Universidad de Granada

Fachhochschule Münster University of Applied Sciences

Münster University of Applied Sciences
Department of Electrical Engineering and Computer Science

Bachelor Thesis

KALMAN FILTERING APPLIED TO ORIENTATION ESTIMATION IN HUMAN BODY MOTION ANALYSIS

Author:
Robin Weiß

Supervisors:
Prof. Dr.-Ing. Peter Glösekötter
Ph.D. Alberto Olivares Vicente
Prof. Dr. med. Kai Bötzel

A thesis submitted in partial fulfilment of the requirements for the degree of Bachelor of Science in Electrical Engineering

May 2015

STATEMENT OF AUTHORSHIP

I hereby certify that this bachelor thesis has been composed by myself, and describes my own work, unless otherwise acknowledged in the text. All references and verbatim extracts have been quoted, and all sources of information have been specifically acknowledged. It has not been accepted in any previous application for a degree.

Granada, 2$^{\text{th}}$ May 2015

Robin Weiß

PREFACE

This thesis was submitted in partial fulfilment of the requirements for the degree of Bachelor of Science in Electrical Engineering. It describes the implementation of a new Kalman filter based orientation algorithm to improve the estimation of orientation angles by means of inertial sensors.

I took part in the joint research project "Human Body Motion Analysis of Patients with Neurodegenerative Diseases by Means of Inertial Sensors" between the Research Centre for Information and Communications Technologies of the University of Granada (CITIC-UGR), Spain, and the Department of Neurology of the Klinikum Großhadern, which is part of the Ludwig Maximilian University of Munich, Germany. The goal of the overall project was to obtain several gait parameters by wearable inertial sensors and validate them against conventional methods such as force plates and cameras in combination with visual markers. Physicians and medical researchers are interested in this approach of body motion analysis, as unobtrusive wearable sensors can assist the diagnosis of neurodegenerative diseases such as Parkinson's. Prior to this thesis I completed a three-months internship at the CITIC-UGR in which I worked on the synchronisation of a force measuring plate and inertial sensors within the above-mentioned project.

ABSTRACT

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Etiam lobortis facilisis sem. Nullam nec mi et neque pharetra sollicitudin. Praesent imperdiet mi nec ante. Donec ullamcorper, felis non sodales commodo, lectus velit ultrices augue, a dignissim nibh lectus placerat pede. Vivamus nunc nunc, molestie ut, ultricies vel, semper in, velit. Ut porttitor. Praesent in sapien. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Duis fringilla tristique neque. Sed interdum libero ut metus. Pellentesque placerat. Nam rutrum augue a leo. Morbi sed elit sit amet ante lobortis sollicitudin. Praesent blandit blandit mauris. Praesent lectus tellus, aliquet aliquam, luctus a, egestas a, turpis. Mauris lacinia lorem sit amet ipsum. Nunc quis urna dictum turpis accumsan semper.

CONTENTS

LIST OF FIGURES

ABBREVIATIONS

ARW	Angle random walk
CITIC-UGR	Research Centre for Information and Communications Technologies of the University of Granada
EKF	Extended Kalman filter
IMU	Inertial measurement unit
LTSD	Long term spectral detector
MARG sensors	Magnetic, angular rate, and gravity sensors
MEMS	Microelectromechanical systems
MIMU	Magnetic inertial measurement unit
Pendubot	Pendulum robot
RMSE	Root-mean-square error

NOTATION

$\mathbf{a}$	Acceleration vector, $\mathbf{a} \in \mathbb{R}^3$
$\mathbf{B}$	Control matrix that relates the control input to the state $\mathbf{x}$
$\mathbf{C}_{bn}$	Transformation matrix transforming a position vector from the body frame to the world frame
$\mathbf{C}_{nb}$	Transformation matrix transforming a position vector from the world frame to the body frame
d	Desired filter response of a linear discrete-time filter
e	Error signal of a linear discrete-time filter
$\mathbf{f}$	Force vector, $\mathbf{f} \in \mathbb{R}^3$
$\mathbf{h}$	Functional denoting the *non-linear* measurement matrix function of a discrete dynamical system
$\mathbf{H}$	Measurement sensitivity matrix defining the linear relationship between the state of the dynamical system and the measurements that can be made
k	Discrete time normalised to sampling interval (sample number), $k \in \mathbb{N}^0$
K	Weighting factor
$\mathbf{K}$	Kalman gain matrix
m	Mass

