)), \quad (3.33)$$

where θ are the unknown parameters to be estimated, in this case the joint positions, and $V(\theta, y(1 : N))$ is a cost function that depends on the parameters and the observed measurements $y(1 : N)$. Note that the ML estimator (3.26) described earlier can also be viewed as an unconstrained optimization problem where

$$V(\theta, y(1 : N)) = - \sum_{t=1}^N \ln p(y(t)|\theta). \quad (3.34)$$

How the cost function is formulated $V(\theta, y(1 : N))$ will have implications on how the optimization problem can be solved and how accurate we will be able to estimate θ given the data at hand. In the remainder of this section we illustrate some common choices of cost functions and how they can be solved, which provides a background for the methods compared in Paper II.

3.4.1 Linear least squares

A common choice of cost function is as a sum of squared errors

$$V(\theta, y(1 : N)) = \sum_{t=1}^N \|e(t, \theta)\|_2^2, \quad (3.35)$$

where $e(t, \theta)$ represents the estimation errors. Ideally, if the true parameters are θ_0 , then $V(\theta_0, y(1 : N)) = 0$ would hold if we were able to model the measurements $y(1 : N)$ exactly. This is not the case when we have noisy measurements, but ideally $V(\theta_0, y(1 : N))$ should be the global minimum of the cost function. If $e(t, \theta)$ is linear with respect to θ , e.g.

$$e(t, \theta) = y(t) - \varphi^\top(t)\theta, \quad (3.36)$$

where $\varphi^\top(t)$ can be a matrix and may depend on $y(1 : N)$ but not on θ , the cost function (3.35) becomes

$$V(\theta, y(1 : N)) = \sum_{t=1}^N \left\| y(t) - \varphi^\top(t)\theta \right\|_2^2, \quad (3.37)$$

and the unconstrained optimization problem becomes a *linear least squares* problem, which has the closed form solution

$$\hat{\theta} = \left(\sum_{t=1}^N \varphi(t)\varphi^\top(t) \right)^{-1} \sum_{t=1}^N \varphi(t)y(t), \quad (3.38)$$

provided that $\sum_{t=1}^N \varphi(t)\varphi^\top(t)$ is invertible.

3.4.2 Gradient descent

If the estimation errors are nonlinear in θ , the optimal estimator will not be available in closed form in general. In such cases we can use iterative optimization methods that, given some initial estimate $\hat{\theta}(0)$, finds a slightly better estimate that lowers the value of the cost function in each step, until it hopefully converges close to the true parameter θ_0 . One such method is the *gradient descent* method that updates the estimate in the negative direction of the gradient of the cost function

$$\hat{\theta}(k+1) = \hat{\theta}(k) - \alpha_k G(\hat{\theta}(k), y(1 : N)), \quad (3.39)$$

where

$$G(\hat{\theta}(k), y(1 : N)) = \left. \frac{d}{d\theta} V(\theta, y(1 : N)) \right|_{\theta=\hat{\theta}(k)}, \quad (3.40)$$

and $\alpha_k > 0$ is a scalar parameter that determines the step length of the iterative procedure.

3.4.3 Hessian-based methods

The gradient descent method tends to converge slower as the estimate approaches a minimum. To speed up the convergence considerably, one can include information about the second derivatives, or the Hessian matrix of the cost function, in the update step. The iterative procedure in (3.39) then becomes

$$\hat{\theta}(k+1) = \hat{\theta}(k) - \alpha_k H^{-1}(\hat{\theta}(k), y(1 : N)) G(\hat{\theta}(k), y(1 : N)), \quad (3.41)$$

where

$$H(\hat{\theta}(k), y(1:N)) = \frac{d^2}{d\theta^2} V(\theta, y(1:N)) \Big|_{\theta=\hat{\theta}(k)}. \quad (3.42)$$

This method is known as *Newton's method* or the *Newton-Raphson method* (Boyd and Vandenberghe 2004).

For many problems, the Hessian can be difficult to compute. In that case one can approximate the Hessian instead. One such method that is applicable in the case where we have a cost function that is a sum of squared errors as in (3.35) is the Gauss-Newton method. In the Gauss-Newton method (Wright and Nocedal 2006), the iterative procedure is given by

$$\hat{\theta}(k+1) = \hat{\theta}(k) - \alpha_k \left(J_k^\top J_k \right)^{-1} J_k e(1:N, \hat{\theta}(k)), \quad (3.43)$$

where $e(1:N, \hat{\theta}(k))$ is a column vector with all errors and J_k is the Jacobian matrix of the error vector

$$J(\hat{\theta}(k), y(1:N)) = J_k = \begin{bmatrix} \frac{\partial}{\partial \theta_1} e(1, \theta) & \dots & \frac{\partial}{\partial \theta_m} e(1, \theta) \\ \vdots & & \vdots \\ \frac{\partial}{\partial \theta_1} e(N, \theta) & \dots & \frac{\partial}{\partial \theta_m} e(N, \theta) \end{bmatrix}. \quad (3.44)$$

For the iterative optimization algorithms mentioned above, the step length α_k is an important parameter for convergence. A large α_k could get your estimates closer to the minimum in less iterations but may cause the estimates to overshoot when they get too close. On the other hand a small α_k can guarantee that the estimates will end up closer to the minimum, but will require more iterations. One possibility is to adaptively choose the step length for each iteration to try and improve the convergence rate without sacrificing too much accuracy. This can be done using *line search* algorithms (Boyd and Vandenberghe 2004).

3.5 Identification of neuromuscular control

In Chapter 2 we introduced biomechanical models and neuromuscular controller models that are commonly used to study human balance. In this section we discuss how data from inertial sensors can be used to identify the unknown dynamics of a neuromuscular controller in standing human balance. For this purpose, we use the methodology of *system identification*. This section will only give a very brief introduction to the methodology in the context of the specific application within this thesis. For more general and extensive introductions to the field, the reader may refer some of the

many books that have been written on the subject (Ljung 1999; Pintelon and Schoukens 2012; Söderström and Stoica 1989).

The methodology of system identification may be summarized in the following four steps:

- Step 1. Experiment:** Design and perform an experiment on the system to be identified. The outcome of the experiment is a data set, \mathcal{D} , that consists of empirical observations of the inputs and outputs of the system.
- Step 2. Model selection:** Assume a model structure, a set of mathematical models that could explain the dynamics of the system.
- Step 3. Estimation:** Find the *best* model within the chosen model structure given a subset of the observed data, called the *identification data*. What is considered *best* depends on the application of the model.
- Step 4. Validation:** See how well the estimated model can explain a different subset of the observed data, called the *validation data*.

This procedure is often reiterated, starting at any of the four steps depending on the outcome of the procedure, until a satisfactory result is obtained. For example, it may be necessary to perform a new experiment if the original data is of poor quality or try different model structures if the estimated model is not able to explain the validation data.

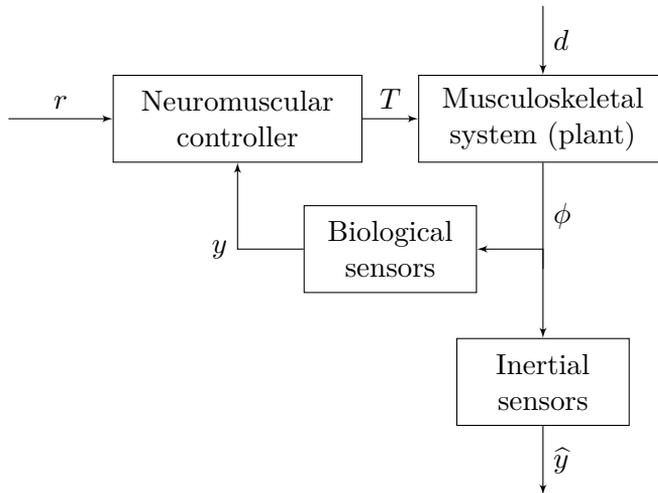


Figure 3.5: A closed neuromuscular control loop in humans where inertial sensors provide measurements.

In section 2.4 we described how a standing human can be modeled as a closed loop system, where the biomechanical model of a standing human is the *plant* that is controlled by the neuromuscular controller. We now extend this model to include (external) inertial sensors, which will provide us with measurements that we can use for identification, see Figure 3.5. The data that we use for identification is denoted

$$\mathcal{D} = \left\{ \left(\hat{y}(t), \hat{T}(t) \right) \right\}, \quad t = 1, \dots, N, \quad (3.45)$$

where $\hat{y}(t)$ and $\hat{T}(t)$ are the observations we have obtained from the inertial sensors that relates to the controller input and output, respectively. Since the input to the neuromuscular controller comes from the biological sensors, we have no way of knowing exactly what information that is used. We will therefore assume that the neuromuscular controller have access only to information that we can obtain from the inertial sensors. Specifically, we may obtain information about the joint angles, ϕ , and angular velocities, $\dot{\phi}$, from the gyroscope and through the use of orientation estimation. This assumption is also used in the PD controller models described in section 2.4.2. Observations of the controller output, the joint torques, T , is not as straightforward to obtain since it also acts as input to the plant. The best we can do with inertial sensors alone is to first model the plant, which in this case corresponds to the biomechanical models described in Chapter 2. Once we have established a model for the plant it is possible to estimate T , using the inverse dynamics of the plant model. This procedure is described in more detail in Paper III. The remainder of this section will focus on the system identification methodology, the four steps presented earlier.

3.5.1 Experiment design

To obtain data for identification and validation we need to perform an experiment where we can observe the inputs and outputs of the system. The most important aspect of the experiment, which will determine how much information we can obtain about the system, is *excitation*. By subjecting the system to an external excitation signal and measuring how the system responds we can obtain more informative measurements than without the excitation. By tailoring the excitation signal we can choose to excite the system in a frequency band of interest and for a desired amplitude range of the inputs and outputs. Therefore, typical excitation signals are sums of periodic signals, so called *multisines*, or filtered white-noise signals. In general, for nonlinear systems, the information we obtain from the experiment depends on the excitation in both frequency and amplitude, and we can only hope for our models to describe the system behavior within the excited regions. For linear systems however, it is sufficient to have excitation in

frequency as the system dynamics do not change with the amplitude of the excitation.

For the closed neuromuscular control loop in Figure 3.5, we have two external inputs to the system that could in theory be used as an excitation signal. The reference signal, r , that acts as an input to the neuromuscular controller, and an external disturbance, d , that acts as an input to the plant, or biomechanical model. The reference signal is determined by the task that the human wants to perform, and is therefore unsuitable as an excitation signal. You can affect the reference signal by changing the task that is to be performed, but it is not possible to determine or measure the reference signal with currently available technology. Thus, external disturbances are the most common type of excitation. The experimental setups used to generate external disturbances often consist of moving or tilting platforms, linear motors that push and pull on the subject at specific locations or combinations of these (Boonstra et al. 2013; Cenciari et al. 2010; Maki et al. 1987), see Figure 3.6. These disturbances can be seen as external forces that act on the subject, and through the biomechanical model it is possible to calculate their contribution as torques that may be added to the joint torques, T , in which case the input to the plant becomes $T + d$. Other types of external disturbances, that instead attempt to disturb the biological sensors, have been used. For example, moving visual scenes (Carver et al. 2006; Kuo 2005), galvanic vestibular stimulation (Fitzpatrick and Day 2004; Fransson et al. 2003) or vibratory muscle stimulation (Vuillerme et al. 2002). These types of disturbances are useful if you want to determine the contributions from specific biological sensors to human balance, but requires models of the biological sensors, such that the effect of the disturbances can be determined.

Another important aspect of experiment design is to make sure that the sensors are properly calibrated and that their relationship to the plant dynamics is determined. It is always better to take some extra time, if the conditions allow for it, to make sure that the experimental conditions are as close to ideal as possible, since the quality of the measurements will directly impact all subsequent steps of the identification procedure.

3.5.2 Model selection

Selecting an appropriate model structure will affect both how well the model is able to describe the observed data and how complicated it is to determine the *best* model. Down below we will discuss some of the most common model structures, and their potential strengths and weaknesses. We will assume the general form of the neuromuscular control law

$$T(t) = h(\varphi(t), \theta), \quad (3.46)$$

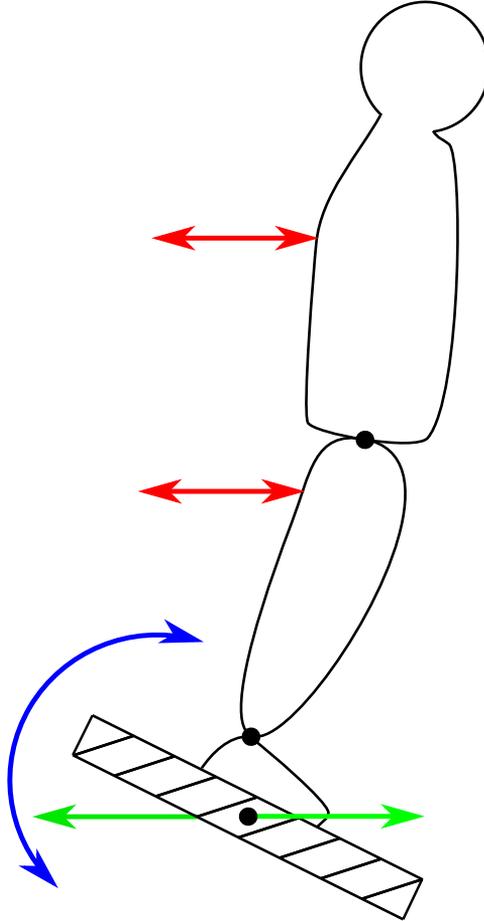


Figure 3.6: Illustrates three different commonly used external disturbances used to excite the human balance system. Horizontally moving platform (green), tilting platform (blue) and pushing/pulling at specific locations on the segments (red).

where we let $\varphi(t)$ represent the information that we assume that the controller has access to at time t and θ are a set of parameters that are constant. Note that we can include time-delayed information in $\varphi(t)$, and therefore the only restrictions of (3.46) when compared to the general control law in (2.31), is that we assume that θ and the functional form of $h(\cdot)$ are not time dependent.

One of the first things to consider before choosing a model structure is if there is any prior knowledge about the system that could be incorporated in the model. For example, the biomechanical models described in Chapter 2 assume that the plant dynamics follow the laws of classical Newtonian mechanics. Such models are often referred to as *grey box* models, where causal relationships may be partially known from first principles such as Newton's laws or Maxwell's equations, but some physical parameters are unknown and have to be estimated from the data. If certain physical laws apply to your system it may be beneficial to formulate a grey box model, but often this is not the case. Assumptions and approximations can be made so that grey box models may be formulated, but this can quickly become cumbersome for complex systems. Therefore, alternative model structures that are flexible, yet "simple" are often sought.

Since our data \mathcal{D} consists of observations of the controller inputs and outputs, one of the least restricting assumptions to make about $h(\cdot)$ is that it is causal, meaning it depends only on the current and past plant outputs and past controller outputs. These types of models are commonly referred to as *black box* models. The information that such a controller has access to at time t is then

$$\varphi(t) = \begin{bmatrix} T(t-1) & T(t-2) & \dots & T(t-n_a) \\ \phi(t) & \phi(t-1) & \dots & \phi(t-n_b+1) \end{bmatrix}^\top \quad (3.47)$$

where n_a and n_b determine the *model order*, how many past inputs and outputs that $T(t)$ depends on. Choosing n_a and n_b is therefore one aspect in specifying a black box model structure. Another aspect is to select a functional form for $h(\cdot)$. A linear model could have the form

$$T(t) = \varphi^\top(t)\theta \quad (3.48)$$

$$\theta = \begin{bmatrix} a_1 & a_2 & \dots & a_{n_a} & b_1 & b_2 & \dots & b_{n_b} \end{bmatrix}^\top, \quad (3.49)$$

which is an autoregressive exogenous (ARX) model. Nonlinear black box models are more diverse, but one common choice is a *function expansion*

$$T(t) = \sum_{k=1}^{n_k} w_k \gamma_k(\varphi(t)), \quad (3.50)$$

where $\theta = \{w_1, w_2, \dots, w_{n_k}\}$ are scalar weights and $\gamma_k(\cdot)$ is an arbitrary nonlinear *basis function*. Many popular choices of basis functions exist, and the reader is referred to the system identification literature for more on the subject.

3.5.3 Estimation

To find the *best* possible model within the chosen model structure we must first define what it means for our model to be *good*. A common use of mathematical models is prediction. Given the input $\varphi(t)$ we want the model to predict the output $T(t)$ as accurately as possible. The best model in that case is one that minimizes the prediction errors, which can be formulated as an optimization problem for the model parameters θ

$$\hat{\theta} = \arg \min_{\theta} \sum_{t=1}^N \mathcal{L} \left(\hat{T}(t) - h(\hat{\varphi}(t), \theta) \right), \quad (3.51)$$

where $\hat{T}(t)$ and $\hat{\varphi}(t)$ belong to our identification data and $\mathcal{L}(\cdot)$ is a scalar valued *loss function*, that attains its minimum value when its argument is zero and increases in value as the argument deviates from zero. This class of estimation methods is known as *prediction error methods* (PEM), and the purpose of the loss function is to penalize larger prediction errors such that $\hat{\theta}$ will yield the predictor model with a minimum loss. The optimization problem (3.51) can then be solved using the techniques described in sections 3.4.1–3.4.3.

Some common choices of loss functions are the squared Euclidean (l_2) norm $\|\cdot\|_2^2$ or the l_1 -norm $\|\cdot\|_1$. The difference between these two loss functions is that the squared Euclidean norm will penalize larger prediction errors more than the l_1 -norm. Penalizing large errors too much can sometimes have detrimental effects if your data contains *outliers*, measurements that significantly deviate from the true model. This can for example happen if the sensors are subjected to unforeseen disturbances. In that case, the outliers, which are often a minority set of the whole data set, will have an unproportionally large affect on the estimated parameters. In that sense, the l_1 -norm is said to be more robust against outliers, as it penalizes small errors more than the l_2 -norm, and is frequently used for this reason. However, the squared Euclidean norm will produce a cost function that is a sum of squared errors, which allows faster optimization methods, such as the Gauss-Newton method (3.43), to be used. The l_1 -norm produces a cost function that is more difficult to minimize, since the l_1 -norm is not differentiable at the zero argument.

3.5.4 Validation

Once we have obtained the *best* model $h(\varphi(t), \hat{\theta})$ we need to evaluate its performance on data that is separate from the identification data. The reason for this is to make sure that the model generalizes to new unseen data. It may be possible to find an almost perfect predictor model for the identification data if we have a flexible model structure with a lot of parameters, which then turns out to perform poorly on separate validation data. This is known as *overfitting*. Here we also assume the *true model* is not available, in which case validation could have been made by comparing $\hat{\theta}$ to the true parameters.

To evaluate the performance of the predictor model there are several performance metrics that can be used. One straight forward metric used to evaluate models estimated using PEM is the average loss

$$\mathcal{L}_{\text{avg}} = \frac{1}{N} \sum_{t=1}^N \mathcal{L} \left(\hat{T}(t) - h(\hat{\varphi}(t), \theta) \right), \quad (3.52)$$

based directly on the cost function. Two other commonly used performance metrics are the mean squared error (MSE) and its square root (RMSE), which are defined as

$$\text{MSE} = \frac{1}{N} \sum_{t=1}^N \|\hat{T}(t) - h(\hat{\varphi}(t), \hat{\theta})\|_2^2 \quad (3.53)$$

$$\text{RMSE} = \sqrt{\text{MSE}}. \quad (3.54)$$

The MSE is an estimate of the second order moment of the prediction errors, which is a metric for the variance and bias of the predictor. For unbiased predictors the MSE and RMSE is an estimate of the variance and the standard deviation of the prediction errors, respectively. Smaller MSE and RMSE is typically considered better. However, in some cases a biased predictor may be considered better than an unbiased predictor if the variance is smaller, in which case it makes more sense to use bias and variance as separate performance metrics instead. Thus, it is important to keep in mind that the metric you choose for evaluation should reflect what you want the model to be good at. Then the metric can be used to compare the performance of different models. If a model predicts the location of a pirate treasure, you might also want it to minimize the amount of digging required.

Another type of validation is to simulate the model output according to

$$\hat{T}_s(t, \theta) = h(\hat{\varphi}_s(t), \hat{\theta}), \quad (3.55)$$

where $\widehat{\varphi}_s(t)$ may contain past simulated outputs. For example, when simulating an ARX model we have that

$$\widehat{\varphi}_s(t) = \begin{bmatrix} \widehat{T}_s(t-1) & \widehat{T}_s(t-2) & \dots & \widehat{T}_s(t-n_a) \\ \widehat{\phi}(t) & \widehat{\phi}(t-1) & \dots & \widehat{\phi}(t-n_b+1) \end{bmatrix}^\top \quad (3.56)$$

$$\widehat{T}_s(t) = \begin{cases} \widehat{T}(t) & t < n_a + 1 \\ h(\widehat{\varphi}_s(t), \widehat{\theta}) & \text{otherwise} \end{cases} \quad (3.57)$$

Evaluating the average loss, MSE or RMSE for the simulated outputs are not as straight forward as for the predicted outputs, since the simulation errors often tend to grow larger over time, whereas the prediction errors often are stationary. In such scenarios, the FIT metric is often a better metric for the simulated model outputs, and is defined as

$$\text{FIT} = 100 \left(1 - \frac{\|\widehat{T}(1:N) - \widehat{T}_s(1:N, \theta)\|_2}{\|\widehat{T}(1:N) - \bar{T}\mathbf{1}\|_2} \right), \quad (3.58)$$

where \bar{T} is the empirical mean of $\widehat{T}(1:N)$ and $\mathbf{1}$ is a vector of ones. A FIT of 100 means that the simulated output matches the validation data exactly, while a negative FIT indicates that simply choosing the empirical mean of the validation data as model output would outperform the actual model output.

Chapter 4

Concluding remarks

This chapter is a short summary of the main contributions of this thesis, which are presented in Paper I–IV. This is followed by an outlook towards possible future work, since this Licentiate thesis marks roughly the half-way towards a PhD thesis.

4.1 Summary of contributions

The main contributions of this thesis are the applications and practical use of the presented methods. Each paper describes possible solutions to existing problems and shows results that are based on measurements from real sensors that we have collected ourselves. In Paper I we show how low-quality accelerometers can be calibrated to compensate for intrinsic sensor errors. Paper II connects the inertial sensors to the biomechanical models that we use to describe the dynamics of human movement. We show how inertial sensors can be used in nonlinear system identification in Paper III and in Paper IV we propose a method using smart phone sensors for quantifying hand tremor, a movement disorder that is a primary symptom in neurological disorders such as Parkinson’s disease or Essential tremor. Personally, I believe that these papers are able to illustrate not only the usefulness of the methods themselves, but also the wide utility and accessibility of inertial and magnetic sensors. In the future, such sensors could be regular tools that are used in a manner similar to rulers and stopwatches.

4.2 Future work

As described in the introduction to this thesis, the main application of interest is the assessment of human balance, and it is in that direction that the future research is currently heading. Of particular interest is the

identification of neuromuscular control, which we view as the main mechanisms behind human balance. This is the motivation for large portions of Chapter 2 and Chapter 3. The goal of our research is an unobtrusive method for identifying the balancing mechanisms in individuals, that can be used by physicians to make assessments and assist with tailoring an individualized therapy. With that said, there are several obstacles that we have identified, that have to be dealt with before such a method can be effectively used.

- Ideally data should be collected from individuals without the need for them to actively engage with the sensors or any external equipment. This poses several challenges. The sensors have to be recalibrated automatically when the need arises. Performing an on-line calibration scheme when the sensors have access to recently collected, good quality, calibration data, will most likely be needed.
- We have to be able to infer what the individual is doing based on the collected data, which not only means activity recognition, but also to know when specific biomechanical models match with what the sensors are measuring. Exploiting the kinematic constraints of the biomechanical model, as in Paper II, could be a good starting point, as large errors in the cost function could give away the measurements that do not follow the model.
- Finding a suitable experimental method and model structure for the identification of neuromuscular control. We have already begun reviewing present literature about the identification of neuromuscular controllers in human balance, many of these references can be found in the related sections in Chapter 2 and Chapter 3. Our ambition is to review the methods and models that have been studied so far, and evaluate how well these methods can be adapted and applied in a mobile setting. A model structure that can not only accurately describe the dynamics of the neuromuscular controller, but also be interpreted and used for assessment is desired.
- As discussed in section 3.5.1, to obtain informative observations of the neuromuscular control mechanisms, we need to apply an external excitation to the closed loop system that is human balance. Traditionally, such excitations are applied using moving platforms or linear motors that push and pull on the subject, something that requires a non-mobile experimental setup. Finding a method to excite the balance system enough so that the identified model can be used for assessment, is another important problem to solve.

Glossary and notation

Abbreviations

Lists abbreviations used in the introductory chapters by order of appearance.

CNS	Central nervous system
PD	Parkinson's disease (in the context of neurological disorders) <i>or</i> Proportional derivative (in the context of controllers)
ET	Essential tremor
DBS	Deep brain stimulation
CoM	Center of mass
EoM	Equation of motion
LTI	Linear time invariant
LQ	Linear quadratic
FRF	Frequency response function
MEMS	Microelectromechanical systems
PDF	Probability density function
EKF	Extended Kalman filter
ML	Maximum likelihood
ARX	Autoregressive exogenous
PEM	Prediction error method
MSE	Mean squared error
RMSE	Root mean squared error

References

- Avci, A., S. Bosch, M. Marin-Perianu, R. Marin-Perianu and P. Havinga (2010). ‘Activity recognition using inertial sensing for healthcare, well-being and sports applications: A survey’. In: *Architecture of computing systems (ARCS), 2010 23rd international conference on*. VDE, pp. 1–10.
- Boonstra, T. A., A. C. Schouten and H. Van der Kooij (2013). ‘Identification of the contribution of the ankle and hip joints to multi-segmental balance control’. In: *Journal of neuroengineering and rehabilitation* 10.1, p. 23.
- Boyd, S. and L. Vandenberghe (2004). *Convex optimization*. Cambridge university press.
- Carver, S., T. Kiemel and J. J. Jeka (2006). ‘Modeling the dynamics of sensory reweighting’. In: *Biological cybernetics* 95.2, pp. 123–134.
- Cenciarini, M., P. J. Loughlin, P. J. Sparto and M. S. Redfern (2010). ‘Stiffness and damping in postural control increase with age’. In: *IEEE Transactions on biomedical engineering* 57.2, pp. 267–275.
- Diebel, J. (2006). ‘Representing attitude: Euler angles, unit quaternions, and rotation vectors’. In: *Matrix* 58.15-16, pp. 1–35.
- Engelhart, D., T. A. Boonstra, R. G. K. M. Aarts, A. Schouten and H. van der Kooij (2016). ‘Comparison of closed-loop system identification techniques to quantify multi-joint balance control’. In: *Annual Reviews in Control* 41, pp. 58–70.
- Fitzpatrick, R. C. and B. L. Day (2004). ‘Probing the human vestibular system with galvanic stimulation’. In: *Journal of applied physiology* 96.6, pp. 2301–2316.
- Fransson, P.-A., A. Hafstrom, M. Karlberg, M. Magnusson, A. Tjader and R. Johansson (2003). ‘Postural control adaptation during galvanic vestibular and vibratory proprioceptive stimulation’. In: *IEEE Transactions on Biomedical Engineering* 50.12, pp. 1310–1319.
- Gage, W. H., D. A. Winter, J. S. Frank and A. L. Adkin (2004). ‘Kinematic and kinetic validity of the inverted pendulum model in quiet standing’. In: *Gait & posture* 19.2, pp. 124–132.
- Gustafsson, F. (2010). *Statistical sensor fusion*. Studentlitteratur.