n	Discrete time
$\mathbf{\Omega}_{\mathbf{E}\rightarrow\mathbf{E}}$	Function that transforms a position vector $\mathbf{b}$ in the vector space $\mathbf{E}$ into the vector $\mathbf{b}^{'}$ in the vector space $\mathbf{E}^{'}$
$\mathbf{P}$	Covariance matrix of state estimation uncertainty
$\mathbf{p}$	Position vector in a three-dimensional vector space
ϕ	Roll angle that determines the rotation around the x-axis
$\boldsymbol{\phi}$	Functional denoting the *non-linear* transition matrix function of a discrete dynamical system
$\mathbf{\Phi}$	State transition matrix of a discrete linear dynamical system
ψ	Yaw angle that determines the rotation around the z-axis
$\mathbf{Q}$	Covariance matrix of process noise in the system state dynamics
$\mathbf{R}$	Covariance matrix of observational (measurement) uncertainty
μ	Mean value of conditional probability density
σ^2	Variance
σ	Standard deviation
t	Continuous time
$\mathbf{T}$	Transformation matrix
θ	Pitch angle that determines the rotation around the y-axis
u	Nominal velocity
$\mathbf{v}$	Measurement noise vector
w	Noise term
$w_0, w_1, w_2, \ldots$	Impulse response of a linear discrete-time filter

$\mathbf{w}$	Process noise vector
x	One-dimensional location
$\hat{x}$	Estimate of x
x, y, z	Axes of the fixed world frame
X, Y, Z	Axes of the moving body frame
$\mathbf{x}$	State vector of a linear dynamical system
$\mathbf{x}_k$	The kth element of a sequence $\ldots, \mathbf{x}_{k-1}, \mathbf{x}_k, \mathbf{x}_{k+1}, \ldots$ of vectors
$\hat{\mathbf{x}}$	Estimate of the state vector of a linear dynamical system
$\hat{\mathbf{x}}_k^-$	A priori estimate of $\hat{\mathbf{x}}$, conditioned on all prior measurements except the one at time t_k
$\mathbf{y}$	Observation vector of a dynamical system
$y(0), y(1), y(2), \ldots$	Time series that serves as input to a linear discrete-time filter
z^{-1}	Unit-delay
$\mathbf{z}$	Vector of measured values

1 INTRODUCTION

Monitoring and assessment of human body motion, in particular the analysis of gait, has become an integral part of medical diagnosis, therapy techniques, and rehabilitation [8]. *Gait analysis* involves the measurement and assessment of quantitative parameters that characterise human locomotion. First research in this field was conducted in the late 19$^{\text{th}}$ century [8]. The quantitative data enable physicians to diagnose a variety of medical conditions, validate treatment success, set goals in rehabilitation and regularly alter them when necessary. However, standard gait analysis based on multi-camera motion capture systems and force platforms require specialised gait laboratories, expensive equipment, and lengthy setup times. Moreover, the assessments of gait based on measurements performed in clinical settings might not be truly representative [9].

Unobtrusive wearable sensors mitigate the aforementioned limitations. Low cost sensors have been employed in clinical and home environments to constantly monitor the movements of patients [10]. The progressive miniaturisation of inertial and magnetic field sensors has made them more acceptable to patients and has consequently lead to an increasingly pervasive adoption for medical applications [11], especially in the truly representative home environment. Among others, wearable inertial and magnetic sensors are used to assess *gait kinematics*. An extensive description is presented in Section 1.4.2. *Kinematics* is a branch of classical mechanics, which is concerned with motion of objects without reference to the forces causing the motion. Position, velocity and acceleration are of particular interest in kinematics.

Determining the position of the legs is essential in gait analysis. The position, i. e. the orientation, can be estimated from inertial data. For the application in health care accurate orientation estimates are crucial. A high degree of precision based on data from miniaturised sensors necessitates adequate signal processing, in order to mitigate the influence of disruptive factors, such as bias instability and noise, among others. The signal processing of inertial and magnetic data encompasses calibration, adaptive filtering, and sensor fusion. The latter two are a part of this work.

1.1 MOTIVATION

Gait analysis provides a powerful means to derive diagnostic information about the functioning of the musculoskeletal, vestibular, and central and peripheral nervous system [12]. Accurate orientation estimation of the extremities by means of wearable inertial and magnetic field sensors allows objective assessment of human gait and can therefore benefit medical applications without the aforementioned constraints of camera based motion capture systems. A more reliable and more precise orientation estimation would enable an even more informative gait analysis. A multitude of applications in the medical field would profit from such an enhanced gait analysis [13]. The direct relation to health care and the resulting possibility to improve the quality of life of many patients was the motivation for this thesis.

1.2 GOALS

The goal of this thesis was implementing a new Kalman filter based orientation algorithm proposed by Bennett et al. in [5]. Thus the estimation of orientation angles of the human leg by means of inertial and magnetic sensors should be improved. The filter algorithm should be implemented using MATLAB® and validated against existing algorithms by comparing their respective root-mean-square error (RMSE). An existing system for human body motion analysis based on wearable sensors was available, so that no new hardware had to be developed to gather the movement data. A detailed description of the so-called GaitWatch system is found at the end of Chapter 2.

1.3 METHODOLOGY

This document presents my work within the overall project in a chronological order. Subsequent to the previous introductory overview of the topic and the definition of the project objectives, this chapter ends with a description of the state of the art. To accomplish the tasks defined in the previous section, I had to acquire knowledge regarding various subjects. Chapters 2 to 4 outline the necessary fundamentals of MARG sensors, orientation estimation and digital filters, respectively. This enables comprehension of the overall project, even for readers that are not familiar with some of the subjects. Those readers are referred to Chapters 2 to 4 at this point, before reading the state of the art. The actual implementation of the Kalman filter, including a prior theoretical design is given in Chapter 5. This chapter also encompasses the experimental setup, the results and a discussion of the latter. Finally, Chapter 6 covers conclusions and future work.

I implemented the filter algorithm in the numerical computing environment MATLAB®. As additional means to communicate with my supervisor and in order to enable him to follow the progress of my work at any time we used Pivotal Tracker, a tool for agile project management, and GitHub, a repository hosting service based on the distributed version control system Git. This thesis was written in LaTeX.

1.4 STATE OF THE ART

There are several research works in the literature dealing with orientation estimation by means of inertial sensors. Kalman filters have been used successfully to improve the estimation of orientation angles from inertial data. The state of the art at the commencement of the project is described below. Subsequently, applications of wearable inertial sensors in health care and current attempts to revolutionise medical research assisted by those sensors are presented.

1.4.1 *Kalman Filtering in Orientation Estimation*

Considering the fact that inertial and magnetic field sensors are used to establish objective body motion parameters that affect medical diagnosis, therapy, and rehabilitation, the necessity of a high level of accuracy becomes obvious. In order to obtain precise orientation estimates from sensor data it is essential to mitigate the effects of measurement noise and to combine the advantages of different sensors through sensor fusion. Therefore, a wide variety of Kalman filter algorithms have been developed in the literature. It is common practice to fuse accelerometer and gyroscope measurements to mitigate their respective drawbacks and thus obtain more accurate orientation estimates.

Luinge et al. [14] alleged that the gravitational component of the acceleration signal has a greater magnitude than the component caused by motion for many human movements. They estimated the tilt angle, which is defined as the angle between the sensor axes and the vertical. The separate estimates from an accelerometer and a gyroscopes were fused with a Kalman filter. To test their method they moved the sensors around by hand for 30 seconds and then put it in a known position. They orientation obtained by integrating the angular rate only served as a reference. They concluded that a fusion of accelerometer and gyroscope signals accounts for a considerable improvement of the orientation estimation. This approach lacks of dynamical comparison since it only compares the errors at specific static positions.

Due to human motion intensity usually being subject to change, Olivares Vicente implemented a *gated Kalman filter* in [15]. They modelled linear acceleration during intense motion as noise and improved the performance of the Kalman filter by dynamically adjusting the variance of both the process and measurement noise, according to the motion intensity. Therefore, they applied a long term spectral detector (LTSD) and set the variance between two predefined values. Then, the gated Kalman filter fused information from the accelerometer and the gyroscope signals. With this method they improved the adapting capability of the filter and consequently the precision of the orientation estimation.

Bennett et al. demonstrated in [5] that accelerometer angle estimates are inaccurate for typical motions of the leg. They affirmed the need to decouple the acceleration due to motion from the acceleration due to gravity, since the former cannot be neglected during fast motions. Therefore, they deployed a *kinematic model* of the leg to determine the acceleration that occurs due to motion and corrected the acceleration signal accordingly. An extended Kalman filter fused the corrected acceleration data with measurements of a gyroscope. This method improved upon the raw acceleration method during motion and at rest by an 83% smaller RMSE. Their proposed approach is the foundation of the filter algorithm implemented in Section 5.2.