- Halvorsen, K. and F. Olsson (2016). ‘Pose estimation of cyclic movement using inertial sensor data’. In: *Statistical Signal Processing Workshop (SSP), 2016 IEEE*. (Palma de Mallorca, Spain). IEEE, pp. 1–5.
- Halvorsen, K. and F. Olsson (2017). ‘Robust tracking of periodic motion in the plane using inertial sensor data’. In: *IEEE Sensors 2017*. (Glasgow, Scotland).
- Herculano-Houzel, S. (2009). ‘The human brain in numbers: a linearly scaled-up primate brain’. In: *Frontiers in human neuroscience* 3, p. 31.
- Hol, J. D. (June 2011). ‘Sensor Fusion and Calibration of Inertial Sensors, Vision, Ultra Wideband and GPS’. PhD thesis. Sweden: Linköping University.
- InvenSense (Sept. 2017). *6 Axis MotionTracking*. URL: <http://www.invensense.com/products/motion-tracking/6-axis/>.
- Kailath, T. (1980). *Linear systems*. Vol. 156. Prentice-Hall Englewood Cliffs, NJ.
- Kalman, R. E. (1960). ‘A new approach to linear filtering and prediction problems’. In: *Journal of basic Engineering* 82.1, pp. 35–45.
- Kok, M. (2016). ‘Probabilistic modeling for sensor fusion with inertial measurements’. PhD thesis. Linköping University.
- Kok, M. and T. B. Schön (2016). ‘Magnetometer calibration using inertial sensors’. In: *IEEE Sensors Journal* 16.14, pp. 5679–5689.
- Kok, M., N. Wahlström, T. B. Schön and F. Gustafsson (2013). ‘MEMS-based inertial navigation based on a magnetic field map’. In: *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, pp. 6466–6470.
- Kooij, H. van der, E. van Asseldonk and F. C. van der Helm (2005). ‘Comparison of different methods to identify and quantify balance control’. In: *Journal of neuroscience methods* 145.1-2, pp. 175–203.
- Kranz, M., A. Möller, N. Hammerla, S. Diewald, T. Plötz, P. Olivier and L. Roalter (2013). ‘The mobile fitness coach: Towards individualized skill assessment using personalized mobile devices’. In: *Pervasive and Mobile Computing* 9.2, pp. 203–215.
- Kuo, A. D. (1995). ‘An optimal control model for analyzing human postural balance’. In: *IEEE transactions on biomedical engineering* 42.1, pp. 87–101.
- Kuo, A. D. (2005). ‘An optimal state estimation model of sensory integration in human postural balance’. In: *Journal of Neural Engineering* 2.3, S235.
- Lantmäteriet (Mar. 2018). *Tyngdkraften*. URL: <https://www.lantmateriet.se/sv/Kartor-och-geografisk-information/GPS-och-geodetisk-matning/Referenssystem/Tyngdkraftssystem/Tyngdkraften/>.

- LaViola Jr, J. J. (2008). ‘Bringing VR and spatial 3D interaction to the masses through video games’. In: *IEEE Computer Graphics and Applications* 28.5.
- Ljung, L. (1999). *System Identification: theory for the user*. 2nd. Upper Saddle River, NJ: Prentice Hall.
- Mahboobin, A., P. J. Loughlin, M. S. Redfern and P. J. Sparto (2005). ‘Sensory re-weighting in human postural control during moving-scene perturbations’. In: *Experimental Brain Research* 167.2, pp. 260–267.
- Maki, B. E., P. J. Holliday and G. R. Fernie (1987). ‘A posture control model and balance test for the prediction of relative postural stability’. In: *IEEE Transactions on Biomedical Engineering* 10, pp. 797–810.
- Mattsson, P., D. Zachariah and P. Stoica (2018). ‘Recursive nonlinear-system identification using latent variables’. accepted for publication in *Automatica*.
- Medvedev, A., F. Olsson and T. Wigren (2017). ‘Tremor quantification through data-driven nonlinear system modeling’. In: *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*. (Melbourne, Australia). IEEE, pp. 5943–5948.
- Olsson, F. and K. Halvorsen (2017). ‘Experimental evaluation of joint position estimation using inertial sensors’. In: *Information Fusion (Fusion), 2017 20th International Conference on*. (Xi’an, China). IEEE, pp. 1–8.
- Olsson, F., K. Halvorsen, D. Zachariah and P. Mattsson (2018). *Identification of nonlinear feedback mechanisms operating in closed loop using inertial sensors*. To be presented at the 18th IFAC symposium on system identification (SYSID). Stockholm, Sweden.
- Olsson, F., M. Kok, K. Halvorsen and T. B. Schön (2016). ‘Accelerometer calibration using sensor fusion with a gyroscope’. In: *Statistical Signal Processing Workshop (SSP), 2016 IEEE*. (Palma de Mallorca, Spain). IEEE, pp. 1–5.
- Olsson, F. and A. Medvedev (2018a). *Non-parametric Time-domain Tremor Quantification with Smart Phone for Therapy Individualization*. Submitted for publication.
- Olsson, F. and A. Medvedev (2018b). *Tremor severity rating by Markov chains*. To be presented at the 18th IFAC symposium on system identification (SYSID). Stockholm, Sweden.
- Park, S., F. B. Horak and A. D. Kuo (2004). ‘Postural feedback responses scale with biomechanical constraints in human standing’. In: *Experimental brain research* 154.4, pp. 417–427.
- Peterka, R. J. (2002). ‘Sensorimotor integration in human postural control’. In: *Journal of neurophysiology* 88.3, pp. 1097–1118.

- Peterka, R. J. and P. J. Loughlin (2004). ‘Dynamic regulation of sensorimotor integration in human postural control’. In: *Journal of neurophysiology* 91.1, pp. 410–423.
- Pintelon, R. and J. Schoukens (2012). *System Identification: A frequency domain approach*. 2nd. Hoboken, NJ: John Wiley & Sons, Inc.
- Roetenberg, D., H. Luinge and P. Slycke (2009). ‘Xsens MVN: full 6DOF human motion tracking using miniature inertial sensors’. In: *Xsens Motion Technologies BV, Tech. Rep 1*.
- Rugh, W. J. (1996). *Linear system theory*. Vol. 2. prentice hall Upper Saddle River, NJ.
- Särkkä, S. (2007). *Notes on quaternions*. Tech. rep. Helsinki University of Technology.
- Söderström, T. and P. Stoica (1989). *System identification*. Prentice-Hall.
- Solin, A., M. Kok, N. Wahlström, T. B. Schön and S. Särkkä (2015). ‘Modeling and interpolation of the ambient magnetic field by Gaussian processes’. In: *arXiv preprint arXiv:1509.04634*.
- Stoica, P., R. L. Moses et al. (2005). *Spectral analysis of signals*. Vol. 1. Pearson Prentice Hall Upper Saddle River, NJ.
- Vuillerme, N., F. Danion, N. Forestier and V. Nougier (2002). ‘Postural sway under muscle vibration and muscle fatigue in humans’. In: *Neuroscience letters* 333.2, pp. 131–135.
- WHO, Department of Ageing and Life Course (ALC) (2008). ‘WHO global report on falls prevention in older age’. In: *World Health Organization*.
- Winter, D. A. (1995). ‘Human balance and posture control during standing and walking’. In: *Gait & posture* 3.4, pp. 193–214.
- Winter, D. A., A. E. Patla, F. Prince, M. Ishac and K. Gielo-Perczak (1998). ‘Stiffness control of balance in quiet standing’. In: *Journal of neurophysiology* 80.3, pp. 1211–1221.
- Woodman, O. J. (2007). *An introduction to inertial navigation*. Tech. rep. University of Cambridge, Computer Laboratory.
- Wright, S. and J. Nocedal (2006). *Numerical Optimization*. 2nd. Springer Series in Operations Research.
- Xsens (2017). *MTw Awinda Wireless Motion Tracker*. (Online) <https://www.xsens.com/download/pdf/documentation/mtw2-awinda/MTw2-Awinda.pdf> accessed December 2017.

Title

Accelerometer calibration using sensor fusion with a gyroscope

Authors

Fredrik Olsson, Manon Kok, Kjartan Halvorsen and Thomas B. Schön

Edited version of

F. Olsson, M. Kok, K. Halvorsen and T. B. Schön (2016). ‘Accelerometer calibration using sensor fusion with a gyroscope’. In: *Statistical Signal Processing Workshop (SSP), 2016 IEEE*. (Palma de Mallorca, Spain). IEEE, pp. 1–5

Accelerometer calibration using sensor fusion with a gyroscope

Abstract

In this paper, a calibration method for a triaxial accelerometer using a triaxial gyroscope is presented. The method uses a sensor fusion approach, combining the information from the accelerometers and gyroscopes to find an optimal calibration using Maximum likelihood. The method has been tested by using real sensors in smartphones to perform orientation estimation and verified through Monte Carlo simulations. In both cases, the method is shown to provide a proper calibration, reducing the effect of sensor errors and improving orientation estimates.

5.1 Introduction

Accelerometers and gyroscopes (inertial sensors) measure linear acceleration and angular velocity, respectively. By combining three orthogonal accelerometers and three orthogonal gyroscopes, it is possible to measure in three dimensions. These types of sensors have many different applications, for example in navigation and motion capture (Serrano and Ayazi 2015). Advances in micro-electromechanical systems (MEMS) have made inertial sensors widely available in everyday life, for instance in smartphones. MEMS sensors are relatively small, cheap and have low power consumption. The accuracy of these sensors is highly dependent on a proper calibration that removes systematic errors and sensor biases. Calibration refers to the procedure of measuring some known quantity and estimating sensor parameters such that the measurement output agree with that known information. An example of the type of calibration discussed in this paper can be seen in Fig. 5.1. MEMS inertial sensors are only approximately calibrated by the

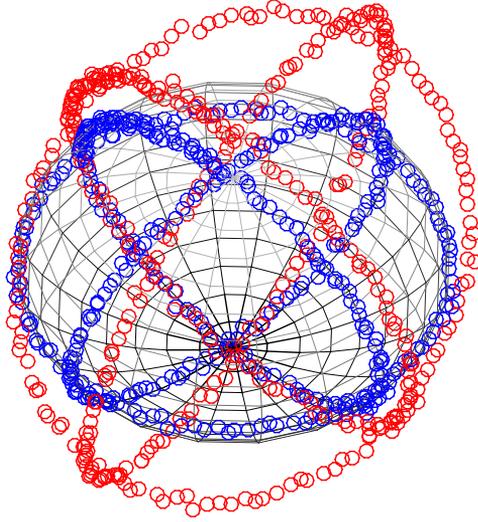


Figure 5.1: Illustration of synthetic calibrated and uncalibrated accelerometer measurements in 3D. The calibrated measurements $\mathbf{y}_t^{\text{cal}}$ (blue) are centered on a sphere with radius $\|\mathbf{g}\|_2$ centered around the origin. The uncalibrated measurements \mathbf{y}_t (red) form an ellipsoid, possibly centered around an offset.

manufacturer, and some sensor errors change over time (Woodman 2007). Therefore, in order to obtain high accuracy measurements, the sensors have to be recalibrated in the field.

Most existing methods for accelerometer calibration use measurements from a set of different static orientations to estimate a set of parameters. These methods are based on the fact that the magnitude of the measured acceleration should be equal to the local gravitational acceleration in static conditions. The choice of calibration method largely depends on the existing systematic errors. Some methods estimate three gains and three bias parameters (Grip and Sabourova 2011; Won and Golnaraghi 2010). This is sufficient if the accelerometer axes can be assumed to be perfectly orthogonal, and if the cross-axis interference caused by electric coupling in the electronics is negligible (Frosio et al. 2012). For lower quality sensors these assumptions are typically not valid, and as a result of this, up to three additional parameters have to be introduced and estimated to compensate for these errors (Forsberg et al. 2013).

The methods discussed above concern the calibration of a stand-alone triaxial accelerometer. However, if there are more sensors available in the same platform, it makes sense to use a sensor fusion approach. More specifically, accelerometers are typically available in combination with gyroscopes.

Using these sensors together, it is possible to formulate the calibration problem as a problem of estimating the sensor’s orientation in the presence of unknown calibration parameters. A similar approach is used by Kok and Schön (2014) for magnetometer calibration. They assume, however, that the accelerometers are calibrated, which may not always hold for lower quality sensors.

When calibrating sensors which are mounted in a larger sensor platform, the relative orientation between the sensors becomes of interest. To compensate for this inter-sensor misalignment, three additional parameters need to be estimated (Fang et al. 2014; Panahandeh et al. 2010), giving a total of 12 calibration parameters.

In this paper we present a calibration method for a triaxial accelerometer using a sensor fusion approach similar to Kok and Schön (2014). However, we only use information from the inertial sensors and disregard the magnetometer. Including the gyroscope measurements allows for more freedom in the way the sensors are rotated during the calibration procedure. The goal of this method is to allow for lower quality sensors to be used in applications which require measurements of a higher accuracy than these sensors initially provide.

5.2 Model and Problem formulation

To formulate the calibration problem for the accelerometer we first need to model the rotation of the sensor platform. Using the angular velocity measurements provided by the gyroscope, we formulate a dynamic model, which uses the gyroscope measurements as inputs to predict the orientation of the sensor

$$\mathbf{x}_{t+1} = \mathbf{x}_t \odot \mathbf{exp} \frac{\Delta_t}{2} \omega_t, \quad (5.1)$$

where the state variables \mathbf{x}_t are unit quaternions, \odot denotes the quaternion multiplication and \mathbf{exp} the vector exponential. For background on quaternion algebra, see e.g. Särkkä (2007). The input to the dynamic model is the angular velocity ω_t , at time t , sampled with interval Δ_t . The gyroscope measurements, $\bar{\omega}_t$, are modelled as

$$\bar{\omega}_t = \omega_t + \mathbf{b}_\omega + \mathbf{v}_t, \quad (5.2)$$

where \mathbf{b}_ω is the gyroscope bias and $\mathbf{v}_t \sim \mathcal{N}(0, \Sigma_\omega)$ is Gaussian measurement noise. Estimating and removing the bias \mathbf{b}_ω from (5.2) will allow us to use the gyroscope measurements as input to (5.1).

Next we model the accelerometer measurements, which depend on the orientation of the sensor platform and the unknown calibration parameters

to be estimated. The raw (uncalibrated) accelerometer measurements, \mathbf{y}_t , are modelled as

$$\mathbf{y}_t = \mathbf{D}\mathbf{R}(\mathbf{x}_t)\mathbf{g} + \mathbf{b}_a + \mathbf{e}_t, \quad (5.3)$$

where $\mathbf{R}(\mathbf{x}_t)$ is a rotation matrix describing the orientation of the sensor. The sensor is assumed to be in constant linear velocity in the gravitational field with acceleration \mathbf{g} , which is true for stationary sensors. The local gravitational acceleration \mathbf{g} is also assumed to be known. The sensor bias is modelled by the vector $\mathbf{b}_a \in \mathbb{R}^{3 \times 1}$ and $\mathbf{e}_t \sim \mathcal{N}(0, \Sigma_a)$ is Gaussian measurement noise. Gains, non-orthogonal sensor axes, cross-axis interference and inter-sensor misalignments are modelled by the matrix $\mathbf{D} \in \mathbb{R}^{3 \times 3}$. In fact, \mathbf{D} can be seen as a product of matrices that model these errors separately, and the number of unknown parameters that make up the elements of \mathbf{D} can be reduced if some of these errors are ignored. The desired error-free accelerometer output is $\mathbf{R}(\mathbf{x}_t)\mathbf{g}$. Therefore, if \mathbf{D} and \mathbf{b}_a are known, the calibrated accelerometer measurements can be calculated as

$$\mathbf{y}_t^{\text{cal}} = \mathbf{D}^{-1}(\mathbf{y}_t - \mathbf{b}_a). \quad (5.4)$$

In 3D space the calibrated accelerometer measurements will form a sphere of radius $\|\mathbf{g}\|_2$ with its center in the origin, while the uncalibrated measurements form an ellipsoid centered around the sensor bias \mathbf{b}_a , see Fig. 5.1.

The calibration problem is formulated as a Maximum likelihood (ML) estimation problem for the unknown model parameters $\theta = \{\mathbf{D}, \mathbf{b}_a\}$, where the ML estimator for θ is given by

$$\hat{\theta} = \arg \max_{\theta} p_{\theta}(\mathbf{y}_{1:T}) = \arg \max_{\theta} \prod_{t=1}^T p_{\theta}(\mathbf{y}_t | \mathbf{y}_{1:t-1}), \quad (5.5)$$

using all measurements from time $t = 1$ to T , and the convention that $\mathbf{y}_{1:0} = \emptyset$. The likelihood function, $p_{\theta}(\mathbf{y}_{1:T})$, is obtained from the one step ahead predictors, $p_{\theta}(\mathbf{y}_t | \mathbf{y}_{1:t-1})$. We use an extended Kalman filter (EKF) to approximate the one step ahead predictors. The EKF uses the nonlinear state space model given by (5.1) and (5.3), and estimates the orientation of the sensor platform as the unit quaternion states \mathbf{x}_t . For more details about the EKF implementation, see e.g. Kok (2014). The one step ahead predictors are then approximated as

$$p_{\theta}(\mathbf{y}_t | \mathbf{y}_{1:t-1}) \approx \mathcal{N}(\mathbf{y}_t | \hat{\mathbf{y}}_{t|t-1}(\theta), \mathbf{S}_t(\theta)), \quad (5.6)$$

which is a probability density function of a Gaussian random variable \mathbf{y}_t with mean $\hat{\mathbf{y}}_{t|t-1}(\theta)$ and covariance matrix $\mathbf{S}_t(\theta)$.

An equivalent formulation of (5.5) is

$$\hat{\theta} = \arg \min_{\theta} V(\theta), \quad (5.7)$$

where $V(\theta)$ is the negative log-likelihood, which will be referred to as the cost function. It is given by

$$V(\theta) = \frac{1}{2} \sum_{t=1}^T \|\mathbf{y}_t - \hat{\mathbf{y}}_{t|t-1}(\theta)\|_{\mathbf{S}_t^{-1}(\theta)}^2 + \log \det \mathbf{S}_t(\theta), \quad (5.8)$$

where $\|\cdot\|_{\mathbf{S}_t^{-1}(\theta)}$ is the 2-norm weighted by the inverse of the covariance matrix. From the Kalman filter theory we can recognize $\mathbf{y}_t - \hat{\mathbf{y}}_{t|t-1}(\theta)$ as the output innovations, and $\mathbf{S}_t(\theta)$ as the innovation covariance. The EKF is hence used to evaluate the cost function (5.8), which allows the optimization problem (5.7) to be solved using an iterative method, see Section 3.

5.3 Calibration algorithm

The calibration algorithm, summarized in Algorithm 1, works on sets of sequential measurements that should be divided into two subsequent segments. In the first segment, the sensors are stationary on a flat surface and in the second segment the sensors are rotated into different orientations. The first segment is used to estimate the initial orientation, \mathbf{x}_0 , of the sensors and to estimate the parameters \mathbf{b}_ω , Σ_a and Σ_ω . The noise covariance matrices Σ_a and Σ_ω are assumed to be diagonal. The gyroscope is assumed to be calibrated after this first part. The second segment is then used to solve the optimization problem in (5.7) for the parameters θ , with an iterative method. In every iteration, the cost function (5.8) is evaluated by running the orientation estimation EKF described in Section 2, using the current parameter estimates. To solve the optimization problem (5.7) we use a Gauss-Newton method (Wright and Nocedal 2006), which requires the gradients and the approximate Hessian of the cost function. However, computing the gradients of (5.8) is not straightforward, since the predicted state $\hat{\mathbf{x}}_{t|t-1}$ depends on the parameters θ through the previous measurement updates of the filter. Therefore, a numerical gradient of the cost function is calculated by approximating the partial derivatives $\frac{dV(\theta)}{d\theta}$ using finite differences. The numerical gradient is then used to approximate the Hessian as $(\frac{dV(\theta)}{d\theta})^T (\frac{dV(\theta)}{d\theta})$. Two independent stopping criteria are used for the optimization algorithm. The first criterion requires the Newton decrement (Boyd and Vandenberghe 2004) to be smaller than some constant ϵ_1 . The second criterion will stop the algorithm if the calculated step length found in the line search is smaller than some other constant ϵ_2 , and does not satisfy the

Algorithm 1 Calibration algorithm

- 1: Estimate accelerometer and gyroscope covariance, gyroscope bias and the initial orientation of the sensor, $\widehat{\Sigma}_a, \widehat{\Sigma}_\omega, \widehat{\mathbf{b}}_\omega, \widehat{\mathbf{x}}_0$ from a stationary portion of the sampled measurements.
 - 2: Initialize $\widehat{\mathbf{D}}_0 = \mathbf{I}_3$, a 3×3 identity matrix and $\mathbf{b}_{a,0} = \mathbf{0}_{3 \times 1}$, a 3×1 zero vector.
 - 3: Set $i = 0$ and repeat:
 - 4: Run the EKF using the current estimates $\widehat{\theta}_i = \{\widehat{\mathbf{D}}_i, \widehat{\mathbf{b}}_{a,i}\}$ and the initial orientation $\widehat{\mathbf{x}}_0$.
 - 5: Approximate the partial derivatives of the cost function (5.8) as finite differences and calculate the approximate Hessian.
 - 6: Calculate $\widehat{\theta}_{i+1}$ by solving (5.7) as an unconstrained optimization problem using a Gauss-Newton method and a line search algorithm (Wright and Nocedal 2006).
 - 7: Set $i = i + 1$ and repeat from step 4 until convergence.
-

Wolfe conditions (Wright and Nocedal 2006). The constants ϵ_1 and ϵ_2 can be chosen as a compromise between accuracy and computation speed.

5.4 Experimental results

The calibration algorithm was tested in two separate experiments using real sensors in two different smartphones (Samsung Galaxy S5 and Google Nexus 4). Specifically, the sensors used were the MPU-6050 for the Google Nexus 4, and the MPU-6500 for the Samsung Galaxy S5, both sensors are manufactured by InvenSense (InvenSense 2017). These sensors were of interest as we have previously seen distinct variations in the magnitudes of the accelerometer measurements for different orientations in stationary conditions, when using these sensors. The method was also verified using synthetic data.

5.4.1 Real sensor data

In the first experiment, sensor data was collected from a Samsung Galaxy S5 smartphone. The smartphone was initially at rest on a flat surface for approximately 10 seconds. Subsequently, the phone was picked up and rotated by hand. Two different datasets were collected using different types of rotation. For the first dataset, the smartphone was rotated into 6 different pre-specified orientations, separated by approximately 90 degrees. The smartphone was stationary in these orientations for periods of approximately 10 seconds before it was rotated again. For the second dataset, the

smartphone was continuously rotated around all of its axes. The datasets were used to calibrate the sensors in the phone using Algorithm 1.

The results for the two datasets are seen in Fig. 5.2. After calibrating the accelerometer the measurement magnitudes become centered, close to $\|\mathbf{g}\|_2 = 9.82\text{m s}^{-2}$. The average magnitude of the calibrated data was $9.81(\sigma = 0.23)\text{m s}^{-2}$ for the first dataset, and $9.80(\sigma = 0.30)\text{m s}^{-2}$ for the second dataset, with standard deviations σ . This is a significant improvement over the uncalibrated measurements, where the magnitude varies for different orientations.

Every time the phone is rotated the accelerometer data violates the measurement model, which assumes that the accelerometer is only affected by the gravitational acceleration. The second dataset therefore contains more outliers than the first dataset. The effect of having outliers in the calibration data is not examined further in this paper. However, it could result in improper calibrations if the number of outliers becomes very large.

5.4.2 Orientation estimation

In the second experiment a Google Nexus 4 smartphone was used to test the calibration performance. The phone was attached to a board which was stationary at first, followed by one complete rotation around the roll, pitch and yaw axes. The rotation speed was in the order of 9°s^{-1} . This was repeated three times. Reflective markers, that were detected and tracked by an optical reference system (Vicon), were attached to the board. Using multiple markers, the orientation of the board was obtained with high accuracy and used as a reference. The measurements collected from the accelerometers and gyroscopes of the phone were used to estimate the orientation of the board using an EKF. The estimated orientations were then compared with the reference measurements. Data from the first stationary part and one rotation segment was used to calibrate the sensors, while the orientation estimation was performed on the whole dataset.

The orientations of interest here are the roll and pitch angles, which are observable by the accelerometer. The yaw angle, or heading, can only be observed by introducing additional sensors, for example a magnetometer. When using the uncalibrated accelerometer the average absolute error of the estimated roll and pitch angles was 3.43° and 1.61° , respectively. The errors were reduced to 1.82° and 0.78° , respectively, when using the calibrated accelerometer measurements.

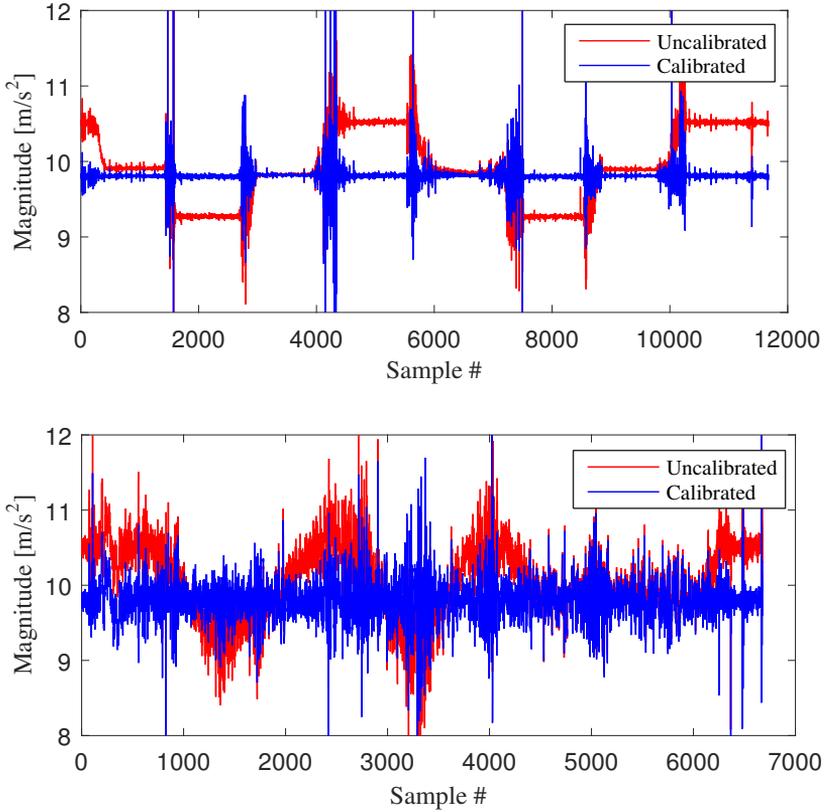


Figure 5.2: Measurement magnitudes of the calibrated and uncalibrated accelerometer in the Samsung Galaxy S5, from the two datasets explained in Section 5.4.1. Top: The phone was rotated into 6 different fixed orientations. Bottom: The phone was continuously rotated by hand.

5.4.3 Synthetic data

Monte Carlo simulations were performed to verify the functionality of the calibration algorithm. Sensors with randomly generated parameters were calibrated using synthetic data. The synthetic accelerometers had a gain in the range $(0.5, 1.5)$, a inter-sensor misalignment in the range of $\pm 10^\circ$, non-orthogonal sensor axes in the range of $\pm 30^\circ$ and a bias in the range of $\pm 1 \text{ m s}^{-2}$. The synthetic gyroscopes had biases in the range of $\pm 1 \text{ rad s}^{-1}$. The diagonal elements of the noise covariance matrices were in the range of $(10^{-3}, 10^{-1})$ and $(10^{-3}, 10^{-2})$ for the synthetic accelerometers and gyroscopes, respectively. All of the mentioned ranges have a uniform distribution. The synthetic data represented one complete rotation around each

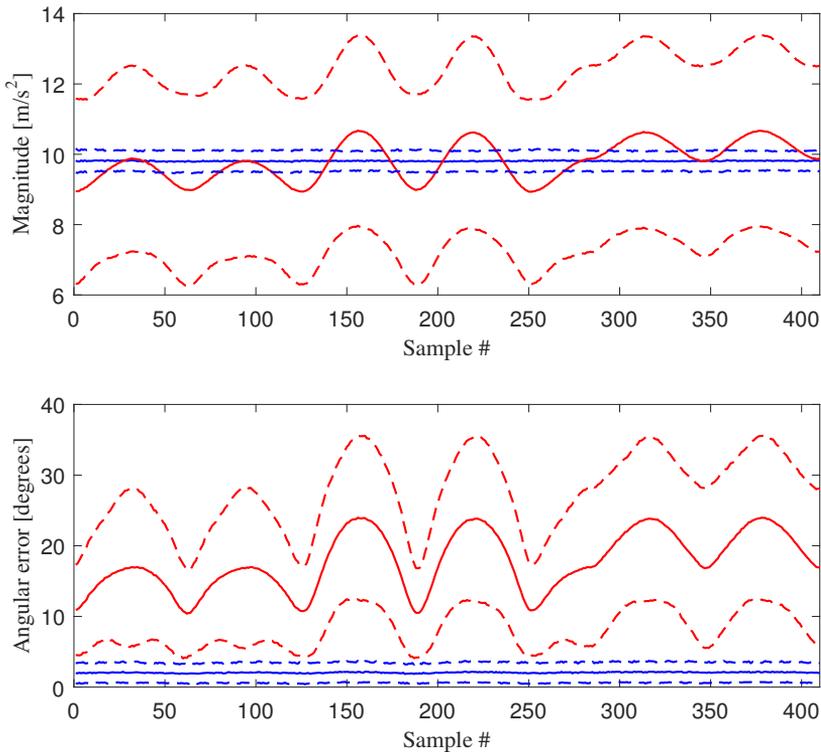


Figure 5.3: Calibrated and uncalibrated measurement magnitudes and angular errors for 1000 Monte Carlo simulations. The blue lines represent the calibrated measurements $\mathbf{y}_t^{\text{cal}}$, and the red lines represent the uncalibrated measurements \mathbf{y}_t . The means are represented as solid lines with a confidence interval of one standard deviation, represented by the dashed lines.

axis. A total of 1000 simulations were performed.

The difference between the calibrated and uncalibrated measurements can be seen in Fig. 5.3. After calibration, the magnitude of the measurements are significantly closer to the true magnitude $\|\mathbf{g}\|_2 = 9.82\text{m s}^{-2}$. The absolute value of the angle between the calibrated measurement vectors $\mathbf{y}_t^{\text{cal}}$, and the ideal (noise- and error free) measurement vectors (the angular error), also decreased to an average of 2 degrees. The average absolute orientation estimation errors also decreased significantly from $8.3(\sigma = 7.1)^\circ$ to $0.32(\sigma = 0.40)^\circ$ for the roll angle, and from $8.3(\sigma = 6.9)^\circ$ to $0.33(\sigma = 0.43)^\circ$ for the pitch angle. To speed up these simulations, the Gauss-Newton method was allowed to perform a maximum of 10 iterations. However, the algorithm converged in 6 iterations or less on average.

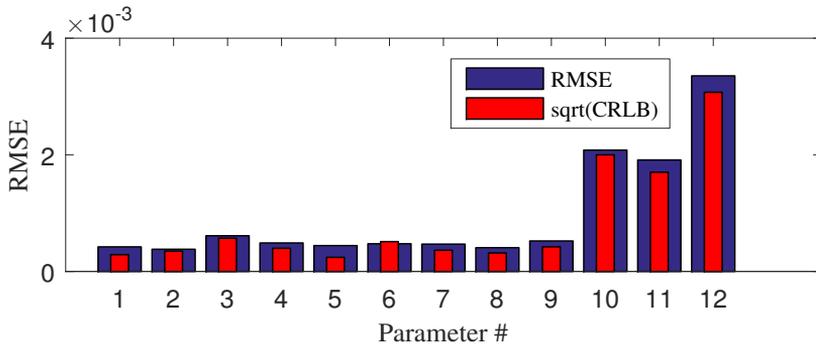


Figure 5.4: Square root of the estimated CRLB and the RMSE for the estimated parameters for 100 simulations using the same set of parameters. The first 9 parameters belong to the vectorized matrix \mathbf{D} and parameters 10-12 belong to the bias vector \mathbf{b}_a .

5.4.4 Cramér-Rao lower bound

A common tool used to evaluate the performance of estimators is the Cramér-Rao lower bound (CRLB), which gives a lower bound on the variance of an unbiased estimator (Gustafsson 2010). The CRLB was calculated using the mean Hessian from 100 simulations using one fixed set of parameters. The square root of the CRLB can be compared with the root-mean-square error (RMSE), for the estimated parameters $\hat{\theta}$, see Fig. 5.4. The RMSE values were found to be close to the square root of the estimated CRLB which suggests that the estimator is efficient. Interestingly, for parameter 6, it can be seen that $\sqrt{\text{CRLB}} > \text{RMSE}$. This could be because of the numerical approximation of the CRLB or because there is a small bias in the parameter estimates.

5.5 Conclusions

In this paper, a calibration method for a triaxial accelerometer, using sensor fusion with a triaxial gyroscope in the same platform, has been presented. The main goal of the method is to allow sensor platforms with lower quality sensors, e.g. smartphones, to be used in applications which require higher accuracy measurements. The method has been tested using both real and synthetic data, and has been shown to accurately calibrate the sensors. The estimated parameters can be used to compensate for gain, inter-sensor misalignment, non-orthogonal sensor axes, cross-axis interference and bias. The method was used to calibrate the accelerometer in a smartphone, which brought the measurement magnitudes closer to $\|\mathbf{g}\|_2$. This was a signific-

ant improvement compared to the uncalibrated accelerometer, where the magnitudes varied for different orientations. It was also shown that the calibration works when using a smartphone to perform orientation estimation, improving the accuracy in the estimated roll and pitch angles. Monte Carlo simulations show a significantly reduced effect of sensor errors in the calibrated measurements. Simulations also showed that the RMSE was close to the $\sqrt{\text{CRLB}}$ for the estimated parameters, which suggests that the Maximum likelihood estimator is efficient. The Gaussian noise model used here cannot describe measurement outliers.

We see a number of interesting improvements and extensions of the method. The problem of too many outliers could be handled by different approaches: outlier detection, by using a heavytailed noise distribution or by using robust Kalman filtering (Mattingley and Boyd 2010). Another possible future extension would be to combine the method with magnetometer calibration for more accurate heading estimates. In addition to these improvements, further analysis with sensors of varying quality, and comparisons to other methods would be interesting as future work.

References

- Boyd, S. and L. Vandenberghe (2004). *Convex optimization*. Cambridge university press.
- Fang, B., W. Chou and L. Ding (2014). ‘An Optimal Calibration Method for a MEMS Inertial Measurement Unit’. In: *International Journal of Advanced Robotic Systems* 11, pp. 1–14.
- Forsberg, T., N. Grip and N. Sabourova (2013). ‘Non-iterative calibration for accelerometers with three non-orthogonal axes, reliable measurement setups and simple supplementary equipment’. In: *Measurement Science and Technology* 24.3, p. 035002.
- Frosio, I., F. Pedersini and N. A. Borghese (June 2012). ‘Autocalibration of Triaxial MEMS Accelerometer With Automatic Sensor Model Selection’. In: *IEEE Sensors Journal* 12.6, pp. 2100–2108.
- Grip, N. and N. Sabourova (2011). ‘Simple non-iterative calibration for triaxial accelerometers’. In: *Measurement Science and Technology* 22.12, p. 125103.
- Gustafsson, F. (2010). *Statistical sensor fusion*. Studentlitteratur.
- InvenSense (Sept. 2017). *6 Axis MotionTracking*. URL: <http://www.invensense.com/products/motion-tracking/6-axis/>.
- Kok, M. (2014). *Probabilistic modeling for positioning applications using inertial sensors*. Licentiate’s thesis, Linköping University, Sweden.
- Kok, M. and T. B. Schön (Aug. 2014). ‘Maximum likelihood calibration of a magnetometer using inertial sensors’. In: *Proceedings of the 19th World Congress of the International Federation of Automatic Control (IFAC)*. Cape Town, South Africa, pp. 92–97.
- Mattingley, J. and S. Boyd (May 2010). ‘Real-Time Convex Optimization in Signal Processing’. In: *IEEE Signal Processing Magazine* 27.3, pp. 50–61.
- Olsson, F., M. Kok, K. Halvorsen and T. B. Schön (2016). ‘Accelerometer calibration using sensor fusion with a gyroscope’. In: *Statistical Signal Processing Workshop (SSP), 2016 IEEE*. (Palma de Mallorca, Spain). IEEE, pp. 1–5.
- Panahandeh, G., I. Skog and M. Jansson (Sept. 2010). ‘Calibration of the Accelerometer Triad of an Inertial Measurement Unit, Maximum Likelihood Estimation and Cramér-Rao Bound’. In: *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. Zurich, Switzerland, pp. 1–6.
- Särkkä, S. (2007). *Notes on quaternions*. Tech. rep. Helsinki University of Technology.
- Serrano, D. E. and F. Ayazi (2015). ‘Resonant MEMS: Fundamentals, Implementation, and Application’. In: ed. by O. Brand, I. Dufour, S. Heinrich,

- F. Josse, G. K. Fedder, C. Hierold, J. G. Korvink and O. Tabata. Wiley-VCH. Chap. MEMS Inertial Sensors, pp. 329–352.
- Won, S. P. and F. Golnaraghi (Aug. 2010). ‘A Triaxial accelerometer Calibration Method Using a Mathematical Model’. In: *IEEE Transactions on Instrumentation and Measurement* 59.8, pp. 2144–2153.
- Woodman, O. J. (Aug. 2007). *An introduction to inertial navigation*. Tech. rep. 696. University of Cambridge Computer Laboratory.
- Wright, S. and J. Nocedal (2006). *Numerical Optimization*. 2nd. Springer Series in Operations Research.

Title

Experimental evaluation of joint position estimation using inertial sensors

Authors

Fredrik Olsson and Kjartan Halvorsen

Edited version of

F. Olsson and K. Halvorsen (2017). 'Experimental evaluation of joint position estimation using inertial sensors'. In: *Information Fusion (Fusion), 2017 20th International Conference on*. (Xi'an, China). IEEE, pp. 1–8

Experimental evaluation of joint position estimation using inertial sensors

Abstract

In this paper we study the problem of estimating the position of a joint that is connecting two rigid links in a biomechanical model. By equipping the two links with inertial sensors, which measure linear acceleration and angular velocity, it is possible to estimate the joint position. Estimation methods for this problem have been proposed before, but experimental evaluation and comparison between methods is lacking. The main contribution of this paper is an experimental evaluation of three different methods, a least-squares method and two iterative optimization methods that minimize a sum of squared errors and a sum of absolute errors. The iterative methods turns out to be superior to the least-squares method. Minimizing the sum of squared errors is faster but not as robust against outliers as minimizing the sum of absolute errors.

6.1 Introduction

Inertial sensors, specifically triaxial accelerometers and gyroscopes which measure linear acceleration and angular velocity in three dimensions, are widely available in our everyday lives. As microelectromechanical systems (MEMS), inertial sensors are staple components in today's smart phones and other wearable devices such as activity monitors, smart wristbands and watches. This has made inertial sensors attractive tools for analysing human motion, for example to perform full body motion capture (Roetenberg et al. 2013), quantifying symptoms of diseases, tracking the progress of rehabilitation (Patel et al. 2012) or aiding elite sportsmen in their training (James 2006). In all of these applications, an important problem to consider is

how the measurements from the sensors relate to the human body. A human body can be approximated as separate links that connect to each other through joints that allow the links to move in certain ways with respect to each other. Such a model of a human is what we refer to as a biomechanical model. Establishing the relationship between the inertial sensors and links and joints of the biomechanical model is referred to as calibrating the biomechanical model, and it is often a necessary step in the mentioned applications.

In this paper we consider the problem of estimating the position of a joint connecting two links in the biomechanical model. One inertial sensor will be attached to each link. We assume the positions of the sensors to be fixed with respect to the center of the joint, but apart from this there will be no further prior knowledge assumed about the positions and orientations of the sensors. When the position of the joint center is fixed with respect to the reference frame of the sensors, it is possible to estimate the position of the joint using the measurements from the two inertial sensors. Similar problems have been studied previously in e.g. Crabolu et al. (2016), McGinnis and Perkins (2013) and Seel et al. (2012), but experimental evaluation and comparisons of the methods are lacking. Our main contribution with this paper is the experimental evaluation of three possible methods for estimating the joint position using the two inertial sensors. Evaluations were made using a rather simple experimental system, an artificial limb consisting of two rigid links connected by a spherical joint.

6.2 Modeling

6.2.1 Relating the inertial measurements to the biomechanical model

Let L_1 and L_2 denote local (Cartesian) reference frames that are fixed in sensor 1 and 2, respectively, and G denote a global earth-fixed reference frame. The measured accelerations a relate to the measured angular velocities ω and the angular accelerations $\dot{\omega}$ in the following way

$$a_1 = a_0^{L_1} + \omega_1 \times (\omega_1 \times r_1) + \dot{\omega}_1 \times r_1 \quad (6.1)$$

$$a_2 = a_0^{L_2} + \omega_2 \times (\omega_2 \times r_2) + \dot{\omega}_2 \times r_2, \quad (6.2)$$

where a_0 is the linear acceleration of the center of rotation expressed in L_1 and L_2 , respectively, as indicated by the superscripts. The measured accelerations, angular velocities and angular accelerations, are expressed in the local reference frame associated with the subscripts. A similar notation is used for the position of the joint, r , when expressed in the local reference

frames. The described model is illustrated in Figure 6.1. We can rewrite (6.1)-(6.2) as

$$a_1 = a_0^{L1} + K(\omega_1, \dot{\omega}_1)r_1 \quad (6.3)$$

$$a_2 = a_0^{L2} + K(\omega_2, \dot{\omega}_2)r_2, \quad (6.4)$$

where

$$K(\omega, \dot{\omega}) = \begin{bmatrix} -\omega_y^2 - \omega_z^2 & \omega_x\omega_y - \dot{\omega}_z & \omega_x\omega_z + \dot{\omega}_y \\ \omega_x\omega_y + \dot{\omega}_z & -\omega_x^2 - \omega_z^2 & \omega_y\omega_z - \dot{\omega}_x \\ \omega_x\omega_z - \dot{\omega}_y & \omega_y\omega_z + \dot{\omega}_x & -\omega_x^2 - \omega_y^2 \end{bmatrix}, \quad (6.5)$$

where the x, y, z subscripts denote the coordinates of the three dimensional vectors. To simplify notation we will use $K(\omega_1, \dot{\omega}_1) = K_1$ and $K(\omega_2, \dot{\omega}_2) = K_2$.

6.2.2 Measurement models

The accelerometers and gyroscopes provide us with noisy measurements of the accelerations and the angular velocities. Simple models of the measurements at sample instant k are given by

$$y_a(k) = a(k) + e_{a,k}, \quad e_{a,k} \sim \mathcal{N}(0, \Sigma_a) \quad (6.6)$$

$$y_\omega(k) = \omega(k) + e_{\omega,k}, \quad e_{\omega,k} \sim \mathcal{N}(0, \Sigma_\omega), \quad (6.7)$$

where we say that the measurement is equal to the physical quantity plus some additive Gaussian measurement noise. Inertial sensors, especially lower quality sensors, tend to also have a time varying bias, causing the measurements to slowly drift over time. A simple way to estimate the bias is to collect data from stationary sensors and see how much they differ from the expected stationary accelerations and angular velocities. This initially estimated bias can then be assumed to be stationary over short time periods and removed from the data. Angular accelerations are also needed by the methods, and these must be estimated from the gyroscope measurements. We will use a simple second-order symmetric difference quotient

$$\dot{\omega}(k) \approx y_{\dot{\omega}}(k) = \frac{y_\omega(k+1) - y_\omega(k-1)}{2h}, \quad (6.8)$$

where h is the sample period. Other, higher order differentiation techniques may also be used. When estimating the position of the joint we will make use of these noisy measurements, so when we refer to a , ω and $\dot{\omega}$ in the upcoming sections we will mean the measurements given by (6.6)-(6.8) and not the true physical quantities.

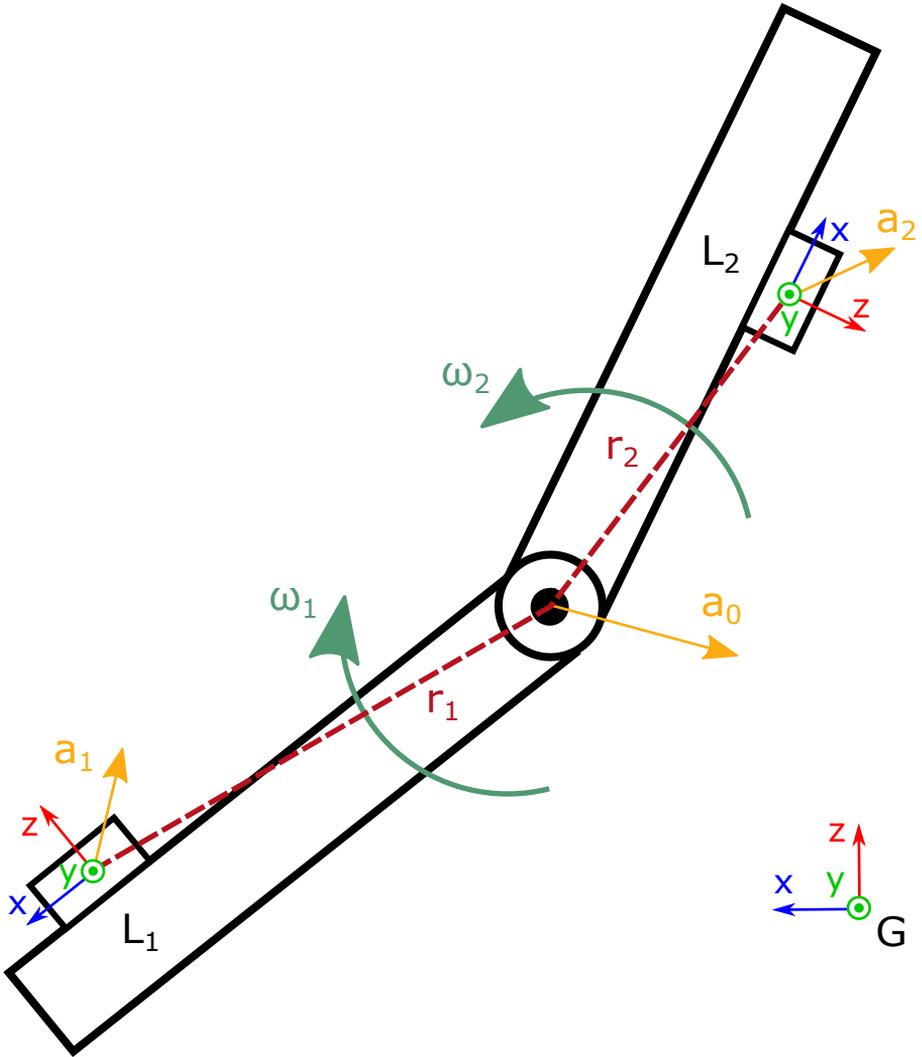


Figure 6.1: The biomechanical model and inertial sensors, where the joint positions with respect to the sensors, r_1 and r_2 are to be estimated.

6.3 Estimation

6.3.1 Least-squares estimate

Note that (6.3)-(6.4) is linear in a_0 and $r = \begin{bmatrix} r_1 & r_2 \end{bmatrix}^\top$. Let $R_{1,2}$ denote the rotation matrices which rotates vectors expressed in the $L_{1,2}$ frames to the G frame. For example we have that

$$R_1 a_0^{L_1} = a_0^G.$$

By using these rotation matrices we can express (6.3)-(6.4) in the G frame

$$\begin{aligned} R_1 a_1 - R_2 a_2 &= a_0^G - a_0^G + R_1 K_1 r_1 - R_2 K_2 r_2 \\ &= R_1 K_1 r_1 - R_2 K_2 r_2, \end{aligned} \quad (6.9)$$

which is a system of equations which is linear in r . Once both accelerations are expressed in a common reference frame, a_0 is eliminated from the equation by subtracting one acceleration from the other. With a set of N available measurements, the following system of equations can be constructed

$$\tilde{K} r = \tilde{a} \quad (6.10)$$

$$\tilde{K} = \begin{bmatrix} \tilde{K}(1) \\ \vdots \\ \tilde{K}(N) \end{bmatrix} = \begin{bmatrix} R_1(1)K_1(1) & -R_2(1)K_2(1) \\ \vdots & \vdots \\ R_1(N)K_1(N) & -R_2(N)K_2(N) \end{bmatrix} \quad (6.11)$$

$$\tilde{a} = \begin{bmatrix} \tilde{a}(1) \\ \vdots \\ \tilde{a}(N) \end{bmatrix} = \begin{bmatrix} R_1(1)a_1(1) - R_2(1)a_2(1) \\ \vdots \\ R_1(N)a_1(N) - R_2(N)a_2(N) \end{bmatrix}, \quad (6.12)$$

where $\tilde{K} \in \mathbb{R}^{3N \times 6}$ and $\tilde{a} \in \mathbb{R}^{3N \times 1}$. A least-squares (LS) estimate of r can be found by minimizing a cost function which corresponds to the sum of squared errors of the linear equations

$$\hat{r}_{LS} = \arg \min_r \sum_{k=1}^N \left\| \tilde{K}(k)r - \tilde{a}(k) \right\|_2^2, \quad (6.13)$$

which has the closed form solution

$$\hat{r}_{LS} = \tilde{K}^\dagger \tilde{a}, \quad (6.14)$$

where \dagger denotes the Moore-Penrose pseudo inverse.

The rotation matrices $R_{1,2}$ can be found by estimating the orientation of the inertial sensors with respect to G . There are several methods that one can use to estimate the orientations of the inertial sensors (Kok 2016), and some inertial sensors include internally computed orientation estimates. However, since estimates of $R_{1,2}$ are needed to compute \hat{r}_{LS} this introduces an additional source of uncertainty that can be hard to quantify. It should be noted that the LS approach presented here extends the methods presented in McGinnis and Perkins (2013) and Crabolu et al. (2016). These methods assume that $a_0 = 0$, which would allow for separate estimation of r_1 and r_2 without having to know the orientation of the sensors. There are not many scenarios where the assumption that the center of the joint is at rest holds when studying human motion, however. So it is less restrictive and more practical to consider the case when $a_0 \neq 0$.

6.3.2 Estimation using iterative optimization methods

Another way of estimating r is to use iterative optimization methods. The method proposed in Seel et al. (2012) minimizes a sum of squared errors (SS) cost function

$$\hat{r}_{SS} = \arg \min_r \sum_{k=1}^N e(k, r)^2 \quad (6.15)$$

$$e(k, r) = \|a_1(k) - K_1(k)r_1\|_2 - \|a_2(k) - K_2(k)r_2\|_2, \quad (6.16)$$

where $\|\cdot\|_2$ stands for the Euclidean norm. A nice property of defining the errors $e(k, r)$ as the difference between two Euclidean norms is that the Euclidean norm is invariant to rotations. Therefore, unlike the LS method, we do not need to estimate the orientation of the sensors to remove a_0 from the equation since $\|a_0^{L_1}\|_2 = \|a_0^{L_2}\|_2$. The optimization problem (6.15) can be solved efficiently using a Gauss-Newton method (Wright and Nocedal 2006) as shown in Seel et al. (2012), which makes use of the partial derivatives

$$\frac{\partial e(k, r)}{\partial r_1} = -\frac{K_1(k)^\top e_1(k, r_1)}{\|e_1(k, r_1)\|_2} \quad (6.17)$$

$$\frac{\partial e(k, r)}{\partial r_2} = \frac{K_2(k)^\top e_2(k, r_2)}{\|e_2(k, r_2)\|_2}, \quad (6.18)$$

where

$$e_i(k, r_i) = a_i(k) - K_i(k)r_i, \quad i = 1, 2. \quad (6.19)$$

An alternative to minimizing the sum of squared errors is to instead minimize the sum of absolute errors (SA)

$$\hat{r}_{SA} = \arg \min_r \sum_{k=1}^N |e(k, r)|. \quad (6.20)$$

A well known property of this type of cost function is robustness against outliers in the data since large residuals are weighted by their absolute value, not by their squared value in the cost function. By outliers we mean measurements that contain some sort of disturbance that the model does not take into account. One common disturbance comes from the fact that human body segments are non-rigid. The sensors are placed on soft tissue, sometimes with clothing between the sensors and the skin. This violates the assumption that the sensor position with respect to the joint is fixed at all times since the surface on which the sensors are placed can move to some degree. These types of disturbances are known as soft tissue artifacts (STA)

(Leardini et al. 2005). Since the optimization problem given by (6.20) is not a sum of squares minimization problem, it is not possible to use the Gauss-Newton method to solve this optimization problem. Other gradient descent-type methods (Boyd and Vandenberghe 2004) may be used, which require the partial derivatives (subgradients)

$$\frac{\partial |e(k, r)|}{\partial r_1} = -\text{sgn}(e(k, r)) \frac{K_1(k)^\top e_1(k, r_1)}{\|e_1(k, r_1)\|_2} \quad (6.21)$$

$$\frac{\partial |e(k, r)|}{\partial r_2} = \text{sgn}(e(k, r)) \frac{K_2(k)^\top e_2(k, r_2)}{\|e_2(k, r_2)\|_2}, \quad (6.22)$$

where

$$\text{sgn}(x) = \begin{cases} -1 & x < 0 \\ 0 & x = 0 \\ 1 & x > 0 \end{cases}. \quad (6.23)$$

It is important to note that neither of the estimation methods described in this section guarantee that a unique solution exists in general. If the data contains unmodeled disturbances or outliers, multiple solutions or local minima may exist. The resulting estimates may then in some cases describe the outliers rather than the kinematics of the model of interest.

6.4 Experiment

An experiment was performed using a system consisting of two rigid cylindrical links joined together by a spherical joint, which allows the links to rotate with up to two degrees of freedom with respect to each other. Two Xsens MTw Awinda wireless inertial sensors (Xsens 2017) were placed near the ends of the links. The setup of the experimental system can be seen in Figure 6.2. The links are rigid such that the position of the joint relative to the sensors can be considered fixed for the full duration of the experiment. The positions of the joint relative to the sensors were measured beforehand to be

$$r_1 = \begin{bmatrix} 334 & 0 & 28 \end{bmatrix}^\top (\pm 2) \quad (6.24)$$

$$r_2 = \begin{bmatrix} -325 & 0 & 28 \end{bmatrix}^\top (\pm 2), \quad (6.25)$$

expressed in the $L_{1,2}$ frames in millimeters. The experiment consisted of collecting inertial measurements while picking up the system and performing certain motions by hand. The sample rates of the sensors were 100 Hz and a total of 15219 data samples were collected during the experiment. The



Figure 6.2: The system that was used in the experiment with two Xsens MTw inertial sensors attached.

data can be divided into segments $S_i, i = 1, 2, \dots, 8$, each of which contains one specific type of motion, see Figure 6.3. The motions contained in each segment are:

- S_1 : The system was at rest on a flat surface. This segment was used to calibrate gyroscope bias and noise covariance. Contains 1354 samples.
- S_2 : The system was picked up and held such that the two segments were in an approximately vertical orientation. Contains 1132 samples.
- S_3 : The links were moved back and forth with respect to the joint. The majority of the rotation aligned with the y -axes of the sensors. Contains 1625 samples.
- S_4 : The links were moved back and forth with respect to the joint. The majority of the rotation aligned with the z -axes of the sensors. Contains 1372 samples.
- S_5 : The links were moved back and forth with respect to the joint. Alternated rotations around the y - and z -axes of the sensors. Contains 2213 samples.
- S_6 : The links were spun such that the bottom link followed an approximately clockwise circular trajectory. The system was flipped upside down in the middle of the segment and the motion repeated. Contains 2666 samples.
- S_7 : Same as S_6 but counter-clockwise motion. Contains 2698 samples.
- S_8 : The system was placed back on the flat surface as in S_1 . Contains 2159 samples.