1.4.2 *Wearable Sensors in Health Care*

Inertial sensors can be found in smart phones, fitness trackers, and other wearable devices, among others. With increasing capability of body sensor networks and wearable computing, they have become prevalent in research environments for estimation and tracking of human body motion [5]. They are used in activity monitoring [16–18], rehabilitation [19, 20], sports training [18, 21], and localisation [22, 23].

Many neurodegenerative diseases such as, for instance, Parkinson's disease, impair stable stance and gait and reduce the patient's mobility. Thus, they diminish the quality of life significantly. *Parkinson's disease* is a movement disorder that is characterised by marked slow movements, tremors, and unstable posture. Especially in advanced stages of the disease many patients exhibit an episodic, brief inability to step, which delays gait initiation or interrupts ongoing gait. This phenomenon is called freezing of gait. With the progression of the disease, Parkinson's patients are increasingly dependent on help from others to accomplish daily tasks. One of the most reliable diagnostic criterion of the disease is gait [8]. Hence, wearable motion sensors have been used successfully to classify the severity of the disease objectively [24–26].

Stroke patients, who regained their walking ability, need to undergo rehabilitation to recover their independent mobility. Ambulatory gait analysis provides a means to assess the function of the lower extremities of hemiparetic post-stroke patients and follow the progress of rehabilitation [8, 27]. In addition, the presence of neurologic gait abnormalities is used as a significant predictor of the risk of development of dementia [28]. Also, emergency falls of elderly people can be detected [29–31].

1.4.3 *The Impact of Wearable Sensors on Medical Research*

The fact that most of the current smartphones come equipped with inertial and magnetic sensors and powerful processor has opened up new possibilities for medical research. The number of smartphones across the globe is predicted to surpass two billion in 2016 [32]. Recently, large companies have developed software that takes advantage of the associated potential to revolutionise medical research.

In August 2014, the Michael J. Fox Foundation for Parkinson's Research and Intel Corporation [33] announced a collaborative research study on objective measurement of Parkinson symptoms. They aim to collect movement data of thousands of patients twenty-four seven at over 300 samples per second by means of unobtrusive wearable devices and store them on a big data analytics platform. The data platform, deployed on a cloud infrastructure, supports an analytics application that processes the data and detects changes in real time. Thus, by detecting anomalies, the progression of the disease can be measured objectively without the need for scientists to focus on the underlying computing technologies. Physicians and researchers are intended to have access to the data as well as be able to submit their own anonymised patient data for analysis. According to [33] the correlation of data that quantifies symptoms such as slowness of movement, tremor, and sleep quality with molecular data could advance drug development and provide a deeper insight into the clinical course of Parkinson's disease.

In March 2015, Apple Inc. announced a platform-independent, open source software framework called ResearchKit [34] that, amongst others, takes advantage of the MARG sensors in the iPhone to track movement of patients in daily life. Thus medical researchers obtain robust data with far more regularity than it was possible when patients complete tasks at hospitals or other research facilities in irregularly intervals. Moreover, according to Apple Inc. ResearchKit simplifies recruiting participants from all over the world, which results in a more varied study group that provides a more useful representation of the population. Together with the University of Rochester and Sage Bionetworks they announced the iPhone app 'mPower' [35], which measures balance and gait of Parkinson patients to help researchers better understand how various symptoms are connected to Parkinson's disease.

2

MARG SENSORS

Broadly speaking, sensors are physical devices used to detect changes in the output of a system. MARG sensors is a collective term for magnetic, angular rate, and gravitational sensors, which encompasses inertial sensors, as well as magnetic field sensors, also referred to as magnetometers. Inertial sensors itself generally fall into two categories: instruments sensing linear inertial displacement (accelerometers) and rotational inertial rate sensors (gyroscopes). They are applied in various contexts to quantify vibration, motion, and shock [1]. Particularly, the development of microelectromechanical systems (MEMS) opened up many medical applications as stated in Section 1.4.2. They have low manufacturing costs, small physical size, and low power consumption [1]. This chapter compiles the functional principles of MARG sensors and introduces inertial measurement units (IMUs) as a combination of those. At the end of the chapter the aforementioned GaitWatch device that was used to gather the movement data for the experiments is described in detail.