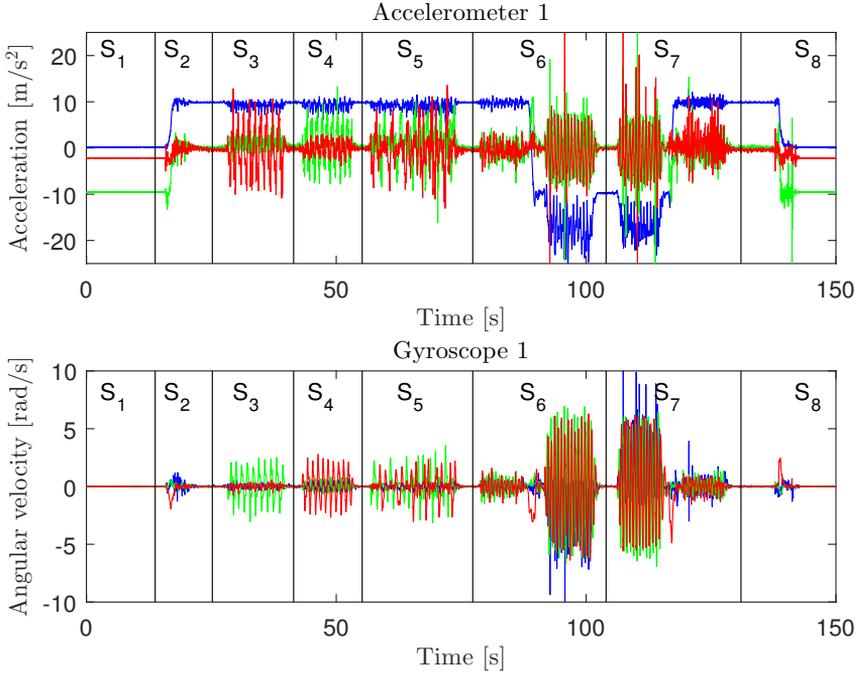


Figure 6.3: Data collected from sensor 1 during the experiment.

6.5 Results

6.5.1 Metrics for evaluation of estimators

The performance of the estimators are evaluated in terms of the average Euclidean error and its standard deviation

$$E_{\text{avg}}(\hat{r}_i) = \frac{1}{M} \sum_{j=1}^M E_j(\hat{r}_i), \quad i = 1, 2 \quad (6.26)$$

$$\text{Std}[E_j(\hat{r}_i)] = \sqrt{\frac{1}{M} \sum_{j=1}^M (E_j(\hat{r}_i) - E_{\text{avg}}(\hat{r}_i))^2} \quad (6.27)$$

$$E_j(\hat{r}_i) = \|\hat{r}_i(j) - r_i\|_2, \quad (6.28)$$

where $E_j(\hat{r}_i)$ denotes the Euclidean norm of the error for the j :th estimate of r_i . The average Euclidean error $E_{\text{avg}}(\hat{r}_i)$ and its standard deviation $\text{Std}[E_j(\hat{r}_i)]$ are computed for a total of M estimates. The rationale for choosing the average Euclidean error over e.g. the more commonly used root mean square error (RMSE) is that it may be interpreted as the average physical distance between estimates with respect to the true joint center (Li

and Zhao 2006). This results in a more intuitive understanding as to how close the estimators are to estimating the true position of the joint center. The standard deviation of the Euclidean error give us information about the variation or consistency of the estimates.

6.5.2 Method verification using simulated data with ideal excitation

To see that the methods were correctly implemented and working we first used simulated measurements for estimating $r_{1,2}$. Angular velocities and angular accelerations were randomly generated from zero-mean Gaussian distributions with standard deviations of 10 rad/s and 10 rad/s², respectively. Accelerometer measurements were then generated from the models (6.1) and (6.2) using the measured sensor positions (6.24) and (6.25). The accelerations of the joint a_0 were uniformly random rotations of the gravitational acceleration vector $g = [0 \ 0 \ 9.82]^\top$. Additive zero-mean Gaussian noise with variances $10^{-2} \text{ m}^2/\text{s}^4$, $10^{-5} \text{ rad}^2/\text{s}^2$ and $5 \times 10^{-2} \text{ rad}^2/\text{s}^4$, were added to the accelerations, angular velocities and angular accelerations, respectively. The first two noise levels are typical for the accelerometers and gyroscopes found in the Xsens MTw sensors, and the noise added to the angular accelerations takes into account the second order finite difference quotient that is used to differentiate the noisy gyroscope signals. In addition to measurement noise, orientation errors were simulated as zero-mean Gaussian random variables with a standard deviation of 5°. By allowing the angular velocities and angular accelerations to be randomly generated like this, we do not take constraints of the real biomechanical model into account, such as the physically possible range of motion. The randomly generated measurements give us an indication of how well the methods would perform if the system could be ideally excited, with no constraints limiting the motion. A total of $M = 100$ estimation runs were performed using $N = 500$ randomly generated measurements in each run. The results for these runs are shown in Table 6.1.

6.5.3 Estimation using the experimental data

The joint positions $r_{1,2}$ were estimated using the data collected from the experiment described in Section 6.4. At first, estimation was done using data from segments S_3 to S_7 . To avoid using non-informative stationary data, a threshold of 1.2 rad/s was used to exclude all samples where the norm of the gyroscope measurement was below this threshold. This threshold was also used in all subsequent estimations. As a result, a total of $N = 6280$ measurements were selected from the segments S_3 to S_7 . Using the same

Table 6.1: Shows E_{avg} [mm] and $\text{Std}[E_j]$ [mm] over $M = 100$ runs with $N = 500$ simulated measurements.

Data	Method	\hat{r}_1		\hat{r}_2	
		E_{avg}	$\text{Std}[E_j]$	E_{avg}	$\text{Std}[E_j]$
Simulated	LS	0.22	0.09	0.23	0.10
	SS	0.07	0.03	0.07	0.03
	SA	0.08	0.03	0.09	0.04

Table 6.2: Shows E_{avg} [mm] and $\text{Std}[E_j]$ [mm] over $M = 100$ runs with $N = 6280$ samples from segments S_3 - S_7 .

Data	Method	\hat{r}_1		\hat{r}_2	
		E_{avg}	$\text{Std}[E_j]$	E_{avg}	$\text{Std}[E_j]$
S_3 - S_7	LS	16.8	0.0	11.3	0.0
	SS	13.8	0.0	8.9	0.0
	SA	13.0	0.0	7.1	0.0

set of measurements, $M = 100$ estimation runs were performed with the same initial conditions for the two iterative estimates \hat{r}_{SS} and \hat{r}_{SA} . Initial conditions were randomly generated for each estimation run from a standard zero-mean Gaussian distribution with unit-variance. The rotation matrices $R_{1,2}$ required for the LS estimate, \hat{r}_{LS} were estimated using an extended Kalman filter (EKF) with orientation deviation states (Kok 2016). The global reference frame G was the geodetic reference frame with the x -axis pointing north, the y -axis pointing west and the z -axis pointing up. The results for these runs are shown in Table 6.2.

To see how the accuracy of the estimates may vary for different types of motion, the joint positions were estimated using samples from each segment separately. From each segment $N = 500$ measurements were randomly chosen for each estimation run for a total of $M = 100$ runs per segment. For segments S_6 and S_7 , 250 measurements were randomly chosen from the first and the second half of the segments, respectively, since the sensors were flipped upside down in the middle of the segment. This was to make sure we always got measurements of the rotation of both sensors. The results for these runs are shown in Table 6.3.

Table 6.3: Shows E_{avg} [mm] and $\text{Std}[E_j]$ [mm] over $M = 100$ runs with $N = 500$ randomly picked samples from different segments.

Data	Method	\hat{r}_1		\hat{r}_2	
		E_{avg}	$\text{Std}[E_j]$	E_{avg}	$\text{Std}[E_j]$
S_3	LS	332.7	19.4	181.8	18.3
	SS	17.5	0.5	6.6	0.4
	SA	17.5	2.3	6.8	1.6
S_4	LS	806.5	13.4	655.5	11.5
	SS	15.9	0.4	5.1	0.3
	SA	15.0	1.3	6.0	0.7
S_5	LS	325.6	23.4	129.3	27.5
	SS	13.8	0.5	6.8	0.5
	SA	13.0	1.3	7.3	1.0
S_6	LS	40.0	6.4	33.9	10.5
	SS	24.2	23.6	10.7	8.8
	SA	13.1	0.2	7.4	0.3
S_7	LS	20.9	8.1	21.1	7.1
	SS	22.4	12.7	9.4	2.4
	SA	13.0	0.3	7.1	0.3

6.5.4 Estimation with simulated soft tissue artifacts

In this experiment we investigated how the methods performed when the inertial measurements were corrupted by disturbances. Possible disturbances when analysing real human motion may appear in the form of STAs. Since the links in the experimental system are rigid we simulated these disturbances. A model that has been shown to emulate STAs in the thigh reasonably well expresses the displacement of markers attached to the skin as a linear combination of the hip joint angles (Camomilla et al. 2013). Differentiating this model twice w.r.t. time yields a linear relationship between the linear acceleration of the displacement of skin and the angular acceleration. Using this model structure as a basis, we generated STAs as an additive disturbance to the acceleration measurements as a linear combination of the angular acceleration measurements

$$d_a = H\dot{\omega}, \quad (6.29)$$

Table 6.4: Shows E_{avg} [mm] and $\text{Std}[E_j]$ [mm] over $M = 100$ runs with $N = 6280$ samples from segments S_3 - S_7 . With simulated disturbances added to the accelerometer measurements.

Data	Method	\hat{r}_1		\hat{r}_2	
		E_{avg}	$\text{Std}[E_j]$	E_{avg}	$\text{Std}[E_j]$
S_3 - S_7	LS	31.6	12.2	22.7	12.4
	SS	17.6	5.6	11.2	4.2
	SA	15.2	3.7	10.4	4.0

where H is a 3×3 matrix containing the STA model parameters. In practice, the elements of H depend on multiple factors such as sensor placement and the physiology of the person. Therefore, the elements of H were randomized from a zero-mean Gaussian distribution with a standard deviation of $\frac{18}{\pi}$ mm/rad at the start of each estimation run. This generated H matrices with elements of a magnitude comparable to what was found in (Camomilla et al. 2013). A total of $M = 100$ estimation runs were performed using the same $N = 6280$ samples from segments S_3 to S_7 that were used to produce the results shown in Table 6.2. The results for these runs are shown in Table 6.4.

6.5.5 Analysis of the cost functions of the iterative method

Throughout the estimation runs we found that the iterative methods performed significantly worse than expected for certain parts of the data, see Table 6.3. We analyzed how the cost functions (6.15) and (6.20) depend on the values of the x -coordinates of \hat{r}_1 and \hat{r}_2 in a small neighborhood around the true joint positions r_1 and r_2 . Figure 6.4 show the level curves of the cost functions for a specific set of data from one estimation run using 250 random samples from the first half and 250 random samples from the second half of the S_6 segment. Similarly, Figure 6.5 show the level curves of the cost functions, but when using 500 random samples from the first half of the S_6 segment instead.

From Figure 6.4 we see that the SS cost function has two distinct local minima for this specific set of data, while the SA cost function only has one global minima. This would explain why the standard deviations of the SS estimates were significantly larger compared to the SA estimates for segments S_6 and S_7 . It is clear from this example that some of the random data samples picked during the estimation runs can cause multiple local minima to appear. The Gauss-Newton method will then converge to the

local minima closest to the initial estimate.

Figure 6.5 show us what happens when the majority of the data consists of rotations of only one of the sensors. We see that the cost functions are much steeper in the r_2 direction than the r_1 direction since the first half of the S_6 segment consists mostly of rotations of the second sensor. A problem that may happen in cases such as these is that the iterative methods may converge to an r_1 coordinate that is anywhere in the valley, since the level curves of the cost function are nearly flat in one direction. This shows how important it is to choose data where both sensors have been sufficiently excited by the motion.

6.6 Discussion

6.6.1 Motion is important

The functionality of the three estimators was verified by the results shown in Table 6.1, where measurements were simulated with an ideal excitation. However, a motion that would allow for such excitation is impossible to perform under the conditions presented by our experimental setup and in most scenarios involving human motion due to limitations in mobility. The experimental data we collected contains multiple different types of motion, and even when using data from segments S_3 - S_7 , the average Euclidean error increases for all three estimators as shown by the results in Table 6.2 when compared to Table 6.1. All three estimators were consistent when using the data from the S_3 - S_7 segments however, as is evident from the standard deviations, which were smaller than 0.05 mm.

The importance of the motion becomes clearer from the results shown in Table 6.3, where data was randomly picked from single segments. The errors vary significantly across the five segments. The LS estimator in particular show poor performance for segments S_3 - S_5 . The motion during those segments is mostly constrained to one or two planes which makes the K -matrix from (6.5) close to singular, causing numerical issues when computing the LS estimate. In Section 6.5.5 we also showed the importance of having sufficient rotation of both sensors as the cost functions tend to become flat in some direction otherwise, as shown in Figure 6.5. Therefore, the best strategy is to combine multiple types of motion which allows rotation of both sensors.

6.6.2 The main difference of the estimators lies in the cost functions

The three estimators minimize different cost functions to find an estimate of r . The cost functions are similar in the sense that they consist of sums

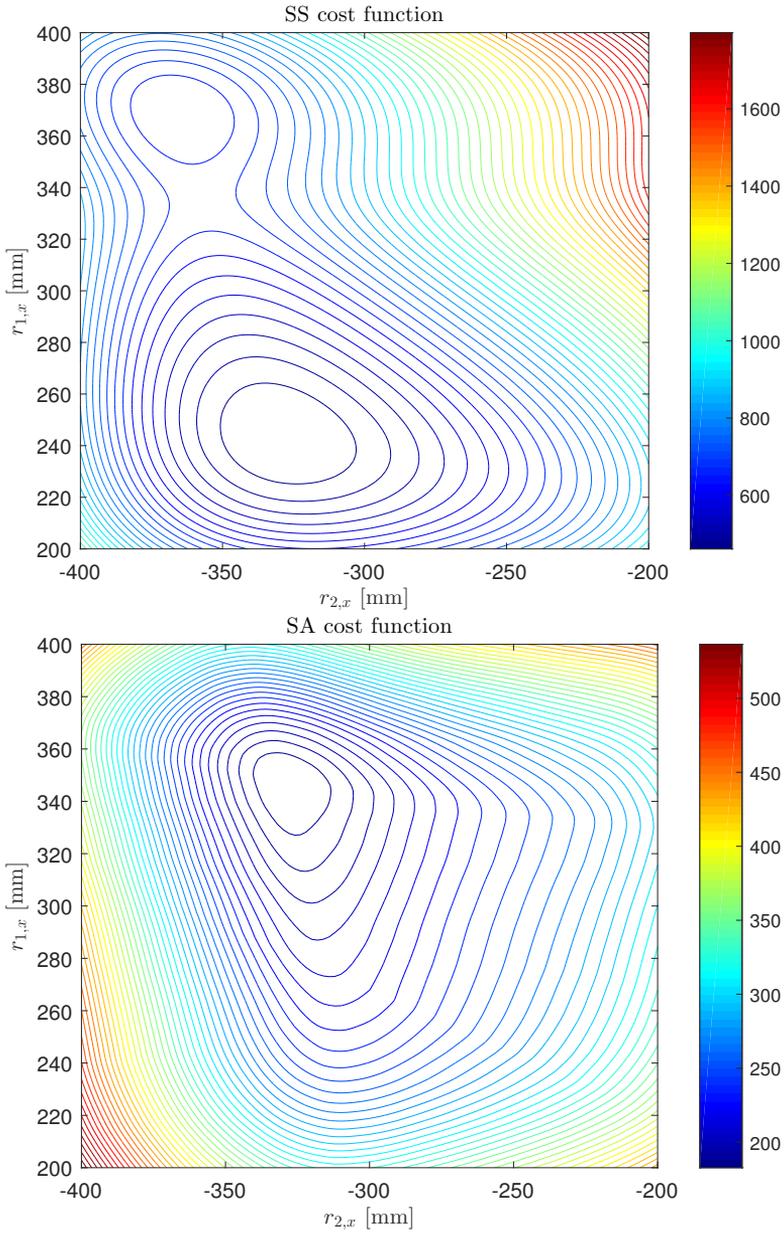


Figure 6.4: The level curves of the SS and SA cost functions in a neighborhood around the true x -coordinates. Using $N = 500$ random samples from both halves of segment S_6 .

of errors that should go to zero as the relationships in (6.1)-(6.2) become satisfied. However, our measurements are corrupted by noise and possibly

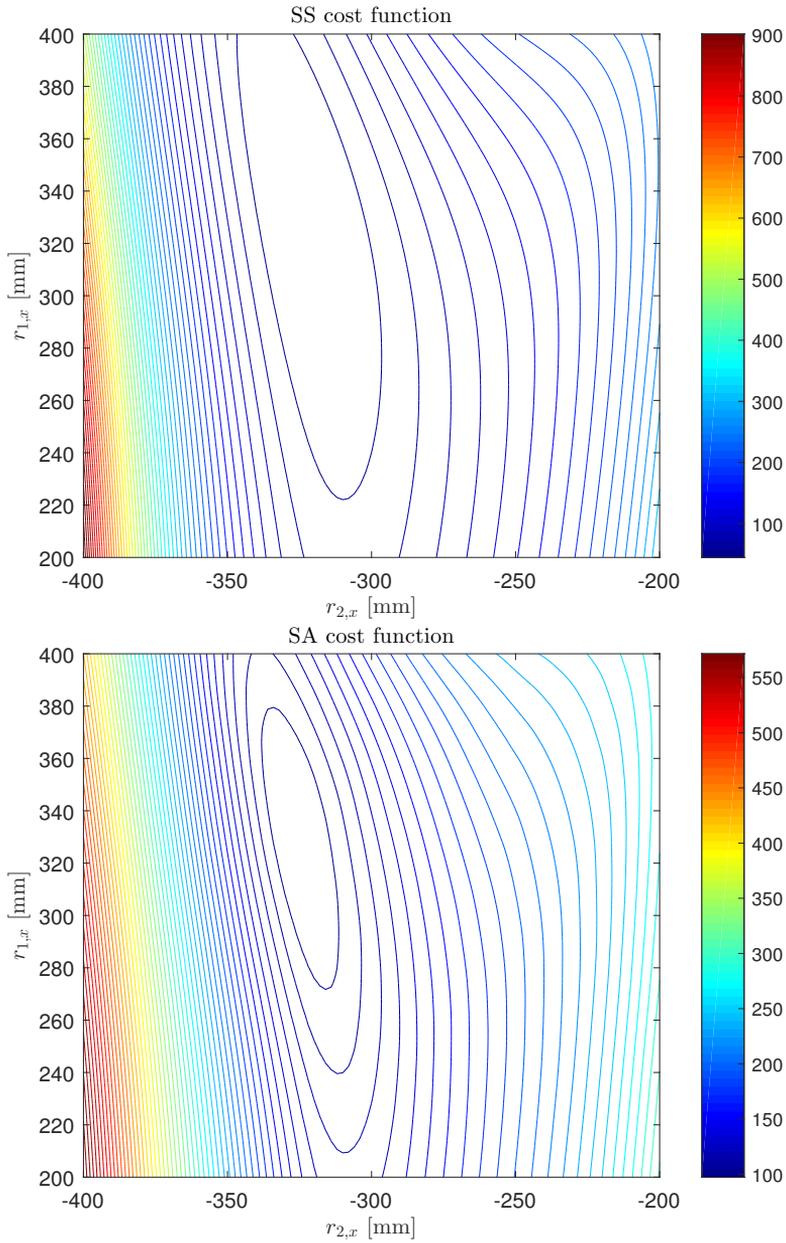


Figure 6.5: The level curves of the SS and SA cost functions in a neighborhood around the true x -coordinates. Using $N = 500$ random samples from only the first half of segment S_6 .

other disturbances that are not taken into account by the model. Measurements that deviate significantly from the model, so called outliers, could

have a significant impact on the estimates depending on how the errors are defined in the cost functions.

The LS (6.13) and the SS (6.15) cost functions model the errors differently. The LS cost function have errors that are linear in the unknown parameter r , but requires additional information about the orientation of the sensors, while the SS cost function have errors that are nonlinear in r , but are invariant to the orientation. What the LS and SS estimators have in common is that they seek to minimize the sum over squared errors, which gives these estimators some nice numerical properties. The LS estimates have a closed form solution, which can be computed efficiently. The SS estimates can be found using a Gauss-Newton method, which typically converges faster than gradient descent methods, since information about the Hessian of the cost function is utilized. However, since the cost functions contain the squared errors, it also means that large errors can have a significant impact on the estimates. Outliers can give rise to multiple local minima as in Figure 6.4, and may pull the estimates away from the true quantity, which will be discussed further in Section 6.6.4.

The SA cost function (6.20) models the errors similarly to the SS cost function, but contains a sum of the absolute errors rather than the squared errors. Outliers will therefore have a less significant impact on the value of the cost function, and will thus affect the estimates less. However, the SA cost function requires more effort to find its minimum compared to the SS cost function. The Hessian of the SA cost function cannot be formed since the absolute value is not differentiable in the origin, and we have to rely solely on the information of the subgradients, resulting in slower convergence.

6.6.3 The least-squares estimator relies on accurate orientation estimates

Another important distinction of the LS estimator is that it requires orientation estimates in the form of the rotation matrices $R_{1,2}$. An EKF with orientation deviation states (Kok 2016) was used to obtain orientation estimates for the whole data set. In essence, this type of EKF uses the gyroscope measurements to track the dynamic rotation of an inertial sensor. Since this involves integrating noisy measurements, the orientation estimates will have an error that increases in magnitude over time. To compensate for this, the EKF may also incorporate measurements from an accelerometer and/or a magnetometer. A stationary accelerometer measures the local gravitational acceleration, which is an observation of the sensor’s inclination and a magnetometer measures the Earth magnetic field which is an observation of the heading. However, to incorporate these measurements in the EKF one has to make some assumptions.

The accelerometer is assumed to be stationary, which is clearly not the case for most of our data. Therefore, only accelerometer measurements of magnitude within $\pm 5\%$ of the magnitude of the local gravitational acceleration is used as an observation of inclination in the EKF. Choosing this threshold is important. A small threshold may cause the integration error to grow, since there will be fewer inclination observations available. On the other hand, a large threshold include more erroneous inclination observations since the assumption of a stationary sensor is violated more. We found 5% to be a good threshold for this data set. However, while tuning this threshold we found that the LS estimation errors varied significantly, which indicates that the LS estimator is sensitive to the orientation estimates. A good strategy could be to record a motion that allows the sensor to be close to stationary at regular intervals. A smaller threshold could then be used since the data set is designed to include regular inclination observations, while excluding most data that violates the assumption of a stationary sensor.

The Xsens MTw sensors used also include magnetometers which could be used to obtain heading observations. However, magnetometers measure magnetic disturbances in addition to the Earth magnetic field, which makes them unreliable in certain environments. We found that only using magnetometer measurements from the S_1 segment to obtain an initial heading of the sensors, but not using the magnetometer measurements as heading observations in the EKF, worked best for our data set.

If we represent the unknown orientation estimation errors as rotation matrices $E_{1,2}$ equation (6.9) is not satisfied. We get

$$\begin{aligned} E_1 R_1 a_1 - E_2 R_2 a_2 &= \\ &= (E_1 - E_2) a_0^G + E_1 R_1 K_1 r_1 - E_2 R_2 K_2 r_2 \\ &\neq E_1 R_1 K_1 r_1 - E_2 R_2 K_2 r_2, \end{aligned} \tag{6.30}$$

where the unknown acceleration of the joint center a_0 remains in the equation. Large orientation estimation errors in combination with large accelerations of the joint center are therefore detrimental to the LS estimator. As mentioned in Section 6.5.3, we only used measurements where the magnitude of the gyroscope measurement was larger than 1.2 rad/s. This threshold also turned out to be important for the LS estimator. It is clear from (6.30) that the $K_{1,2}$ matrices carry information about $r_{1,2}$. Since $K_{1,2}$ depend on the angular velocities and angular accelerations, choosing to only use measurements of sufficiently large magnitude may suppress the error term, $(E_1 - E_2) a_0^G$ in (6.30).

The additional work needed to tune an orientation estimation method to work well with a particular data set may be seen as a disadvantage of

the LS estimator when compared to the SS and SA estimators. However, it is possible that a better strategy for tuning the orientation estimation and selecting a threshold for the magnitude of the measurements could improve the performance of the LS estimator.

6.6.4 The SA estimator is least sensitive to outliers

The results in Table 6.3 shows that for the segments S_6 and S_7 the SS estimator performs significantly worse than the SA estimator. The motion in those segments consisted of spinning the links in an approximately circular trajectory. The spherical joint in our experimental setup is surrounded by cylindrical metallic edges, that collided occasionally during the recording of these motions. The impact from these collisions propagated to the end of the links and were picked up by the sensors, which can be seen as some of the distinct impulse like peaks in Figure 6.3. As the model given by (6.1)-(6.2) describes the acceleration of the sensors as the acceleration of the joint center plus a rotation of the links around the joint center, it does not model these impulses. We can therefore conclude that our data set contain some outliers, particularly located in the S_6 and S_7 segments. It then makes sense why the SA estimator performed better for those segments, as previously discussed in Section 6.6.2 the SA cost function gives less weight to large errors than the SS cost function.

In human motion analysis, the measurements will contain different types of outliers, such as STAs caused by the movement of the skin with respect to the bone. We simulated STAs as an additive disturbance to the accelerometer measurements, comparable to the thigh STA model introduced in (Camomilla et al. 2013). Comparing the results in Table 6.4 with Table 6.2 we see that all three estimators perform worse with these simulated STAs. The SA estimator seem to perform slightly better than the SS estimator. However, taking into account the uncertainty in measuring the position of the joint center, the difference in the accuracy of the SS and SA estimator lies within the margin of the measurement uncertainty. Therefore, we conclude that the SS and SA estimators perform equally well when using the $N = 6280$ samples from segments $S_3 - S_7$, both with and without simulated STAs.

6.7 Conclusion

The main contribution of this paper has been the experimental evaluation of three different methods for estimating the joint position. The estimation is done using inertial measurements collected from sensors attached to the two links that connect via the joint. In terms of accuracy,

the SS and SA estimators performed better over all than the LS estimator. The SS estimator was shown to be more sensitive to impulse like outliers created by colliding metallic parts of the experimental setup, while the SA estimator performed well under these conditions. However, the SS estimator have the advantage of faster convergence. The LS estimator was found to be most sensitive to the type of motion recorded and to be highly dependent on accurate orientation estimates, which makes it less reliable than the SS and SA estimators. Depending on what requirements a potential user of these estimators have in terms of robustness and computation time, either the SS or SA estimators may be used.

Acknowledgment

Thanks to Dave Zachariah and Andreas Svensson for proofreading the manuscript and for providing useful feedback that contributed to improving the quality of the paper.

References

- Boyd, S. and L. Vandenberghe (2004). *Convex optimization*. Cambridge university press.
- Camomilla, V., A. Cereatti, L. Chéze and A. Cappozzo (2013). ‘A hip joint kinematic driven model for the generation of realistic thigh soft tissue artefacts’. In: *Journal of Biomechanics* 46.3, pp. 625–630.
- Crabolu, M., D. Pani and A. Cereatti (2016). ‘Evaluation of the accuracy in the determination of the center of rotation by magneto-inertial sensors’. In: *Sensors Applications Symposium IEEE*.
- James, D. A. (2006). ‘The Application of Inertial Sensors in Elite Sports Monitoring’. In: *The Engineering of Sport 6: Volume 3: Developments for Innovation*. Ed. by E. F. Moritz and S. Haake. New York, NY: Springer New York, pp. 289–294. ISBN: 978-0-387-45951-6.
- Kok, M. (2016). ‘Probabilistic modeling for sensor fusion with inertial measurements’. PhD thesis. Linköping University.
- Leardini, A., L. Chiari, U. D. Croce and A. Cappozzo (2005). ‘Human movement analysis using stereophotogrammetry: Part 3. Soft tissue artifact assessment and compensation’. In: *Gait and Posture* 21.2, pp. 212–225.
- Li, X. R. and Z. Zhao (2006). ‘Evaluation of Estimation Algorithms Part I: Incomprehensive Measures of Performance’. In: *IEEE Transactions on Aerospace and Electronic Systems* 42.4, pp. 1340–1358.
- McGinnis, R. S. and N. C. Perkins (2013). ‘Inertial sensor based method for identifying spherical joint center of rotation’. In: *Journal of Biomechanics* 46, pp. 2546–2549.
- Olsson, F. and K. Halvorsen (2017). ‘Experimental evaluation of joint position estimation using inertial sensors’. In: *Information Fusion (Fusion), 2017 20th International Conference on*. (Xi’an, China). IEEE, pp. 1–8.
- Patel, S., H. Park, P. Bonato, L. Chan and M. Rodgers (Apr. 2012). ‘A review of wearable sensors and systems with applications in rehabilitation’. In: *Journal of NeuroEngineering and Rehabilitation* 9.1.
- Roetenberg, D., H. Luinge and P. Slycke (Apr. 2013). *Xsens MVN: Full 6DOF Human Motion Tracking Using Miniature Inertial Sensors*. Tech. rep. Xsens Technologies.
- Seel, T., T. Schauer and J. Raisch (2012). ‘Joint axis and position estimation from inertial measurement data by exploiting kinematic constraints’. In: *IEEE International Conference on Control Applications*, pp. 45–49.
- Wright, S. and J. Nocedal (2006). *Numerical Optimization*. 2nd. Springer Series in Operations Research.
- Xsens (2017). *MTw Awinda Wireless Motion Tracker*. (Online) <https://www.xsens.com/download/pdf/documentation/mtw2-awinda/MTw2-Awinda.pdf> accessed December 2017.

Title

Identification of nonlinear feedback mechanisms operating in closed loop using inertial sensors

Authors

Fredrik Olsson, Kjartan Halvorsen, Dave Zachariah and Per Mattsson

Edited version of

F. Olsson, K. Halvorsen, D. Zachariah and P. Mattsson (2018). *Identification of nonlinear feedback mechanisms operating in closed loop using inertial sensors*. To be presented at the 18th IFAC symposium on system identification (SYSID). Stockholm, Sweden.

Identification of nonlinear feedback mechanisms operating in closed loop using inertial sensors

Abstract

In this paper we study the problem of identifying linear and nonlinear feedback mechanisms, or controllers, operating in closed loop. A recently developed identification method for nonlinear systems, the LAVA method, is used for this purpose. Identification data is obtained from inertial sensors, that provide information about the movement of the system, in the form of linear acceleration and angular velocity measurements. This information is different from the information that is available to the controller to be identified, which makes use of unknown internal sensors instead. We provide two examples, a simulated neuromuscular controller in standing human balance, and a lead-filter controlling a physical position servo using a DC motor. Both linear and nonlinear controllers are used in the examples. We show that the LAVA method is able to identify sparse, parsimonious models of the controllers.

7.1 Introduction

Systems that operate in closed loop are controlled via feedback, and are typically designed such that the output of the system follows some reference signal. Often, the feedback controllers are designed based on a mathematical model of the system, which, for instance, can be obtained using system identification methods. However, there exists systems in nature that operate in closed loop and whose feedback controllers are unknown to us. For instance, feedback is ubiquitous in biological systems. The human balance system is

one example, where the controller corresponds to the central nervous system (CNS), which senses the posture of the body and activates our muscles, such that we may stand upright against the forces of gravity instead of falling.

In this paper, we investigate the properties of the LAVA method, see Mattsson et al. (2018), for the case of identifying linear and nonlinear controllers operating in closed loop. The LAVA method was developed specifically for identification of nonlinear systems. Here, this method is used to estimate both linear and nonlinear controllers in two different cases, a simulated standing human balance scenario and a real-world controller in a position servo.

7.2 Modeling and methods

7.2.1 Identification of a controller operating in closed loop

We refer to a *plant* as some physical object which we may interact with through some input signal u . Interactions with the system will affect its state and can be observed in the output y . In classic system identification, the objective is to construct a mathematical model that describes the dynamics of the plant, or how y relates to u . The model is constructed based on data, which consist of observed input-output pairs, obtained through experiments. Typically, the experiments consist in exciting the system by choosing u such that the data obtained contains sufficient information about the system dynamics to facilitate identifiability. In that case, we say that u provides sufficient excitation to the system.

The above describes the situation where the plant is a system operating in open loop, which is characterized by the fact that u is chosen manually. In this work we are instead interested in systems that operate in closed loop, see Fig. 7.1, where u is chosen as the output of a controller. The controller can typically be seen as having two inputs, a reference signal r and (measurements of) the outputs y of the system to be controlled. The objective of the controller is then to choose u in order to bring the observed y as close to the reference signal r as possible. Some systems, for example biological systems, exist naturally as closed loop systems, where both the controller and the plant may be unknown.

In order for a controller, that operates in closed loop, to be identifiable, it is not sufficient that the plant output y provides sufficient excitation to the controller. As shown by Ljung (1999), an external excitation signal is also necessary to obtain data which facilitates identifiability. Therefore, we introduce an external disturbance signal d , as an additional input to the plant.

The data that is used to construct a model of the controller will be

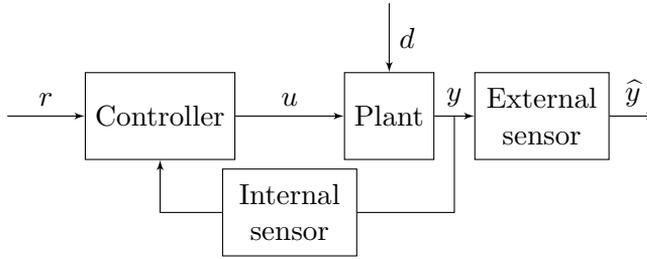


Figure 7.1: A system operating in closed loop, with feedback provided by a controller.

denoted by \mathcal{D} , and referred to as the identification data. The identification data consist of discrete-time measurements \hat{u} and \hat{y} of u and y , respectively

$$\mathcal{D} = \left\{ \left(\hat{u}(t), \hat{y}(t) \right) \right\}, \quad t = 1, \dots, N, \quad (7.1)$$

where t is the sample index and the data set consists of N measurements. We make use of information from external sensors to find the estimate \hat{y} of y . Note that the information we obtain about y may be different from the information that the controller receives, since the signal y is sensed in different ways (see Fig. 7.1). There may be unknown internal sensors that provides the controller with its own internal estimate of y , this is especially true in the case of biological systems. For example, our senses communicate information to the brain, which is used to control our actions, and generally we can not assume that any external sensor will provide the exact same information as the biological sensory systems, to our identification data. Therefore, it's important to keep in mind that the models we construct will be based on the information we have available, which is generally not the same information that the controller has available.

7.2.2 Inertial sensors

We make use of inertial sensors to collect the data used for identification. Inertial sensors consists of accelerometers, that measure linear acceleration, and gyroscopes, that measure angular velocity. These sensors are light-weight and have low power consumption, which makes them suitable for identification of systems where the output can be observed through movement. We will primarily use inertial sensors to obtain observations of the orientation and angular velocity in the experiments explained in Section 7.3 and Section 7.4. Sensor fusion of the accelerometer and gyroscope, and possibly a magnetometer that measures the local magnetic field, makes it possible to estimate the orientation of the sensor. Here, orientation estimates

were obtained using an extended Kalman filter with orientation deviation states, see Kok (2016).

7.2.3 Method for nonlinear system identification

We begin by considering a linear dynamical predictor model of the controller output

$$u(t) = \Theta\varphi(t) + \varepsilon(t), \quad (7.2)$$

where $\varphi(t)$ is the ARX regressor

$$\varphi(t) = [u^\top(t-1) \quad \dots \quad u^\top(t-n_a) \quad y^\top(t) \quad \dots \quad y^\top(t-n_b+1) \quad 1]^\top, \quad (7.3)$$

and n_a and n_b are non-negative integers, which decide the order of the linear predictor model. The matrix Θ contains the model parameters and $\varepsilon(t)$ is the prediction error. This nominal linear predictor yields a parsimonious model of the controller around an operating point. However, as the controller deviates from the operating point, the errors $\varepsilon(t)$ become large due to model errors, and are poorly approximated by a white noise process as assumed in ARX-models.

We extend the nominal predictor by considering a data-driven model of the prediction errors:

$$\varepsilon(t) = Z\gamma(t), \quad (7.4)$$

where $\gamma(t)$ is any given vector-valued function of past inputs and outputs, and Z is a matrix of unknown parameters. The goal then is to identify a model

$$\hat{u}(t) = \hat{\Theta}\varphi(t) + \hat{Z}\gamma(t), \quad (7.5)$$

where the first term tries to capture linear structures in the data and the second term is an overparametrized error model that tries to capture deviations from the nominal linear predictor. The identification method used is the latent variable (LAVA) approach in Mattsson et al. (2018), which regularizes the overparameterized model learning towards the linear predictor class in a data-adaptive manner and seeks a sparse parameter matrix \hat{Z} . This is a property which makes the LAVA method favour parsimonious models, meaning linear models with as few parameters as possible are favoured over nonlinear models with a large set of parameters.

In the examples below, we let $\gamma(t)$ be the nonlinear Laplace operator basis (Mattsson et al. 2018; Solin and Särkkä 2014). This is a multiscale basis with universal approximation properties, similar to wavelet basis.

7.2.4 Evaluation metrics

We use validation data that is separate from the identification data, as input to the identified model to simulate the controller output, which is then used to evaluate the model. The simulated controller output is computed recursively as

$$\begin{aligned} \hat{u}_s(t) = f\left(\hat{u}_s(t-1), \dots, \hat{u}_s(t-n_a), \right. \\ \left. \hat{y}(t), \dots, \hat{y}(t-n_b+1); \hat{\Theta}, \hat{Z}\right), \end{aligned} \quad (7.6)$$

where $f(\cdot)$ is the function that corresponds to (7.5), and the simulation is performed for $t = L, \dots, N$, where $L = \max(n_a, n_b) + 1$ and

$$\hat{u}_s(L-k) = \hat{u}(L-k), \quad k = 1, \dots, n_a. \quad (7.7)$$

We will use the RMSE and FIT metrics to evaluate the performance of the identified models. Here, we define the RMSE as the root of the average squared simulation errors

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{t=1}^N \|\hat{u}_s(t) - u(t)\|_2^2}, \quad (7.8)$$

and the FIT metric as

$$\text{FIT} = 100 \left(1 - \frac{\|u - \hat{u}_s\|_2}{\|u - \bar{u} \mathbf{1}\|_2} \right), \quad (7.9)$$

where \bar{u} is the empirical mean of u and $\mathbf{1}$ is a vector of ones.

We also compare the number of elements in \hat{Z} , n_z , to the number of identified nonzero elements, which is given by the l_0 -norm, $\|\hat{Z}\|_0$. This will serve as a metric for sparsity.

7.3 Simulated standing human balance

The state of a standing human is determined by the orientations and angular velocities of the body segments. For upright standing, the desired state is to have the legs and trunk aligned with the upright vertical orientation of the environment and that the angular velocities of these segments are zero. Deviations from this desired state are communicated to the CNS through the visual, vestibular and proprioceptive sensors. The CNS will then process this information and provide feedback by activating the appropriate muscles, which results in corrective joint torques. The feedback

mechanism, which generates corrective joint torques in order to bring the state closer to the desired state, will be referred to as the neuromuscular controller.

A commonly used model of a standing human is the plane inverted double-pendulum model, where the lower pendulum segment represents the legs and the upper pendulum segment represents the upper body. The two joints in this model represent the ankle and hip joints. This model is motivated by the balancing strategies known as the ankle and hip strategies, which have been found to be the main contributing strategies in upright standing (Hof 2007). Furthermore, a standing human is known to move predominantly in the sagittal plane, provided that the stance width is sufficient (Day et al. 1993).

The inverted double-pendulum has two degrees of freedom ϕ_1 and ϕ_2 , defined as the angles between the upright vertical direction and the leg and trunk segments, respectively. We will use subscript 1 to denote quantities attributed to the leg segment and ankle joint, and 2 for the upper body segment and hip joint. The output of the inverted double-pendulum system is $y = [\phi_1 \ \phi_2 \ \dot{\phi}_1 \ \dot{\phi}_2]^\top$, which contains the angular velocities in addition to the joint angles. The output of the neuromuscular controller are the joint torques $T = [T_1 \ T_2]^\top$. The external disturbance, $d = [d_1 \ d_2]^\top$, used to excite the system is added to T . Therefore, the input to the inverted double-pendulum is $u = T + d$. In addition to the external disturbance we add an internal disturbance v , which models biological noise that acts on the information that the neuromuscular controller receives from the biological sensors about the state of the system. A block-diagram of the standing human balance system is shown in Fig. 7.2.

The identification data obtained from inertial sensors consists of

$$\mathcal{D} = \left\{ \left(\widehat{T}(t), \widehat{y}(t) \right) \right\}, \quad t = 1, \dots, N, \quad (7.10)$$

where \widehat{y} is the estimated output of the inverted double-pendulum and \widehat{T} is the estimated controller output.

Simulation model.

The model of the neuromuscular controller that we use in our simulations is the same as used in Engelhart et al. (2016). This controller consists of a passive part C_p and an active part C_a . Both C_p and C_a contain a proportional and a derivative (PD) feedback from the joint angles, which means that the controller also uses angular velocity information $\dot{\phi}_1, \dot{\phi}_2$. The C_a component models the part of the joint torque that comes from muscle activation via neural signals, and contains neural time delay and muscle

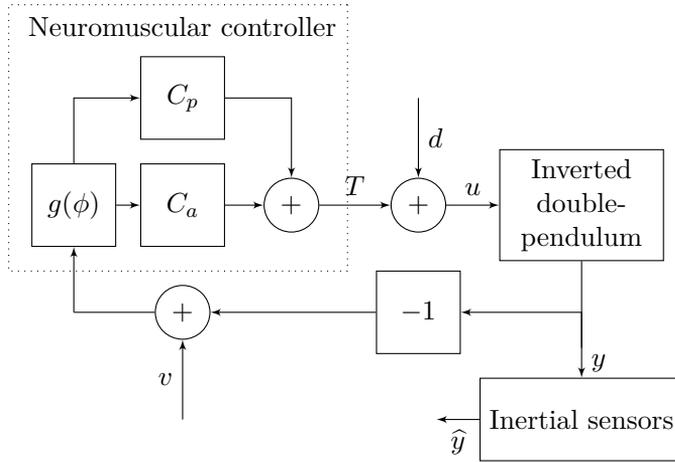


Figure 7.2: A model of standing human balance as a closed loop system.

activation dynamics. The reference signal is constant and corresponds to a zero vector, since a stable upright posture corresponds to $\phi_1 = \phi_2 = 0$ and $\dot{\phi}_1 = \dot{\phi}_2 = 0$.

The equations of motion of the inverted double-pendulum model was calculated using Lagrangian formalism and then linearized using Taylor expansion to obtain a linear state-space model. Physical parameters of the inverted double-pendulum, such as mass, segment length and moment of inertia, were also taken from Engelhart et al. (2016).

The disturbance d was a two-dimensional multisine signal, where the periodic components had a 10 Nm amplitude. The periodic components were initialized at a random phase and excited 27 frequencies, which were logarithmically spaced in the frequency band between 0.05 Hz to 5 Hz. The disturbance signal was simulated for 80 s, which is equal to four periods of the lowest frequency component. After the first two periods, the sign of the d_2 was inverted to make it independent of d_1 , since both d_1 and d_2 excite the same frequencies. This is equivalent to the multiple experiment approach to excite MIMO systems when using multisine signals (Pintelon and Schoukens 2012). The internal disturbance v was simulated as zero mean Gaussian noise with variance $5 \times 10^{-3} \text{ rad}^2$ that is filtered through a first-order low pass filter with static gain 1 and time constant 1 s.

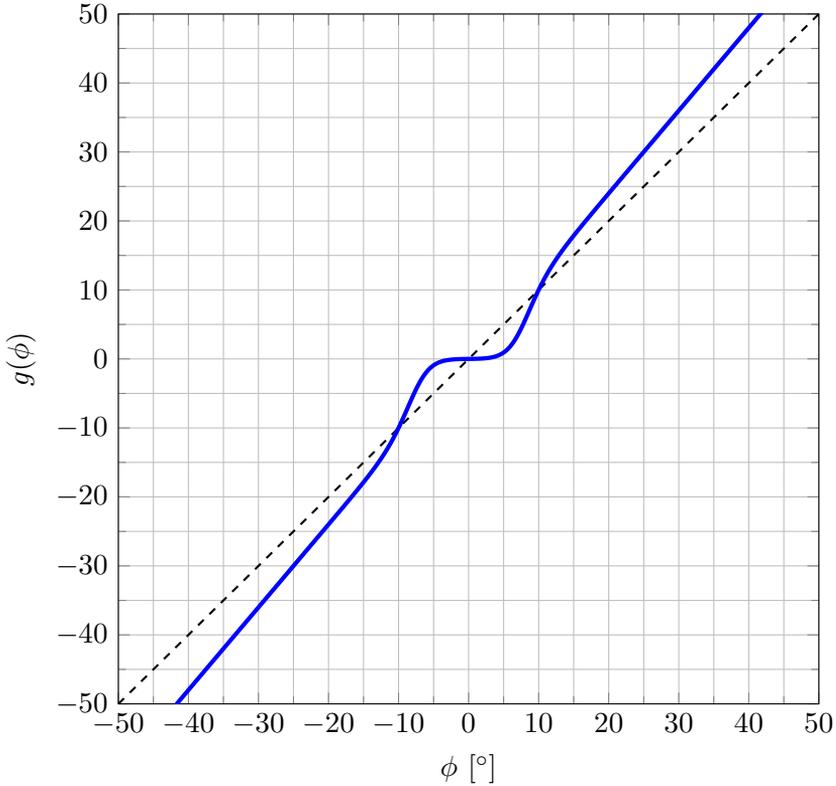


Figure 7.3: The nonlinearity $g(\phi)$ that was applied to the input of the neuromuscular controller (solid blue) compared to the line ϕ (dashed black).

Nonlinear controller.

A nonlinear controller was also implemented by applying the following function to the input to the neuromuscular controller

$$g(\phi) = \frac{e^{\phi' - b}}{e^{\phi' - b} + 1} K\phi \quad (7.11)$$

$$\phi' = \frac{|\phi|\phi_0}{b - \ln(K - 1)}, \quad (7.12)$$

which is a logistic sigmoid function that is modulated by the function $K\phi$. This nonlinearity has the effect that angles below ϕ_0 will be underestimated by the controller and angles above ϕ_0 will approach the line $K\phi$ as $|\phi| \rightarrow \infty$ for $K \geq 0$. Small angles will be close to zero and $b \geq 0$ affects how flat the function is for small ϕ . Fig.7.3 shows the nonlinearity $g(\phi)$ for $K = 1.2$, $b = 5$ and $\phi_0 = 10^\circ$, which was used in the simulations to obtain data from a nonlinear neuromuscular controller. This static nonlinearity models the

Table 7.1: Validation results for different models of the linear neuromuscular controller.

n_a	n_b	n_z	$\ \widehat{Z}\ _0$	RMSE [Nm]	FIT [%]
0	1	0	0	(8.7 6.7)	(94.3 95.0)
0	2	0	0	(8.6 6.3)	(94.4 95.4)
0	3	0	0	(8.6 5.7)	(94.4 95.8)
0	4	0	0	(8.5 5.4)	(94.4 96.1)
0	5	0	0	(8.5 5.3)	(94.5 96.1)
0	6	0	0	(8.5 5.3)	(94.5 96.2)
0	6	162	65	(8.4 5.3)	(94.5 96.1)

Table 7.2: Validation results for different models of the nonlinear neuromuscular controller.

n_a	n_b	n_z	$\ \widehat{Z}\ _0$	RMSE [Nm]	FIT [%]
0	1	0	0	(58.8 40.4)	(59.8 67.3)
0	2	0	0	(42.1 28.8)	(71.2 76.7)
0	3	0	0	(38.4 25.0)	(73.7 79.7)
0	4	0	0	(37.6 24.0)	(74.3 80.6)
0	5	0	0	(37.5 23.6)	(74.3 80.9)
0	6	0	0	(37.7 23.5)	(74.2 81.0)
0	6	162	76	(33.9 21.7)	(76.8 82.4)
0	6	1250	303	(28.5 19.5)	(80.5 84.2)

common hypothesis about human balance that small deviations from the equilibrium are not corrected by the neuromuscular controller.

Measurement models.

Two inertial sensors were simulated to provide the data that was used to identify the neuromuscular controller. One sensor was attached to each segment of the inverted double-pendulum, and the sensors were oriented to have one axis parallel with the segment, and one axis parallel with the rotation axes of the joints. Zero mean Gaussian measurement noise was

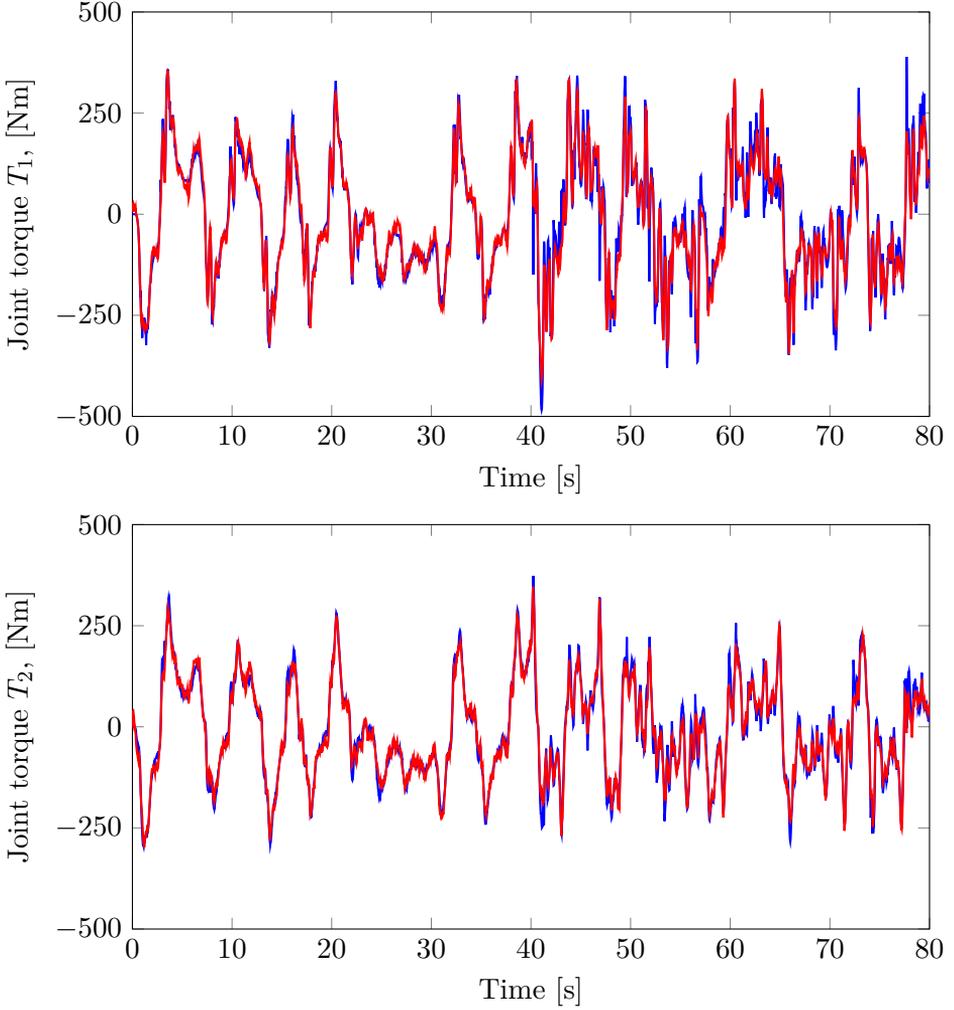


Figure 7.4: The true simulated outputs T of the nonlinear neuromuscular controller (blue) and the simulated output from the identified model (red) for the validation data set.

simulated with variance $1 \times 10^{-2} \text{ m}^2/\text{s}^4$ for the accelerometer and $1 \times 10^{-5} \text{ rad}^2/\text{s}^2$ for the gyroscope. These measurements were then used to obtain an estimate of the states $\hat{y} = [\hat{\phi}_1 \quad \hat{\phi}_2 \quad \hat{\dot{\phi}}_1 \quad \hat{\dot{\phi}}_2]^\top$. The closed loop system of standing human balance was implemented in Simulink (MathWorks, Inc.). The system was sampled with a sample period of $T_s = 0.01 \text{ s}$.

We also need measurements of the joint torque T , generated by the neuromuscular controller, to be able to do identification. These are typically

not available in a realistic situation, but can be estimated from the output of the system, so called *inverse dynamics*. A discrete-time state-space model of the inverted double-pendulum is used, which has the form

$$y(t + 1) = Fy(t) + Gu(t). \tag{7.13}$$

Using the \hat{y} obtained from the inertial sensors, the joint torque can be estimated by solving the following optimization problem

$$\hat{u} = \arg \min_u \sum_{t=1}^{N-1} \|Gu(t) + F\hat{y}(t) - \hat{y}(t + 1)\|_2^2 \tag{7.14}$$

$$+ \|\lambda(u(t + 1) - u(t))\|_2^2$$

$$\hat{T} = \hat{u} - d, \tag{7.15}$$

where λ acts as a smoothing parameter to reduce the noise in the estimate. In our simulation $\lambda = 0.0001$ was chosen. The optimization problem (7.14) was solved using a Gauss-Newton algorithm (Wright and Nocedal 2006), with a sliding window containing 1000 samples and an overlap of 50 samples between consecutive windows.

7.3.1 Results and discussion

Four different data sets were simulated, two using the linear controller and two with the nonlinearity (7.11) applied to the controller input. Different data sets were used for estimation and validation for the linear and nonlinear controllers and each data set contained $N = 8000$ data points. Results for the identified model of the simulated neuromuscular controller are summarized in Table 7.1 for the linear controller, and Table 7.2 for the nonlinear controller.