2.1 ACCELEROMETERS

Accelerometers measure the acceleration of an object relative to an inertial frame. Since acceleration cannot be sensed directly, the force exerted on a reference mass is measured. The resultant acceleration is computed according to Newton's second law $\mathbf{f} = m \cdot \mathbf{a}$, where $\mathbf{f} \in \mathbb{R}^3$ denotes the force vector, m the mass, and $\mathbf{a} \in \mathbb{R}^3$ the acceleration vector. Usually, an accelerometer consists of a small proof mass connected via a spring to the case of the instrument. The proof mass is displaced by Δx with respect to the case, when the instrument experiences a certain acceleration along its sensitive axis. Disregarding drag force, the displacement is directly proportional to the force exerted by the mass and thus to the acceleration. Therefore, by measuring the displacement of the proof mass

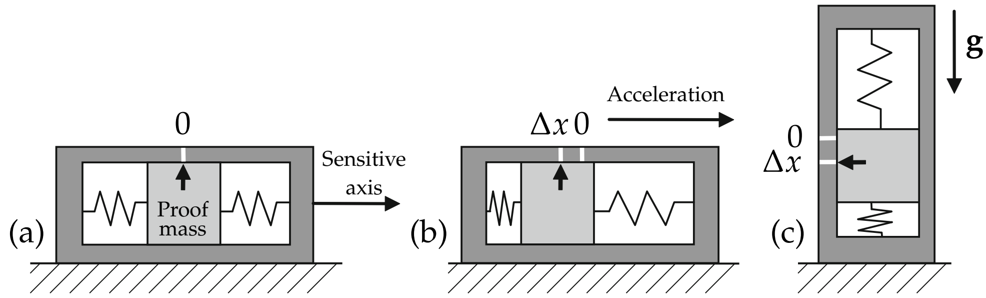

Figure 2.1: A mass-and-spring accelerometer under different conditions: (a) at rest or in uniform motion, (b) accelerating, and (c) at rest, being exposed to the gravity **g**, from [1].

the acceleration can be obtained. Figure 2.1 shows the displacement Δx of the mass in respect to the case of the instrument for three different conditions: (a) at rest or in uniform motion, (b) accelerating, and (c) at rest, being exposed to the gravity g. According to how the mass displacement is sensed, accelerometers can be classified as resistive, capacitive, and piezoelectric. Besides, there are surface acoustic wave, fibre optic, vibrating beam and solid-state MEMS accelerometers. To obtain a three-dimensional accelerometer, three single-axis accelerometers are mounted together. Although a mutually orthogonal mount is common practice, any non-coplanar arrangement is acceptable, as long as the angles between the sensitive axes are known. Nowadays most accelerometers are manufactured using MEMS technology, which was developed for the military and aerospace markets in the 1970s [1].

2.2 GYROSCOPES

Gyroscopes are used for measuring and maintaining angular orientation. In essence, based on two different physical principles, namely the Sagnac and Coriolis effect, gyroscopes sense angular velocity, which is why they are also referred to as angular velocity sensors or angular rate sensors. By integrating the angular velocity the rotation angle can obtained. Here we will only elaborate on the working principle of vibrating gyroscopes, since they are utilised in the GaitWatch device. Armenise et al. give a comprehensive overview of current gyroscope technologies in [2].

Coriolis vibratory gyroscopes, or vibrating gyros for short, sense angular velocity based on the effect of Coriolis force on a vibrating mass.

The Coriolis force is a fictitious force experienced by a mass m moving in a rotating reference frame. It can be calculated as: $\mathbf{f}_C = -2m(\boldsymbol{\omega} \times \mathbf{v})$, where $\mathbf{v}$ is the mass velocity in the rotating reference frame and $\boldsymbol{\omega}$ is the angular velocity of the reference frame. As seen in this equation the Coriolis force is only present when the mass varies its distance with respect to the spin axis. Otherwise, if $\boldsymbol{\omega}$ and $\mathbf{v}$ are parallel, the cross product becomes zero. The two degree-of-freedom spring-mass-damper system shown in Figure 2.2 serves as a simple model of a vibrating angular rate sensor. The mass m can move along the x and y-axis, respectively. The angular velocity around the z-axis is denoted with ω. The drive or primary oscillating mode, that is, the oscillation along x, is excited by the force F_x directed along the x-axis. The oscillation along y, called sense or secondary oscillating mode, is due to system rotation around the z-axis. D_x and D_y are the damping coefficients and k_x and k_y are the spring constants along the x and y-axis, respectively. Typically, the primary oscillating mode is excited by a sinusoidal force with an angular frequency close to the resonance frequency, so that $\Omega_x \cong \sqrt{k_x/m}$. Its amplitude is kept constant at a_x. As shown in [2],the amplitude of the sense mode is then given by