The simulated neuromuscular controller uses only joint angle and angular velocity information. Therefore, $n_a = 0$ was used for all estimated models, so there is no dependence on past controller outputs. The parameter n_b controls up to how many past controller inputs that the linear model depend on, where $n_b = 1$ means that the linear model only depend on current inputs. It can be seen that increasing n_b barely improves the accuracy for the model of the linear controller. Adding nonlinear parameters does not improve the accuracy, even though 65 nonzero nonlinear parameters were identified. The reason why some nonzero nonlinear parameters were found may be explained by the internal noise process v , which is a filtered white noise process, that is not observable by the inertial sensors. It would be interesting to investigate further, how the internal disturbance affects the identified model. Note also that the nonlinear parameters used in the identification in this example only

depend on current controller inputs to keep the number of parameters low, whereas the linear predictor model depend on $n_b - 1$ past controller inputs as well.

For the model of the nonlinear neuromuscular controller it is clear that a linear model is too restrictive, and that increasing n_b only improves the accuracy to some extent. Adding nonlinear parameters improved the accuracy further. As the the number of nonlinear parameters increased, less nonzero parameters were found relative to the total number of allowed nonlinear parameters. Fig. 7.4 shows the simulated output from the identified nonlinear model with best FIT when applied to the validation data.

7.4 Position servo

A position servo is a closed loop system that consists of a motor, an angular sensor and a feedback controller, where the motor axis angle ϕ should follow a desired reference signal r . We define the control error as $e = r - \phi$, and the objective of the controller is to attain $e = 0$. To achieve this objective the controller will generate a control signal in the form of a voltage u , which is used to drive the motor. For this example we use a DC motor which operates in the voltage range of $\pm 2.5\text{V}$. The DC motor is connected to a flat circular disk, whose rotation angle ϕ is to be controlled.

The identification data collected from each experiment was

$$\mathcal{D} = \left\{ \left(u(t), \hat{\phi}(t), \hat{\dot{\phi}}(t) \right) \right\}, \quad t = 1, \dots, N, \quad (7.16)$$

where the controller output $u(t)$ is known exactly, since we have direct access to the implemented controller. The controller input consists of the estimated angle $\hat{\phi}$ and angular velocity $\hat{\dot{\phi}}$. Thus, the inertial sensors provide more information than the controller uses. Note that the actual controller input are measurements of ϕ that comes from an angular transducer built into the experimental setup, whereas the identification data comes from inertial sensors.

The controller used in the experiments is a lead-filter, with the transfer function

$$C(s) = \frac{0.4386s + 1.519}{0.8661s + 1}. \quad (7.17)$$

The controller was implemented on a computer, with a sample rate of $T_s = 0.025\text{s}$. The difference equation for this linear discrete-time controller has the form of

$$\begin{aligned} u(t) &= 0.9715u(t-1) \\ &+ 0.5064e(t) - 0.4632e(t-1). \end{aligned} \quad (7.18)$$

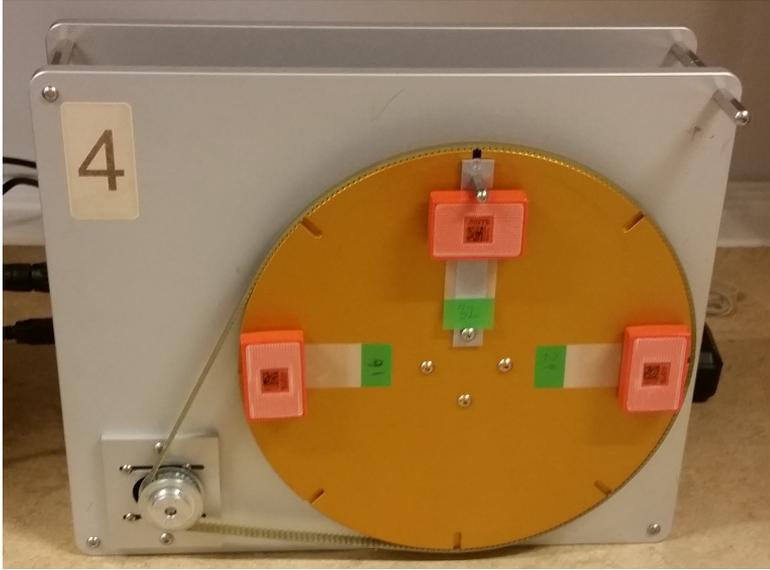


Figure 7.5: The experimental setup for the position servo with inertial sensors attached.

A non-linear controller was also implemented, where a static non-linearity h , defined as

$$h(u, u_{min}, u_{max}) = \begin{cases} 0, & |u| < u_{min} \\ u, & u_{min} \leq |u| \leq u_{max} \\ u_{max}, & |u| > u_{max} \end{cases}, \quad (7.19)$$

which is a saturation and a dead zone, was applied to the controller output $u(t)$ (7.18).

Three Xsens MTw wireless inertial sensors (Xsens 2017), were attached to the circular disk and used to estimate the rotation angle and angular velocity of the disk. The experimental setup can be seen in Fig. 7.5. The disturbance signal d was an additional voltage signal, which was added to the controller output. Therefore, the actual input to the DC motor was $u + d$. The disturbance was generated as a multisine signal with 9 excited frequencies, logarithmically spaced in the frequency band between 0.2 Hz to 8 Hz. Each periodic component had a random initial phase and amplitude 0.3 V.

7.4.1 Results and discussion

Two data sets with $N = 6802$ data points were collected using the linear controller (7.18) and with the added non-linearity (7.19) with $u_{min} = 0.02$ V

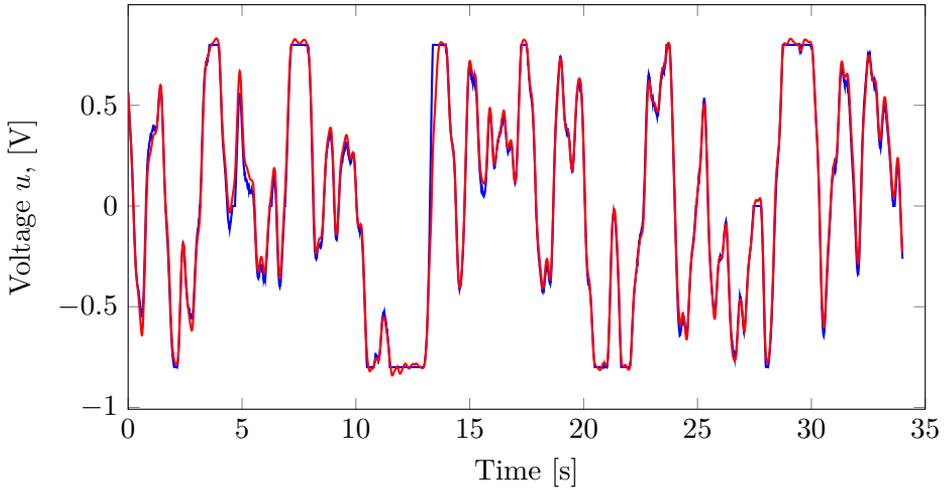


Figure 7.6: The true output u of the nonlinear position servo controller (blue) and the simulated output from the identified model (red) for the validation data set.

and $u_{max} = 0.8\text{ V}$. The first half of each data set was used for estimation and the second half for validation. The constant reference signal $r = 0$ was used for all experiments. Results for the position servo controller are summarized in Table 7.3 for the linear controller, and Table 7.4 for the nonlinear controller.

As the true controller depends on both current and past controller inputs as well as past controller outputs, we try different values for both n_a and n_b . The models of the linear controller were very accurate. Extending the model to include a nonlinear part resulted in only zero-valued nonlinear parameters, which shows that the method is able to identify a parsimonious model in this example.

For the models of the nonlinear controller it can be seen that linear models resulted in poor accuracy. Adding nonlinear parameters improved the model accuracy significantly. Even though as many as 3125 nonlinear parameters were allowed, only 89 were identified as nonzero, producing a simpler, sparse model. Fig. 7.6 shows the simulated output from the identified nonlinear model with best FIT when applied to the validation data.

7.5 Conclusion and future work

In this paper we have investigated the properties of the LAVA method, when used to identify linear and nonlinear controllers operating in closed

Table 7.3: Validation results for different models of the linear servo controller.

n_a	n_b	n_z	$\ \widehat{Z}\ _0$	RMSE [V]	FIT [%]
0	1	0	0	0.1940	58.6
0	2	0	0	0.1740	62.8
1	1	0	0	0.0072	98.5
1	2	0	0	0.0067	98.6
1	2	243	0	0.0067	98.6

Table 7.4: Validation results for different models of the nonlinear servo controller.

n_a	n_b	n_z	$\ \widehat{Z}\ _0$	RMSE [V]	FIT [%]
0	1	0	0	0.2362	53.9
0	2	0	0	0.2338	54.4
1	1	0	0	0.3411	33.4
1	2	0	0	0.3342	34.8
0	2	81	18	0.1532	70.1
0	2	625	36	0.1523	70.3
1	2	243	31	0.0626	87.8
1	2	3125	89	0.0493	90.4

loop. Two examples were provided, a simulated standing human balance scenario and a real position servo using a DC motor. Identification data were obtained through external, inertial sensors. The LAVA method was able to identify sparse nonlinear models that described the dynamics of the nonlinear controllers. In the case of the linear position servo controller, a parsimonious linear model was identified, even though the model allowed for nonlinear parameters. In the case of the linear simulated balance controller some nonzero nonlinear parameters were identified, which did not improve the accuracy of the model.

A flexible model structure is desirable in the identification of real biological feedback systems, where linear models can be too restrictive. The studied method provides flexibility, yet favours the parsimonious, sparse models

in identification. In a future study, we would like to apply the method to identify a real neuromuscular controller in human balance and compare it to previously used methods.

References

- Day, B., M. Steiger, P. Thompson and C. Marsden (1993). ‘Effect of vision and stance width on human body motion when standing: implications for afferent control of lateral sway.’ In: *The Journal of physiology* 469.1, pp. 479–499.
- Engelhart, D., T. A. Boonstra, R. G. K. M. Aarts, A. Schouten and H. van der Kooij (2016). ‘Comparison of closed-loop system identification techniques to quantify multi-joint balance control’. In: *Annual Reviews in Control* 41, pp. 58–70.
- Hof, A. L. (2007). ‘The equations of motion for a standing human reveal three mechanisms for balance’. In: *Journal of biomechanics* 40.2, pp. 451–457.
- Kok, M. (2016). ‘Probabilistic modeling for sensor fusion with inertial measurements’. PhD thesis. Linköping University.
- Ljung, L. (1999). *System Identification: theory for the user*. 2nd. Upper Saddle River, NJ: Prentice Hall.
- Mattsson, P., D. Zachariah and P. Stoica (2018). ‘Recursive nonlinear-system identification using latent variables’. accepted for publication in *Automatica*.
- Olsson, F., K. Halvorsen, D. Zachariah and P. Mattsson (2018). *Identification of nonlinear feedback mechanisms operating in closed loop using inertial sensors*. To be presented at the 18th IFAC symposium on system identification (SYSID). Stockholm, Sweden.
- Pintelon, R. and J. Schoukens (2012). *System Identification: A frequency domain approach*. 2nd. Hoboken, NJ: John Wiley & Sons, Inc.
- Solin, A. and S. Särkkä (2014). ‘Hilbert Space Methods for Reduced-Rank Gaussian Process Regression’. arXiv preprint arXiv:1401.5508.
- Wright, S. and J. Nocedal (2006). *Numerical Optimization*. 2nd. Springer Series in Operations Research.
- Xsens (2017). *MTw Awinda Wireless Motion Tracker*. (Online) <https://www.xsens.com/download/pdf/documentation/mtw2-awinda/MTw2-Awinda.pdf> accessed December 2017.

Paper IV

Title

Non-parametric time-domain tremor quantification with smart phone for therapy individualization

Authors

Fredrik Olsson and Alexander Medvedev

Edited version of

F. Olsson and A. Medvedev (2018). *Non-parametric Time-domain Tremor Quantification with Smart Phone for Therapy Individualization*. Submitted for publication.

Non-parametric time-domain tremor quantification with smart phone for therapy individualization

Abstract

This paper deals with a low-complexity time-domain tremor quantification method for individualization of therapies in medical conditions, where tremor is a cardinal symptom. The method produces a tremor severity estimate based on measurements acquired from a standard sensor platform of a smart phone during a smooth voluntary movement. The estimate is calculated from a data set recorded over tens of seconds but can also be used for unobtrusive tremor monitoring over longer time. Besides tremor amplitude, its frequency is also evaluated over time thus providing the means to distinguish between e.g. rest and action tremor. No analytical model is assumed for the tremor signal form. The characterization of tremor severity is performed by steady-state analysis of a Markov chain, whose states correspond to different intervals of tremor amplitude. The utility of the proposed quantification approach is illustrated by clinical data obtained during a Deep Brain Stimulation programming session. The results are compared with a conventional approach utilizing spectral analysis.

8.1 Introduction

Tremor is an involuntary repetitive movement observed mainly in extremities but also in head and trunk. It is a cardinal symptom of neurological disorders, such as Parkinson's Disease (PD) and Essential Tremor (ET), but is also present in healthy individuals. The latter form is referred to as physiological tremor and worsens under physical or emotional stress.

In PD and ET, both of which are currently incurable conditions, therapy is aimed at alleviating the symptoms of the disease, in particular tremor. Therefore, to characterize therapeutical effect of a medical treatment, a reliable and objective quantification of tremor is necessary. Currently, symptom quantification in neurology is performed by medical personnel using standardized rating scales. For instance, Unified Parkinson's Disease Rating Scale (UPDRS) (The International Parkinson and Movement Disorder Society 2016) takes up tremor under Part II covering motor symptoms and rates it by an integer between zero and four, (Section 2.10). Zero rating corresponds to no tremor and rating four is described in UPDRS as tremor causing problems with most or all daily activities. Apparently, the level of detail in UPDRS is adopted to ocular examination and smaller improvements or deteriorations in the patient condition cannot be adequately captured.

The need for objective and more fine-grained quantification of tremor is motivated by individualized PD treatments. Examples of individualized pharmacotherapies are the infusion treatment with Duodopa, where a pump delivers the drug in gel form to the duodenum (Nyholm 2012), and a levodopa microtablets treatment with a smart dispenser MyFID, (Senek et al. 2017). In both cases, the administered drug dose can be in principle adjusted to the current state of the patient while a reliable method of symptom quantification is missing.

An attempt to use state-of-the-art tremor monitoring technology PKG[®] system from Global Kinetics for dosing of microtablets is described in Johansson et al. (2018). In fact, PKG[®] is designed for longitudinal tremor monitoring and applies off-line data processing which mode is not suitable in titration procedures where a drug-induced change occurs in a matter of minutes. Another recently reported approach utilizes 3D accelerometer and gyroscope measurements during a hand pronation-supination test to capture the dose-effect profiles of PD patients treated with levodopa, (Thomas et al. 2017). The latter technique demands wearable sensors and is not unobtrusive as the test has to be performed each time a patient state estimate is needed.

Another PD therapy that is in great need of individualization is Deep Brain Stimulation (DBS), where the therapeutical effect is achieved by delivering pulsatile electrical stimulation to a certain target area of the brain via implanted leads. The effect of DBS critically depends on the stimulation pulse sequence parameters that are frequency, amplitude, and duty cycle (pulse length). Since a typical DBS lead features from four to eight contacts that also have to be suitably selected for proper stimulation, the complexity of the therapy individualization (usually referred to as DBS programming) is high. In contrast with the pharmacotherapies, the patient's response to a change in the stimulation setting can be observed in a matter of

seconds. Even though model-based methods for DBS individualization are currently under development (Amon and Alesch 2017; Cubo et al. 2016), clinical practice is still based on a trial-and-error procedure of tuning the DBS parameters to best effect evaluated by the clinician and perceived by the patient. Tremor is one of the targeted symptoms in individualizing DBS stimuli and lack of quantification tools effectively prevents objective comparison of the therapeutical effects for different stimulation parameters.

Accurate measurement of tremor is not a challenging task and can be performed in a laboratory environment by any standard motion capture technique. A classical and straightforward approach in this direction is to instruct the patient to draw over a simple geometrical form depicted on the screen of a tablet computer or smart phone (Westin et al. 2010). The deviation from the predefined line is a 2D projection of the tremor movement and suitable for direct symptom severity evaluation. Yet, the expected level of patient collaboration is high and can be perceived as tiring with multiple repetitions. Further, the test situation itself can provoke performance anxiety and potentially worsen the tremor. A non-obtrusive tremor quantification has therefore to build on habitual and unguided voluntary movements.

Once a tremor signal as a deviation from a given voluntary movement is obtained, it has to be processed in order to evaluate tremor severity. A suitable basic technique is then to integrate absolute value of the tremor signal and normalize it with respect to the length of the voluntary movement. However, in most practical cases, the tremor signal is not readily available but has to be estimated from accelerometer measurements. In fact, the voluntary movement is not available either, while it clearly impacts the movement acceleration.

Since derivation is a linear operator and acceleration is second derivative of position, one can attempt to apply signal analysis tools directly to the measured acceleration of the patient's limbs. This is routinely done by assuming (broad-sense) stationarity of tremor meaning that the spectral properties of the signal do not change in time. If, in addition, a spectral decomposition of the voluntary movement and the tremor signal can be asserted, power spectral analysis provides simple and effective tools for tremor quantification (Dai et al. 2015). Yet, as it is shown in Section 8.3 of this paper, actual accelerometer measurements of tremor exhibit time-varying amplitude and frequency. Besides, different peak frequencies appear in different directions of the 3D signal, probably due to nonlinearity of the biomechanical system executing the translational-rotational movement (Stergiou and Decker 2011).

With the intended clinical use in mind, a tremor quantification tool has to provide concise and coherent information to the medical personnel. Fusing measured sensor signals is therefore necessary and a mathematical model is

needed to enable it. As always in engineering and science, the level of detail in such a model is defined by its final use. For tremor quantification, it can be argued that modelling the relations between the measured signals and not the underlying biomechanical system is sufficient because dynamically modifying the system behaviour is not attempted in this framework.

Finally, currently available implementation platforms for tremor quantification deserve to be mentioned. Dedicated devices, such as PKG[®], are popular and attractive from a regulatory point of view, while smart watches and smart phones are gaining on (Senova et al. 2015). The possibility of selecting among a number of competing software solutions on the same hardware platform is highly appealing to both the clinical and off-clinic end user.

The contributions of this paper stand as follows:

- A complete method for tremor quantification in time domain based on data measured with a standard smart phone and producing tremor amplitude distribution is presented.
- Deficiencies of spectral methods of tremor quantification related to time-varying signal amplitude and frequency are demonstrated on clinical data.
- The proposed tremor quantification method is shown to provide a better decision-making ground in comparing therapeutical effect of DBS settings.

The rest of the paper is organized as follows. First, the clinical data sets utilized throughout the paper are presented. Further, spectral analysis is applied to the data sets in Section 8.3 to highlight the difficulties in tremor quantification related to signal non-stationarity and nonlinearity of the underlying biomechanical system. Further, an analytical model-free time-domain approach to tremor signal extraction is explained step-by-step in Section 8.4. The estimated tremor amplitude is characterized by the stationary behavior of a Markov chain designed from data in Section 8.6. The obtained results are discussed with respect to clinical and technological relevance in Section 8.7 and, finally, Conclusions are drawn.

8.2 Data acquisition

The experimental data used in the present study have also been described and utilized in previous work (Medvedev et al. 2017). These data have been obtained from one PD patient with an implanted DBS system. Data were collected during a DBS programming session at the Uppsala University Hospital on May 30, 2017, under ethical approval obtained from Uppsala

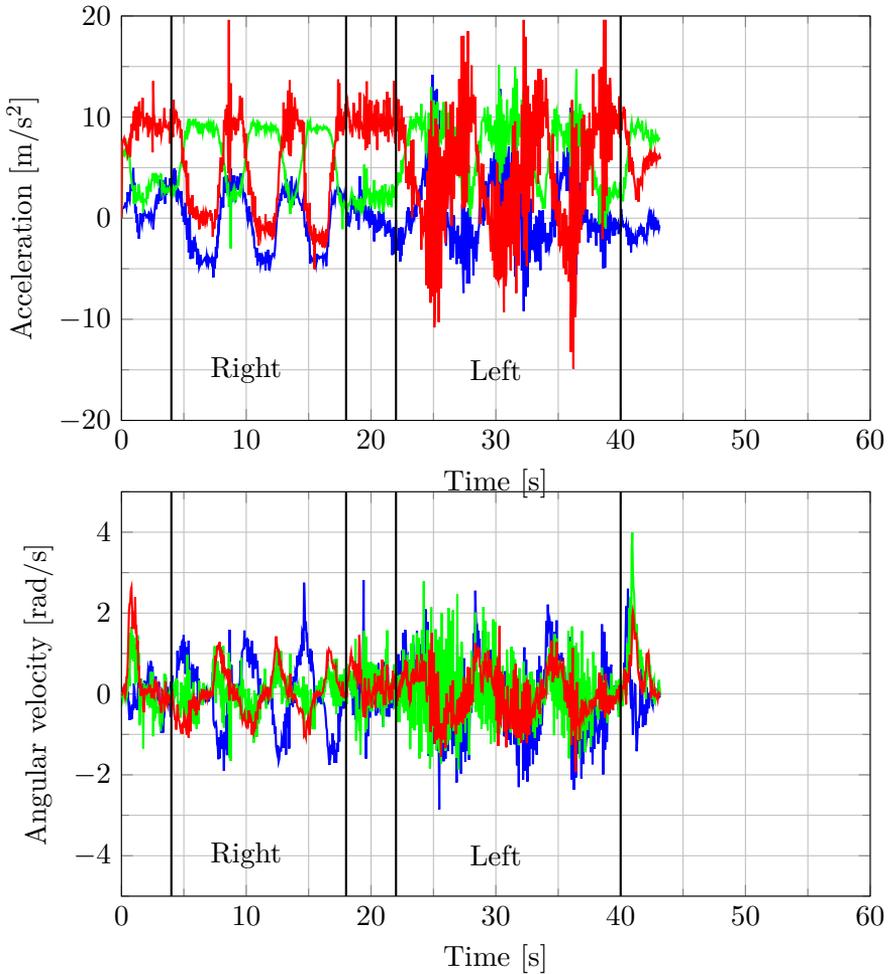


Figure 8.1: Raw accelerometer and gyroscope signals from the trial with DBS off. The colors denote the x- (blue), y- (green) and z-axes (red) of the respective sensor. The black solid lines indicate which data that belongs to the movement with the right and left hands, respectively. Significant tremor was observed visually for the left hand.

Ethical Review Board. A Samsung Galaxy S5 smart phone was used to collect the data. Only signals from the native sensors of the phone were recorded. Linear acceleration and angular velocity were measured by the accelerometer and the gyroscope, respectively, with the MPU-6500 inertial sensor platform manufactured by InvenSense (InvenSense 2017). The local magnetic field was measured by the AK09911C magnetometer manufactured by Asahi Kasei Microdevices (Asahi Kasei Microdevices Corporation 2017).

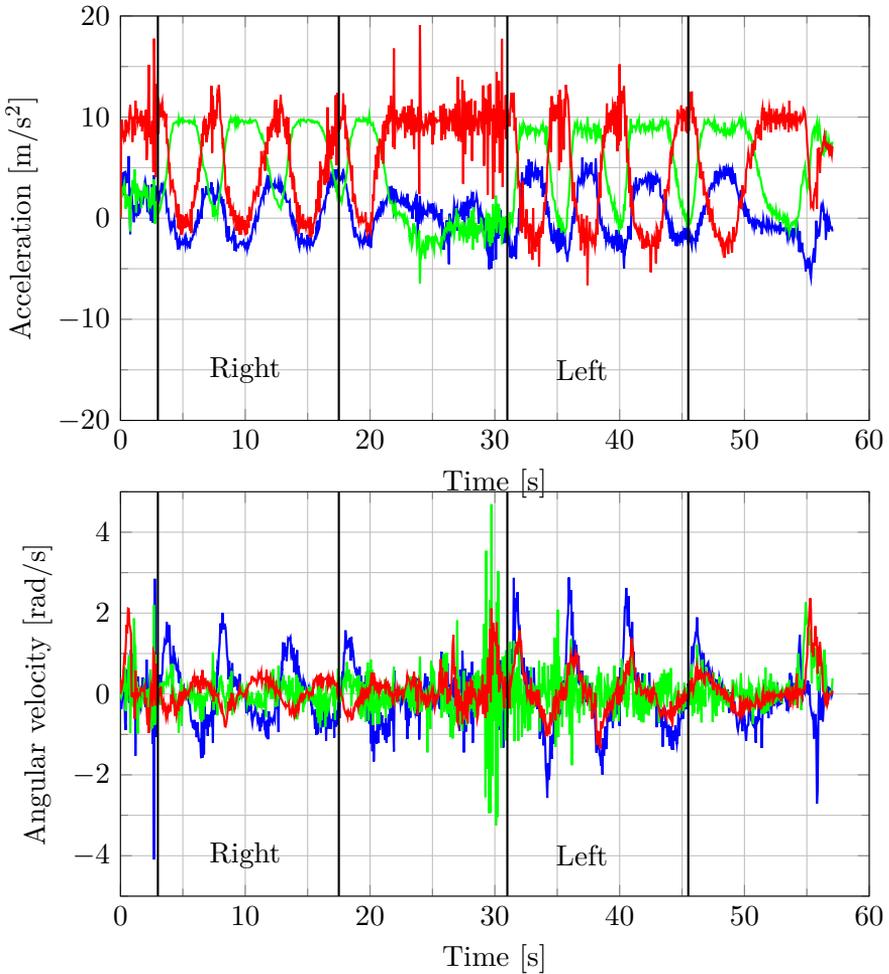


Figure 8.2: Raw accelerometer and gyroscope signals from the trial with DBS1 settings. The colors denote the x- (blue), y- (green) and z-axes (red) of the respective sensor. The black solid lines indicate which data that belongs to the movement with the right and left hands, respectively. Note that here the patient performed four repetitions of the movement instead of three. However, only the first three cycles were used for modeling purposes.

The sample rate of all sensors was $f_s = 100\text{Hz}$.

Data were collected while the patient performed a predefined task with the smart phone. The instructions for the task were as follows. *Pick up the phone from the table, move it to the ear and put then it back to the same position.* The patient was asked to perform this task three times with each hand. The same procedure was repeated for three separate DBS settings:

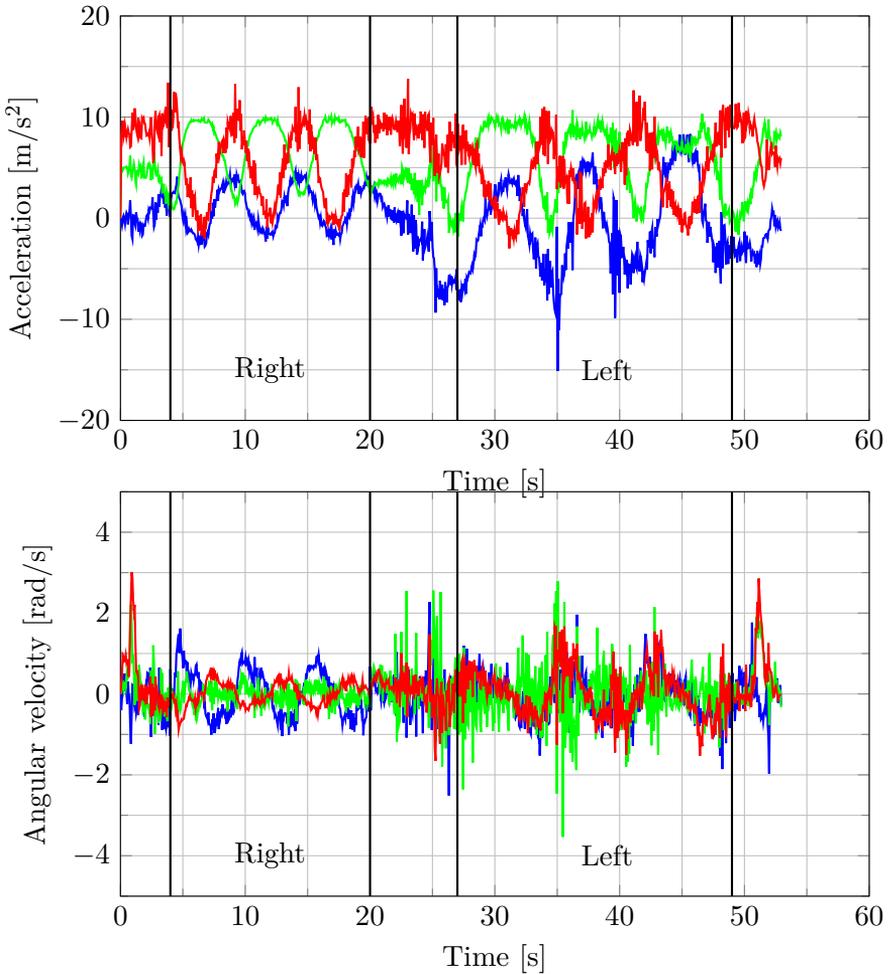


Figure 8.3: Raw accelerometer and gyroscope signals from the trial with DBS2 setting. The colors denote the x- (blue), y- (green) and z-axes (red) of the respective sensor. The black solid lines indicate which data that belongs to the movement with the right and left hands, respectively.

DBS turned completely off (DBS off), the settings that the patient had since before the visit to the hospital (DBS1), and a new settings to be compared with the old one (DBS2). The data collection procedure took less than one minute for each of the DBS settings.

Fig. 8.1–Fig. 8.3 show the raw accelerometer and gyroscope data obtained from the smart phone for the three different DBS settings. Two important features can be noticed in the raw data sets: First, the signals are definitely not stationary, with both frequency and amplitude of the signals

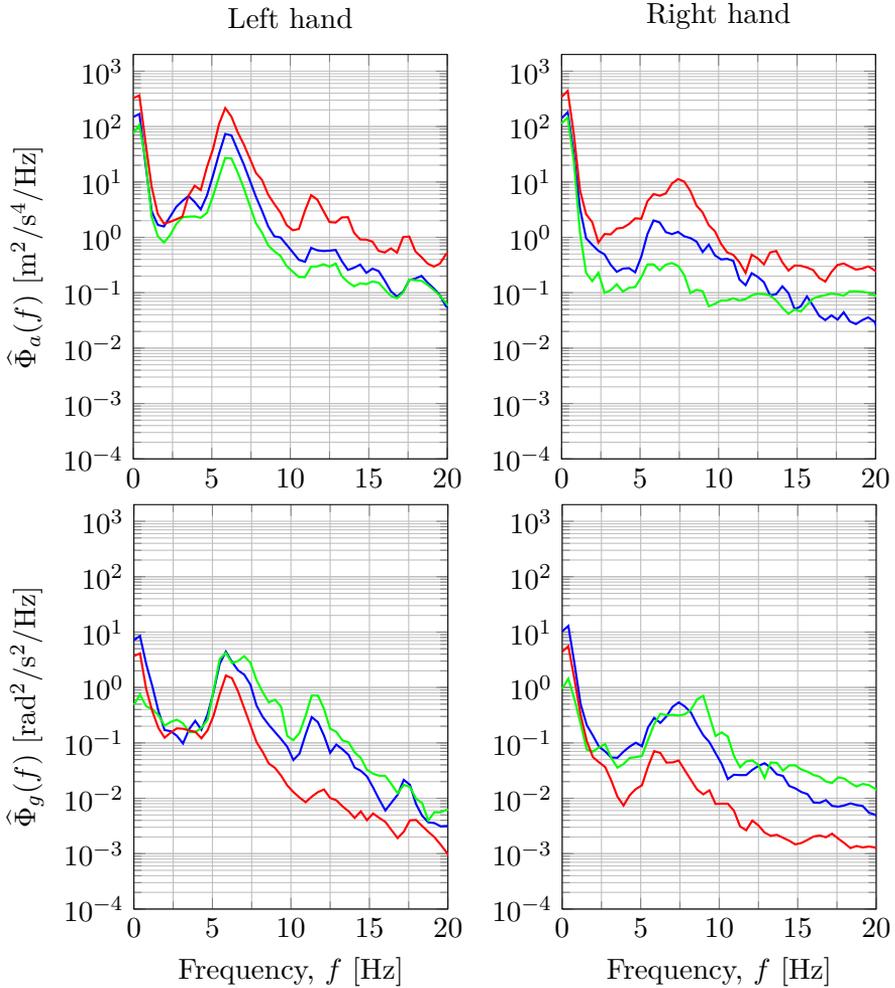


Figure 8.4: PSD estimates of the detrended accelerometer and gyroscope signals with DBS turned off. The colors denote the x- (blue), y- (green) and z-axes (red) of the respective sensor.

varying throughout the record time. Second, differences between Fig. 8.2 and Fig. 8.3 corresponding to DBS on are difficult to discern by ocular inspection. On the contrary, the signals in Fig. 8.1 clearly confirm more prominent tremor with DBS off. The moment when the phone reaches the highest elevation point is also easy to recognize in each of the plots.

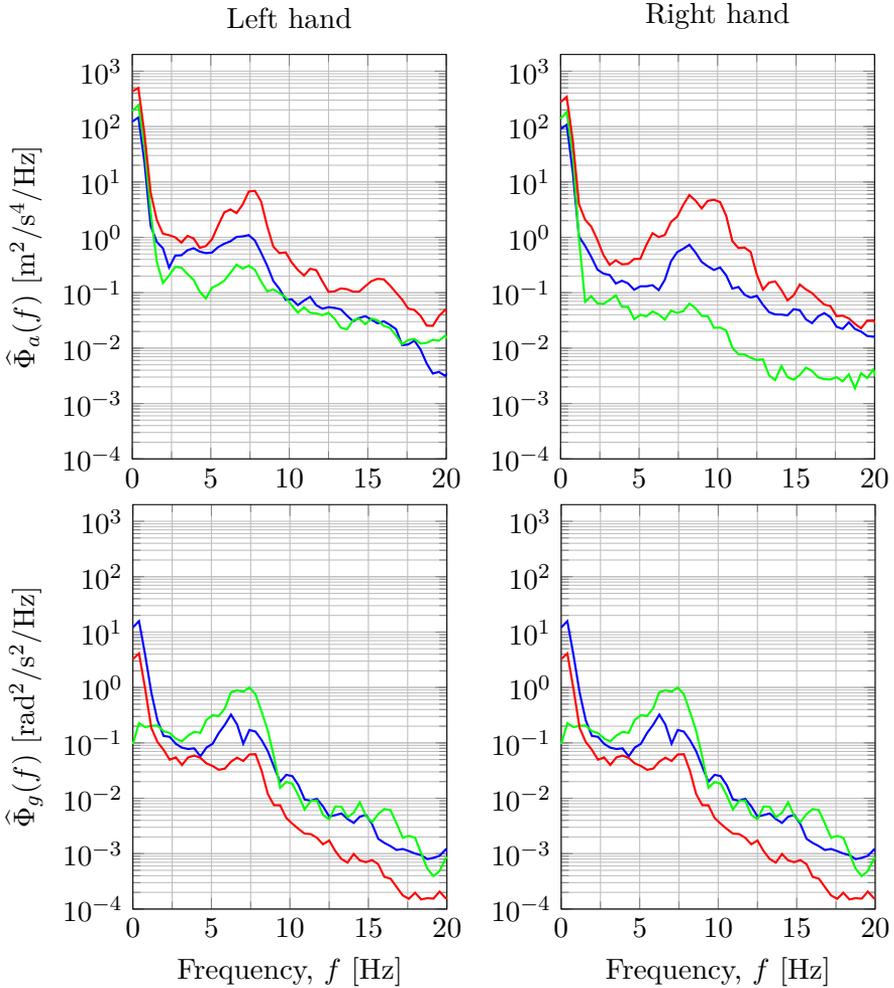


Figure 8.5: PSD estimates of the detrended accelerometer and gyroscope signals for the DBS1 setting. The colors denote the x- (blue), y- (green) and z-axes (red) of the respective sensor.

8.3 Spectral analysis methods

The most commonly encountered techniques for tremor quantification fall within the domain of spectral analysis (Stoica and Moses 1997). Spectral analysis generally involves estimation of the Power Spectral Density (PSD) for the signals of interest. For example, the PSD of a signal is informative regarding what power it carries in the frequency band that is characteristic for tremor, which typically assumed to lie between 3 to 12 Hz (Dai et al. 2015).

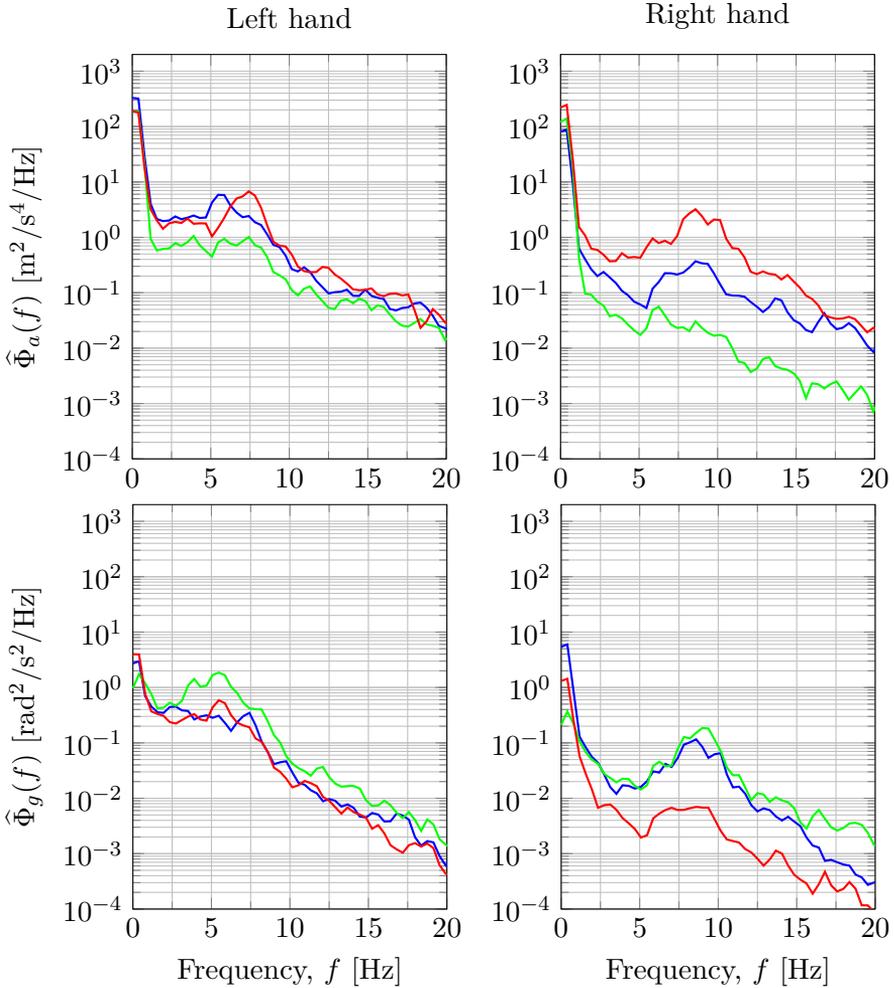


Figure 8.6: PSD estimates of the detrended accelerometer and gyroscope signals for the DBS2 setting. The colors denote the x- (blue), y- (green) and z-axes (red) of the respective sensor.

PSD estimates $\hat{\Phi}_a(f)$ and $\hat{\Phi}_g(f)$ are obtained here for each axis of the accelerometer measurements, y_a , and the gyroscope measurements, y_g , respectively, using Welch's method with a 2s Hamming window and a 50% overlap. The accelerometer and gyroscope data are also split into separate data sets for the right and the left hand since tremor is usually more pronounced on one side than on the other. The resulting PSD estimates are shown in Fig 8.4–Fig. 8.6.

To compute the signal power in a frequency band lower bounded by f_1 and upper bounded by f_2 , the PSD estimate is integrated with respect to

Table 8.1: The computed tremor power for all sensor axes across all trials and for both hands. The lowest tremor powers for each hand and axis is highlighted in boldface.

Hand	Trial	Tremor power [(Signal unit) ²]					
		acc-x	acc-y	acc-z	gyr-x	gyr-y	gyr-z
Right	DBS off	6.265	1.356	31.27	1.489	1.845	0.1928
	DBS1	2.218	0.3142	17.77	0.6252	0.8501	0.0761
	DBS2	1.418	0.2032	10.1	0.3614	0.5434	0.03608
Left	DBS off	112.9	44.2	298.3	7.246	10.5	2.551
	DBS1	4.044	1.196	15.87	0.7867	2.6	0.2473
	DBS1	16.79	4.382	18.13	1.383	5.295	1.543

frequency

$$P_{\text{tr}} = \int_{f_1}^{f_2} \widehat{\Phi}(f) df. \tag{8.1}$$

By selecting $f_1 = 3\text{Hz}$ and $f_2 = 12\text{Hz}$, the contribution of tremor to the power of a signal can then be estimated. This quantity will be referred to as the tremor power in a signal.

The dominant frequency of tremor is defined as the frequency within the interval $\{f_1, f_2\}$ where the PSD estimate is the highest, i.e.

$$f_{\text{max}} = \arg \max_{f \in \{f_1, f_2\}} \widehat{\Phi}(f). \tag{8.2}$$

The dominant frequency f_{max} is typically used to distinguish between different types of tremor.

The tremor power and dominant frequencies were computed for all PSD estimates and are shown in Table 8.1 and Table 8.2. The integral in (8.1) was evaluated numerically using the trapezoidal method.

8.4 Extracting the tremor signal in time domain

Tremor can be naturally viewed as a repetitive involuntary deviation in position from a voluntary movement. In fact, this interpretation agrees well with how physicians rate tremor: The perceived amplitude of the deviations is a measure of the symptom severity and their frequency is a secondary parameter that helps to distinguish between different tremor types. A fundamental difficulty in tremor quantification is then that the actual voluntary motion cannot be measured as it is only intentional and resides in the mind

Table 8.2: The computed dominant frequencies for all sensor axes across all trials and for both hands.

		Dominant frequency [Hz]					
Hand	Trial	acc-x	acc-y	acc-z	gyr-x	gyr-y	gyr-z
Right	DBS off	5.86	7.03	7.42	7.42	8.98	5.86
	DBS1	8.2	3.52	8.2	8.2	8.2	8.2
	DBS2	8.59	6.25	8.59	8.59	8.98	8.59
Left	DBS off	5.86	5.86	5.86	5.86	5.86	5.86
	DBS1	7.42	6.64	7.81	6.25	7.42	7.81
	DBS1	5.47	3.91	7.42	3.52	5.47	5.47

of the patient. Another difficulty is in the non-stationarity of tremor. Variations in tremor amplitude throughout a voluntary movement are indicative of whether the patient exhibits rest tremor or action tremor. Therefore, not only mean tremor amplitude is clinically important but the amplitude evolution of the tremor signal as well.

This section describes how to obtain a 2D estimate of the tremor signal. An estimate on a plane is used since the tremor component along the voluntary motion is difficult to distinguish and quantify. The procedure is covered in detail in previous work (Medvedev et al. 2017) but recapitulated here to facilitate readability and consists in the following four steps:

Step 1. Orientation estimation: Estimate the orientation of the sensor platform with respect to an Earth-fixed navigation frame.

Step 2. Position estimation: Estimate the acceleration of the sensor in the navigation frame and double integrate to estimate the position.

Step 3. Separation of tremor and voluntary movement: Estimate the trajectory of voluntary movement and subtract it from the position estimates.

Step 4. Planar projection and rotation: Project the 3D tremor signal onto planes perpendicular to the gradient of the voluntary movement. Then rotate all planes into the plane with zero vertical component.

8.4.1 Acceleration measurement model

The accelerometer measurements are modelled as

$$y_a(t) = R(t)(a(t) + g) + e_t, \quad (8.3)$$

where $t = 1, \dots, N$ denotes the sample index and e_t is measurement noise that is assumed to be zero-mean Gaussian. The measurements $y_a(t)$ are expressed in the reference frame that is fixed in the accelerometer triad of the sensor platform, referred to as the sensor frame. The true acceleration of the sensor is denoted by $a(t)$ and g the gravitational acceleration. Here, both $a(t)$ and g are expressed in a earth-fixed reference frame that is referred to here as the navigation frame. The orientation of the sensor frame with respect to the navigation frame is obtained through the rotation matrix $R(t)$. The rotation matrices belong to the special orthogonal group $R(t) \in \text{SO}(3)$, and multiplying $R(t)$ with any vector expressed in the navigation frame maps that vector in the sensor frame.

8.4.2 Step 1. Orientation estimation

To estimate the acceleration, $a(t)$, of the sensor platform in the navigation frame, the rotation matrix $R(t)$ needs to be evaluated. An extended Kalman filter (EKF) with orientation deviation states (Kok 2016) was used to obtain estimates $\hat{R}(t)$ of the rotation matrix. The EKF is a sensor fusion algorithm where the information from the gyroscope, accelerometer and magnetometer are fused together based on a mathematical model.

When the EKF is initialized, the navigation frame is defined such that it remains fixed throughout the entire orientation estimation procedure. The initial accelerometer and magnetometer measurements of the sensor as it remains stationary are used to define the navigation frame. The initial orientation is defined as the unit matrix, $R(1) = I$, and the vertical direction is chosen to be aligned with the measured gravitational acceleration g . Similarly, the initial magnetometer measurements are used to define the "north" direction in the navigation frame.

The angular velocity of the sensor platform is observed through the gyroscope. In the time update step of the EKF, the gyroscope measurements are integrated once to update the orientation state.

Integrating gyroscope measurements over time will cause errors to accumulate in the orientation estimates, something that is often referred to as integration drift. To compensate for the drift, the EKF uses the accelerometer and magnetometer measurements to provide absolute orientation references. A stationary accelerometer measures g in the sensor frame, which is an observation of the tilt angle (angle with respect to the vertical). The magnetometer provides the heading or azimuth angle.

8.4.3 Step 2. Position estimation

Position estimates obtained via double integration of the accelerometer signal are prone to drift. The estimated acceleration in the navigation frame contains errors due to the noise in the accelerometer measurements as well as errors in the estimated rotation matrix. The drift phenomenon leads error accumulation over time and, inevitably, to the position estimates divergence from the true position of the sensor.

Since the predefined movement of the sensor in the experimental setup ends at the same position as it starts with, it can be used to reduce the drift. This is done by removing a linear trend from the acceleration signal thus ensuring that the position estimates return to the initial position after the double integration.

With $\widehat{R}(t)$ obtained by the EKF, the acceleration of the sensor in the navigation frame is estimated as

$$\widehat{a}(t) = \text{detrend}(\widehat{R}^\top(t)y_a(t) - g), \quad (8.4)$$

where $\text{detrend}(\cdot)$ is the function that removes a linear trend across all N available measurements for one closed cycle. A simple double integration scheme can then be used to obtain velocity estimates, $\widehat{v}(t)$, and position estimates $\widehat{p}(t)$

$$\widehat{v}(t+1) = \widehat{v}(t) + T\widehat{a}(t), \quad (8.5)$$

$$\widehat{p}(t+1) = \widehat{p}(t) + T\widehat{v}(t) + \frac{T^2}{2}\widehat{a}(t), \quad (8.6)$$

where T is the sample period and the sensor is assumed to be at rest in the beginning of the experiment, i.e. $\widehat{v}(1) = 0$ and $\widehat{p}(1) = 0$.

8.4.4 Step 3. Separation of tremor and voluntary movement

As previously stated, a typical frequency band for tremor is 3–12Hz. Therefore, frequency components under 3Hz are assumed to belong to the voluntary movement. To estimate the trajectory of the voluntary movement, a Savitzky-Golay filter (Savitzky and Golay 1964) with cubic polynomials is utilized to smooth the position estimates. Other smoothing or filtering techniques are also known to work well, as reported in Dimitrakopoulos et al. (2017). As signal processing is not necessarily performed on-line, anti-causal filtering that does not introduce additional phase shift is feasible here.

The 3D tremor signal is then evaluated as

$$p_{t3d}(t) = p_v(t) - \widehat{p}(t), \quad (8.7)$$

where $p_v(t)$ are the Savitzky-Golay filtered position estimates $\hat{p}(t)$. An important parameter of the Savitzky-Golay filter is the window size. To obtain a good estimate of the voluntary movement, the window size should be sufficiently large to cover multiple periods of tremor oscillations but not so large that it filters the trajectory of the voluntary movement. Judging from Fig. 8.1–Fig. 8.3, the period time for the voluntary movement was in the range of 3 – 6s. It has been found that a good rule of thumb for selecting a window size for the considered voluntary movement is to set it to three periods of the lowest tremor frequency component.

8.4.5 Step 4. Planar projection and rotation

The final step of extracting the tremor signal is to obtain a 2D projection for further quantification. This is done by first projecting the 3D tremor signal onto planes that are perpendicular to the gradient of the voluntary movement trajectory, see Fig. 8.7. Each plane is then successively rotated into an orientation with zero vertical component at which the 2D tremor signal is then extracted. The procedure can be summarized by the following equation

$$p_{t2d}(t) = \left(R_{2d}(t) \left[p_{t3d}(t) - \frac{p_{t3d}(t) \cdot v_v(t)}{\|v_v(t)\|_2^2} v_v(t) \right] \right)_{1:2}, \quad (8.8)$$

where $v_v(t)$ is the velocity of the voluntary movement calculated as the numerical gradient of $p_v(t)$ and $\|\cdot\|_2$ denotes the Euclidean norm. The rotation matrix $R_{2d}(t)$ consists of a sequence of multiple rotation steps

$$\begin{aligned} R_{2d}(t) &= R_{\text{align}}(q(1))R_{\text{align}}(q(2)) \dots R_{\text{align}}(q(t)) \\ &= \prod_{k=1}^t R_{\text{align}}(q(k)), \end{aligned} \quad (8.9)$$

where for $k > 1$, $R_{\text{align}}(q(k))$ is the rotation that aligns the normal of the k :th plane with the normal of the $k - 1$:th plane and $R_{\text{align}}(q(1))$ is the rotation that rotates the first plane into the plane with zero vertical component. Here, $R_{\text{align}}(q(k))$ is parameterized by the unit quaternion $q(k)$, a unit vector in \mathbb{R}^4 , that is computed as

$$q(k) = \frac{\tilde{q}(k)}{\|\tilde{q}(k)\|_2}, \quad (8.10)$$

$$\tilde{q}(k) = \begin{bmatrix} n(k-1) \cdot n(k) \\ n(k-1) \times n(k) \end{bmatrix}, \quad (8.11)$$

where $\tilde{q}(k)$ is the non-normalized quaternion that describes the shortest arc rotation from the unit vector $n(k)$ to the unit vector $n(k - 1)$, \cdot denotes the

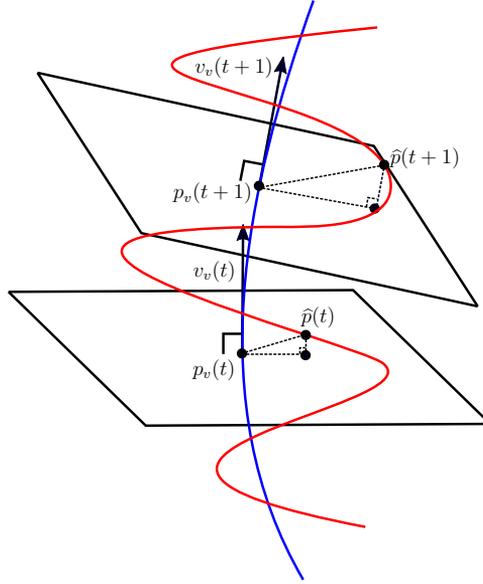


Figure 8.7: Projection of the 3D tremor signal onto planes perpendicular to the gradient (velocity) of the voluntary movement.

dot product, and \times denotes the cross product of two vectors. In the present case, the vectors $n(k)$ correspond to the normal vectors of the planes to be aligned

$$n(k) = \begin{cases} \frac{v_v(k)}{\|v_v(k)\|_2}, & 1 \leq k \leq N \\ \left[\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0 \right]^\top, & k = 0. \end{cases} \quad (8.12)$$

An arbitrary rotation matrix $R \in \text{SO}(3)$ can be parameterized by a unit quaternion q as

$$R(q) = \begin{bmatrix} 2q_1^2 + 2q_2^2 - 1 & 2(q_2q_3 + q_1q_4) & 2(q_2q_4 - q_1q_3) \\ 2(q_2q_3 - q_1q_4) & 2q_1^2 + 2q_3^2 - 1 & 2(q_3q_4 + q_1q_2) \\ 2(q_2q_4 + q_1q_3) & 2(q_3q_4 - q_1q_2) & 2q_1^2 + 2q_4^2 - 1 \end{bmatrix} \quad (8.13)$$

$$q = [q_1 \quad q_2 \quad q_3 \quad q_4]^\top, \quad (8.14)$$

which form yields the rotation matrices $R_{\text{align}}(q(k))$ in (8.9).

8.5 Frequency estimation using extreme points of curvature

The frequency of the tremor signal can be characterized by the timing of the extreme (turning) points of the trajectory. The time between three

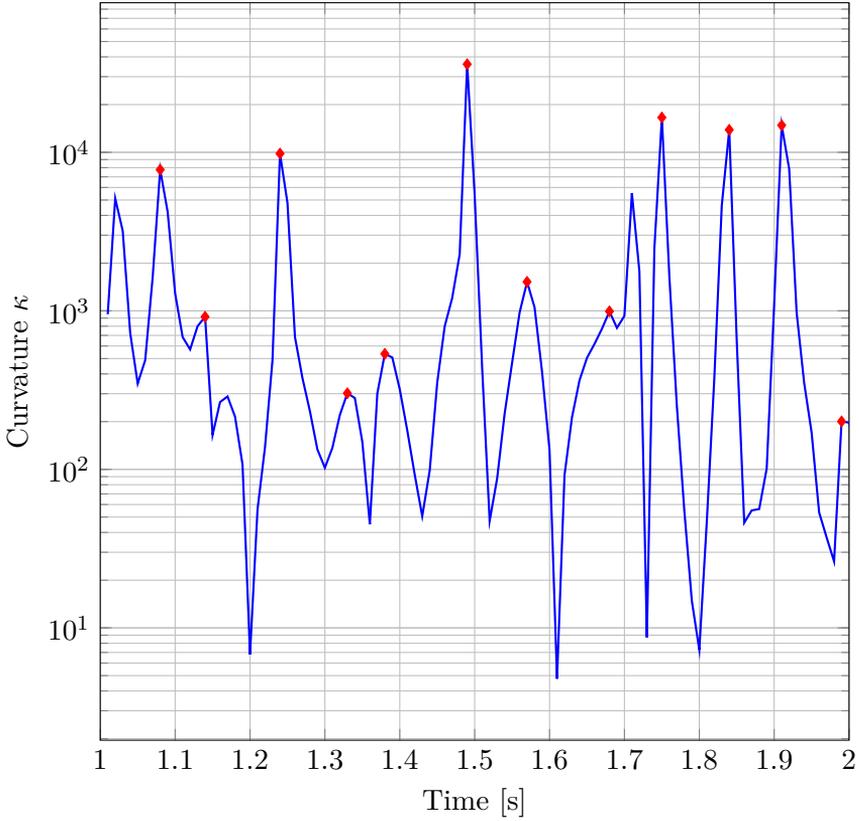


Figure 8.8: The curvature of a one second segment of a tremor signal with detected extreme points, $\kappa(\tau_k)$, indicated by the red markers.

consecutive extreme points is a measure of the tremor period, or inverse of its frequency. Extreme points can be detected in a number of ways. An approach identifying the extreme points of a trajectory as the local maxima of the curvature explored in Dimitrakopoulos et al. (2017) is also adopted in this paper.

The curvature of any twice differentiable trajectory γ in an Euclidean space can be computed as

$$\kappa = \frac{\sqrt{\|\gamma'\|_2^2 \|\gamma''\|_2^2 - (\gamma' \cdot \gamma'')^2}}{\|\gamma'\|_2^3}, \quad (8.15)$$

where γ' is the first derivative, or velocity, and γ'' is the second derivative, or acceleration of the trajectory. In the case at hand, γ corresponds to the tremor signal and γ', γ'' are obtained through numeric differentiation. Since the tremor trajectory is a function of time, so is its curvature, i.e. $\kappa = \kappa(t)$.

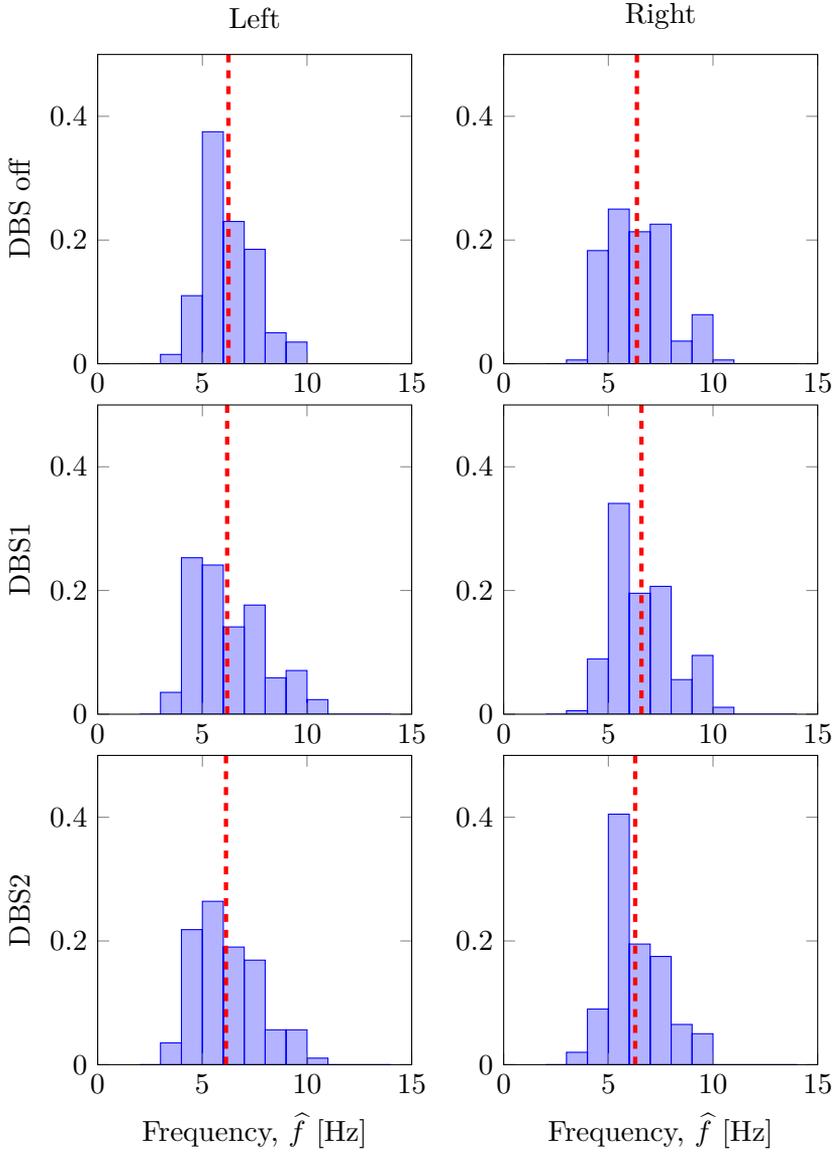


Figure 8.9: Normalized histograms of the frequency estimates based on the time between extreme points $\kappa(\tau_k)$ in the curvature of the tremor signal. The red dashed lines show the means of the estimated frequencies.

Once $\kappa(t)$ is computed for a particular tremor signal, the local maxima of the function are identified at the time instants $\tau_k \in \{\tau_1, \tau_2, \dots, \tau_L\}$. A sliding window of $2l+1$ samples is used to identify all time instants τ_k . If the sample in the middle has the largest value among all samples in the window

it is identified as a local maxima. An example of curvature as a function of time with detected extreme points for a tremor signal is shown in Fig. 8.8. For the tremor signal at hand $l = 4$ is chosen and there are at least eight samples between three consecutive extreme points. The sampling rate of the sensors is 100Hz, so 8 samples correspond to a frequency of 12.5Hz, which is outside of the assumed tremor band. A general rule for sensors with sample frequency f_s is to select l such that

$$\frac{f_s}{2l} > 12 \Leftrightarrow \frac{f_s}{24} > l. \quad (8.16)$$

The frequency estimates are then computed as

$$\hat{f}_k = \frac{1}{\tau_{k+2} - \tau_k}, \quad \forall k = 1, \dots, L - 2. \quad (8.17)$$

Normalized histograms over the estimated frequencies for the tremor signals obtained from the data in Fig. 8.1–Fig. 8.3 are depicted in Fig. 8.9.

8.6 Markov chain model of tremor severity

The tremor signal appears to exhibit variation in amplitude, direction and frequency. The tremor amplitude is what is really debilitating for a patient as it is a measure of how far the tremor drives the actual movement away from the voluntary one. Therefore, adopting a stochastic viewpoint on tremor amplitude variation, it can be modelled as a Markov chain. A similar approach has been applied before to model the center of pressure (COP) trajectories in human balance experiments (Hur et al. 2012; Rasku et al. 2008). The movement of the COP in quiet human standing is an effect of involuntary body sway similar to the involuntary movement associated with tremor. Following the approach of Hur et al. (2012), the invariant density of a data-driven Markov chain model is utilized to assess the severity of tremor.

Markov chain models are stochastic processes that describe how a state X_t of a dynamical system evolves over time. The defining property of Markov chains is that the probability of transitioning to the next state $X_{t+1} = x_{t+1}$ only depends on the current state $X_t = x_t$

$$\begin{aligned} p(X_{t+1} = x_{t+1} | X_t = x_t, X_{t-1} = x_{t-1}, \dots, X_1 = x_1) \\ = p(X_{t+1} = x_{t+1} | X_t = x_t). \end{aligned} \quad (8.18)$$

This property is known as the Markov property. The space of all possible states that the Markov chain may attain is called the state space and denoted by S , so that $x_{1:N} \in S$. In the application at hand, the state space is defined

as discrete levels of increasing symptom severity, which are determined by the amplitudes of the tremor signal

$$S = \{S_i \mid d_{i-1} \leq \|p_{t2d}(t)\|_2 \leq d_i\}, \quad i = 1, \dots, M, \quad (8.19)$$

$$d_i = ri, \quad r \in \mathbb{R}_{>0}, \quad (8.20)$$

where r is the radius of each state. Each state S_i corresponds to a set of points in \mathbb{R}^2 whose distance to the origin is bounded from below and above by the two circular disks with radii d_{i-1} and d_i , respectively. Fig. 8.10 illustrates part of a tremor signal trajectory evolving in the state-space.

The probability of transitioning from one state to another is encoded in the stochastic matrix $P \in \mathbb{R}^{M \times M}$, where each element

$$P_{ij} = p(X_{t+1} = S_j \mid X_t = S_i), \quad i, j \in \{1, \dots, M\}, \quad (8.21)$$

gives the probability of transitioning from the i :th state to the j :th state. For the tremor signal, the matrix P is determined by

$$P_{ij} = \frac{\sum_{t=1}^N \mathbb{1}(X_{t+1} = S_j) \mathbb{1}(X_t = S_i)}{\sum_{t=1}^N \mathbb{1}(X_t = S_i)}, \quad (8.22)$$

where $\mathbb{1}(\cdot)$ is the indicator function.

Let the probability of the state at time t be denoted by the row vector

$$p(X_t) = \left[p(X_t = S_1) \quad \dots \quad p(X_t = S_M) \right]. \quad (8.23)$$

Then it follows that

$$p(X_{t+1}) = p(X_t)P. \quad (8.24)$$

If a Markov chain satisfies the property

$$p(X_t = S_j \mid X_1 = S_i) = 1, \quad \forall i, j \in \{1, \dots, M\}, \quad (8.25)$$

for any positive finite integer t , it is said to be *irreducible* and *recurrent* (Norris 1998). This means that, being initialized at any state $X_1 \in S$ and simulated for a sufficiently long time, the Markov chain is able to reach any state in the state-space with probability 1. If a Markov chain satisfies (8.25), then (8.24) will converge to

$$\pi = \pi P, \quad (8.26)$$

as $t \rightarrow \infty$. The row vector π is thus a left eigenvector of P and is often referred to in the literature as the invariant or stationary distribution of the Markov chain, given that π is scaled such that all elements sum to one.

By selecting the state radius r for the tremor signals sufficiently large, Markov chain models that satisfy (8.25) can be constructed and thus the invariant distribution π be calculated.

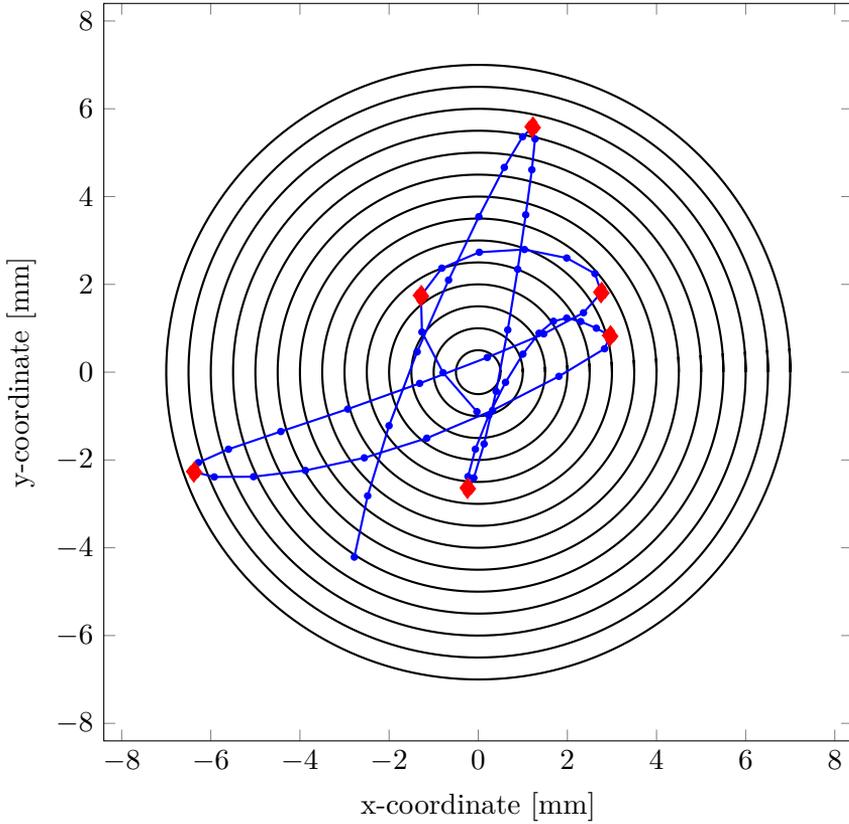


Figure 8.10: Shows a part of a tremor signal trajectory (blue) in a state-space defined as in (8.19) with $M = 14$ and $r = 0.5\text{mm}$. The black circles correspond to the upper bound of the radius of each state. Detected turning points are shown in red.

8.6.1 Tremor quantification parameters

From the obtained invariant distributions π , the following parameters to provide different measures of tremor severity can be extracted.

Mean distance

The average distance between the actual and voluntary movement is defined as

$$d_{\text{mean}} = r \sum_{i=1}^M i\pi_i. \tag{8.27}$$

A larger d_{mean} means that the tremor amplitudes tend to be higher thus driving the movement away from the intended voluntary trajectory.

Standard deviation distance

A measure of uncertainty in tremor amplitude defined as

$$d_{\text{std}} = \sqrt{\sum_{i=1}^M (r_i - d_{\text{mean}})^2 \pi_i}. \quad (8.28)$$

A larger d_{std} indicates that the tremor amplitudes show more variation over the data set.

95% Percentile distance

The distance from the voluntary movement that has a 95% probability of containing the tremor signal, defined as

$$d_{95\%} = r\bar{m}, \quad (8.29)$$

where

$$\begin{aligned} \bar{m} &= \min m \\ \text{subject to } \left(\sum_{i=1}^m \pi_i \right) &\geq 95\%. \end{aligned} \quad (8.30)$$

It is a measure of how heavy the tail of the amplitude distribution is, and how far the largest tremor amplitudes drive the movement away from the intended voluntary trajectory.

8.6.2 Results

Following the procedure described in Section 8.4, a tremor signal $p_{t2d}(t)$ was obtained for both hands and all three DBS settings from the data shown in Fig. 8.1–Fig. 8.3. A state-space model defined by (8.19) was used with a state radius $r = 0.5\text{mm}$ to construct the transition probability matrix as in (8.22). This choice of state radius was sufficient for the data to satisfy the conditions in (8.25). Invariant distributions π have been obtained and are plotted with respect to the respective upper state boundaries d_i in Fig. 8.11. The tremor quantification parameters described in Section 8.6.1 were computed for every π , see Table 8.3.

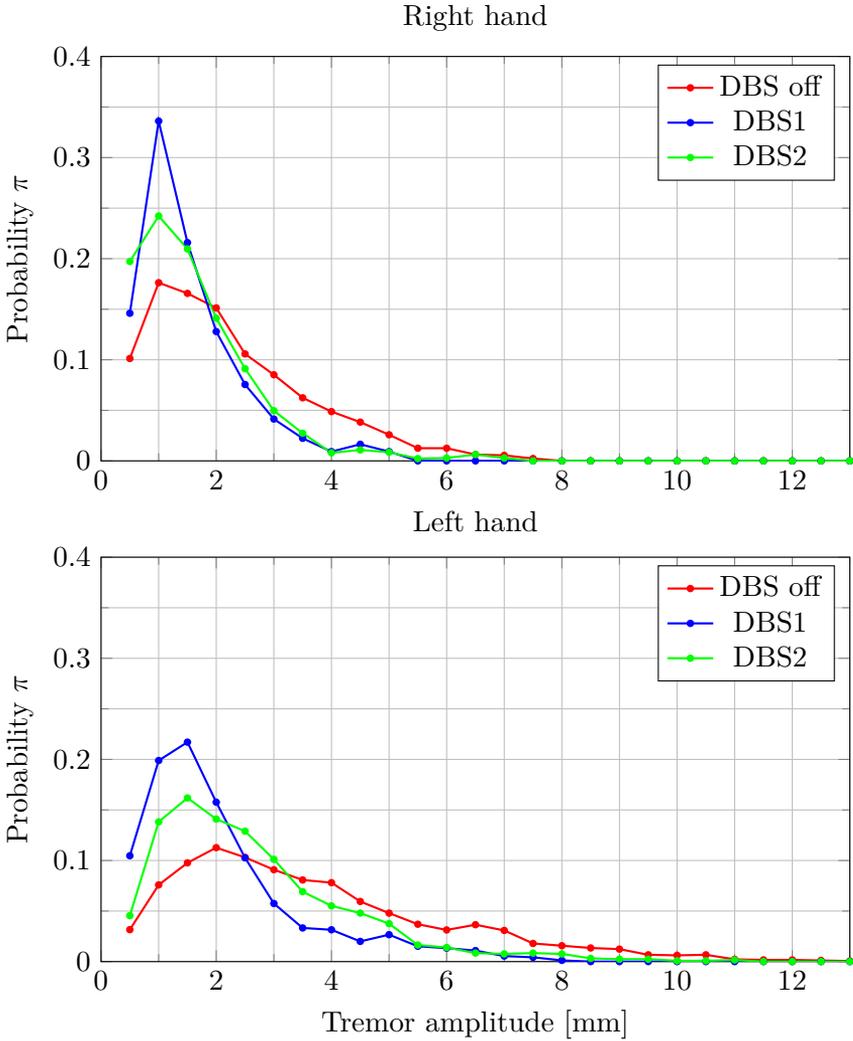


Figure 8.11: Shows the invariant distributions π computed for the different tremor signals. Each tick on the curves correspond to a specific state S_i and the associated amplitudes are the upper bounding radius of the respective state.

8.7 Discussion

The present study compares different ways to quantify tremor. Spectral analysis was used to estimate the tremor power contained in the raw accelerometer and gyroscope measurements and identify the dominant frequencies. The results summarized in Table 8.1 show that the lowest tremor power

Table 8.3: Tremor parameters extracted from invariant distributions shown in Fig. 8.11. The smallest value for each hand is highlighted in boldface.

Trial	Hand	d_{mean} [mm]	d_{std} [mm]	$d_{95\%}$ [mm]
Right	DBS off	2.3	1.4	4.5
	DBS1	1.5	0.9	4
	DBS2	1.6	1.0	3.5
Left	DBS off	3.7	2.3	8
	DBS1	2.1	1.4	6.5
	DBS2	2.7	1.7	6.5

across all sensor axes was obtained with the DBS2 setting for the right hand and with the DBS1 setting for the left hand. It can also be seen in Table 8.2 that the dominant frequencies vary from axis to axis, except for the data collected from the left hand with DBS off, which case yielded, typically for Parkinsonian tremor, a dominant frequency of $f_{\text{max}} = 5.86\text{Hz}$. It was also for DBS off that the largest tremor power among all axes was observed.

There is no doubt that it is possible to detect Parkinsonian tremor by means of the spectral analysis method. However, comparing the therapeutic effect of the stimulation settings DBS1 and DBS2 based on the same method is more difficult. Given the results of this study, a more natural and promising way to quantify tremor is by evaluating the deviation in position from a voluntary movement. This is similar to how physicians currently view tremor when assessing its severity. Section 8.4 describes a way to extract a relevant tremor signal from the accelerometer, gyroscope and magnetometer measurements collected from a smart phone.

In the obtained tremor signals, a time-varying behaviour of the tremor amplitudes and frequencies is observed. This effectively compromises the accuracy of severity estimates calculated through spectral analysis, where signal stationarity is assumed. While the tremor frequency is relevant to the problem of tremor classification, the tremor amplitude is the actual debilitating aspect and has to be rated in order to quantify the symptom severity.

The Markov chain model that is constructed using the tremor signals is a probabilistic model of tremor amplitude variation build for a particular snapshot of the data. The calculated invariant distribution π characterizes the asymptotic behavior of the Markov chain, i.e. it shows what the state distributions would be if the chain were simulated indefinitely long. This apparently in conflict with the non-stationary nature of tremor as it assumes

that the behavior captured by the Markov chain model persists in time. Therefore, the invariant distribution π has to be seen a stationary characteristics of a dynamical model, an abstraction that is necessary to compare the effect of different DBS settings. A straightforward interpretation of π is how likely it is for the observed tremor signal to drive the actual movement a certain distance away from the voluntary one.

The results presented in Fig 8.11 and Table 8.3 suggest that the tremor severity is similar in the right hand for both the DBS1 and DBS2 settings. However, for the left hand, it appears that the DBS1 setting produces a more favourable effect than DBS2 in terms of the mean and standard deviation in distance.

During the DBS programming session, when the data were collected, the physician eventually decided to keep using the DBS1 setting for the patient. This decision was mostly based on the patient's own judgment of what was perceived to be the best setting, since it was difficult for the physician to visually distinguish between the effects of the two DBS settings in question. The results of this study support the decision that was made in this particular case, but more data from different patients as well as data from the same patient under different stimulation settings will be needed to fully evaluate the efficacy of the method. Due to limited data at hand, this fell outside the scope of the present paper.

Since the proposed method is based on objective measurements, it serves as a tool for physicians in individualizing therapies that aim at alleviating tremor. Being implemented on a conventional smart phone equipped with a standard sensor platform, it is also unobtrusive, inexpensive, and highly flexible in use. Data collection may be performed essentially anywhere and incorporated in the regular day-to-day phone usage. For example, the movement used when collecting data for this study is similar to answering or making a phone call and can be easily recognized by an app.

8.8 Conclusion

A non-parametric time-domain tremor quantification method based on measurements collected from inertial sensors in a smart phone is studied and compared with a conventional technique exploiting spectral analysis.

The proposed method consists of first obtaining an estimate of the voluntary movement underlying the data set and then extracting a tremor signal as a deviation in position from voluntary movement. The tremor amplitude variation is characterized by the stationary state distribution of a Markov chain constructed from the tremor signal. The utility of the proposed approach is illustrated on patient data recorded during a DBS programming

session. The obtained tremor quantification parameters give support to the clinical decision made by the physician during the programming session based on the patient's judgment of the DBS settings effect.

Acknowledgements

The authors would like to thank Nils Wesslén, Dag Nyholm, and Ingela Nygren at the Uppsala University Hospital for the opportunity to participate in a DBS programming session and their assistance in collecting data for this study. The authors would like to thank Kjartan Halvorsen for providing useful feedback that helped to improve the quality of the manuscript.

References

- Amon, A. and F. Alesch (2017). ‘Systems for deep brain stimulation: review of technical features’. In: *Journal of Neural Transmission* 124.9, pp. 1083–1091.
- Asahi Kasei Microdevices Corporation (Sept. 2017). *3-axis Magnetic Sensor*. URL: <http://www.akm.com/akm/en/product/detail/0059/>.
- Cubo, R., M. Åström and A. Medvedev (Aug. 2016). ‘Model-based optimization of individualized deep brain stimulation therapy’. In: *IEEE Design and Test* 33.4. Special issue, pp. 74–78.
- Dai, H., P. Zhang and T. C. Lueth (2015). ‘Quantitative assessment of Parkinsonian tremor based on an inertial measurement unit’. In: *Sensors* 15.10, pp. 25055–25071.
- Dimitrakopoulos, K., C. Ellmer, M. Lindström and A. Medvedev (2017). ‘Tremor Quantification through Event-based Movement Trajectory Modeling’. In: *the 2017 IEEE Conference on Control Technology and Applications*. Kohala Coast, HA.
- Hur, P., K. A. Shorter, P. G. Mehta and E. T. Hsiao-Wecksler (Apr. 2012). ‘Invariant Density Analysis: Modeling and Analysis of the Postural Control System Using Markov Chains’. In: *IEEE Transactions on Biomedical Engineering* 59.4, pp. 1094–1100.
- InvenSense (Sept. 2017). *6 Axis MotionTracking*. URL: <http://www.invensense.com/products/motion-tracking/6-axis/>.
- Johansson, D. et al. (Jan. 2018). ‘Individualization of levodopa treatment using a microtablet dispenser and ambulatory accelerometry’. In: *CNS Neuroscience & Therapeutics*.
- Kok, M. (2016). ‘Probabilistic modeling for sensor fusion with inertial measurements’. PhD thesis. Linköping University.
- Medvedev, A., F. Olsson and T. Wigren (2017). ‘Tremor quantification through data-driven nonlinear system modeling’. In: *Decision and Control (CDC), 2017 IEEE 56th Annual Conference on*. (Melbourne, Australia). IEEE, pp. 5943–5948.
- Norris, J. R. (1998). *Markov chains*. 2nd. Cambridge, UK: Cambridge University Press.
- Nyholm, D. (Sept. 2012). ‘Duodopa® treatment for advanced Parkinson’s disease: a review of efficacy and safety’. In: *Parkinsonism Relat Disord* 18.8, pp. 916–929.
- Olsson, F. and A. Medvedev (2018). *Non-parametric Time-domain Tremor Quantification with Smart Phone for Therapy Individualization*. Submitted for publication.

- Rasku, J., M. Juhola, T. Tossavainen, I. Pyykkö and E. Toppila (July 2008). ‘Modelling stabilograms with hidden Markov models’. In: *Journal of Medical Engineering & Technology* 32.4, pp. 273–283.
- Savitzky, A. and M. Golay (1964). ‘Smoothing and Differentiation of Data by Simplified Least Squares Procedures’. In: *Anal. Chem.* 36.8, pp. 1627–1639.
- Senek, M. et al. (2017). ‘Levodopa/carbidopa microtablets in Parkinson’s disease: a study of pharmacokinetics and blinded motor assessment’. In: *European Journal of Clinical Pharmacology* 73.5, pp. 563–571.
- Senova, S., D. Querlioz, C. Thiriez, P. Jedynak, B. Jarraya and S. Palfi (2015). ‘Using the accelerometers integrated in smartphones to evaluate essential tremor’. In: *Stereotactic and functional neurosurgery* 93.2, pp. 94–101.
- Stergiou, N. and L. M. Decker (2011). ‘Human movement variability, nonlinear dynamics, and pathology: is there a connection?’ In: *Human movement science* 30.5, pp. 869–888.
- Stoica, P. and R. L. Moses (1997). *Introduction to Spectral Analysis*. 1st Edition. Prentice Hall.
- The International Parkinson and Movement Disorder Society (2016). *MDS Rating Scales*. URL: <http://www.movementdisorders.org/MDS/Education/Rating-Scales.htm> (visited on 23/11/2016).
- Thomas, I., J. Westin, M. Alam, F. Bergquist, D. Nyholm, M. Senek and M. Memedi (2017). ‘A treatment-response index from wearable sensors for quantifying Parkinson’s disease motor states’. In: *IEEE Journal of Biomedical and Health Informatics* PP.99, pp. 1–1.
- Westin, J., M. Dougherty, D. Nyholm and T. Groth (2010). ‘A home environment test battery for status assessment in patients with advanced Parkinson’s disease’. In: *Computer Methods and Programs in Biomedicine* 98.1, pp. 27–35.

Recent licentiate theses from the Department of Information Technology

- 2018-002** Tatiana Chistiakova: *Ammonium Based Aeration Control in Wastewater Treatment Plants - Modelling and Controller Design*
- 2018-001** Kim-Anh Tran: *Static Instruction Scheduling for High Performance on Energy-Efficient Processors*
- 2017-003** Oscar Samuelsson: *Fault Detection in Water Resource Recovery Facilities*
- 2017-002** Germán Ceballos: *Modeling the Interactions Between Tasks and the Memory System*
- 2017-001** Diana Yamalova: *Hybrid Observers for Systems with Intrinsic Pulse-Modulated Feedback*
- 2016-012** Peter Backeman: *New Techniques for Handling Quantifiers in Boolean and First-Order Logic*
- 2016-011** Andreas Svensson: *Learning Probabilistic Models of Dynamical Phenomena Using Particle Filters*
- 2016-010** Aleksandar Zeljić: *Approximations and Abstractions for Reasoning about Machine Arithmetic*
- 2016-009** Timofey Mukha: *Inflow Generation for Scale-Resolving Simulations of Turbulent Boundary Layers*
- 2016-008** Simon Sticko: *Towards Higher Order Immersed Finite Elements for the Wave Equation*
- 2016-007** Volkan Cambazoglou: *Protocol, Mobility and Adversary Models for the Verification of Security*
- 2016-006** Anton Axelsson: *Context: The Abstract Term for the Concrete*



UPPSALA
UNIVERSITET

Department of Information Technology, Uppsala University, Sweden