$$a_y = -\frac{2a_x\Omega_x\omega}{\sqrt{(\Omega_x^2 - \Omega_y^2)^2 + \Omega_x^2\Omega_y^2/Q_y^2}}, \tag{2.1}$$

where $\Omega_y = \sqrt{k_y/m}$ is the resonance frequency of the secondary resonator and $Q_y = \sqrt{mk_y}/D_y$ its quality factor. The amplitude a_y is directly proportional to the angular rate of the two degree-of-freedom spring-mass-damper system. Thus, ω can be estimated by measuring the amplitude of the oscillation along the y-axis.

Usually, vibrating gyroscopes are manufactured using MEMS technology. MEMS gyros are of low to medium accuracy [1], but due to their size they are ideally suited for medical applications.

2.3 MAGNETOMETERS

Magnetometers measure the strength and the direction of the magnetic field at a point in space. There are numerous techniques used to produce magnetic field sensors, which exploit a broad range of physical phenomena [36]. Lenz and Edelstein give a complete survey of common technologies used for magnetic field sensing in [36]. Many MEMS magnetometers

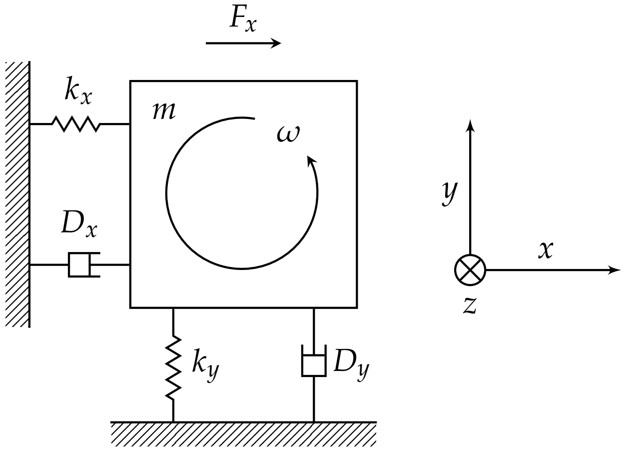

Figure 2.2: A simple model of a Coriolis vibratory gyroscope: Two degree-of-freedom spring-mass-damper system in a rotating reference frame, from [2].

sense mechanical motion of a MEMS structure due to Lorentz force and estimate the strength of the magnetic field according to the displacement. When an external magnetic field interacts with a current-carrying silicon MEMS structure the Lorentz force causes a displacement of this structure. Piezoresistive, capacitive, or optical sensing can be used to detect the displacement of the MEMS structure. MEMS Lorentz force magnetometers are free from hysteresis, require no specialised materials and can be monolithically integrated with other MEMS inertial sensors [37].

2.4 INERTIAL MEASUREMENT UNITS

Devices using a combination of accelerometers and gyroscopes to measure the orientation of a rigid body with up to six degrees of freedom are referred to as IMUs. If they include additional magnetometers they are termed magnetic inertial measurement units (MIMUs). The number of degrees of freedom states the number of independent motions, with respect to a reference frame, that are allowed to the body in space. MIMUs are portable and relatively inexpensive. They can be easily attached to the body and thus allow non-clinical longterm application. Their drawbacks are complex calibration procedures and drift behaviour over time, depending on intensity and duration of the measurement interval. Hence, in order to maintain a satisfactory degree of precision, periodical recomputation of the calibration parameters is required [15].

2.4.1 *The GaitWatch*

The above-mentioned GaitWatch device was designed to monitor the motion of patients while attached to the body. It was developed at the Department of Neurology of the Ludwig-Maximilians University in Munich, Germany, in association with the Department of Signal Theory, Telematics and Communications of the University of Granada, Spain. The system is composed of a set of embedded magnetic and inertial sensors wired to a box containing a microcontroller. This microcontroller is in charge of collecting data from the embedded box sensors, as well as from the external measurement units, and storing them on a memory card. The various units are placed at the patient's trunk, arms, thighs, and shanks as shown in Figure 2.3. The components of the three different kinds of subunits are described below:

- Type A – thighs and shanks:

  IMU Analog Combo Board with 5 Degrees of Freedom [38], containing an IDG500 biaxial gyroscope, from which only y-axis is actually used, with a measurement range of $\pm 500\,^{\circ}/\mathrm{s}$ [39] and a $\pm 3\,\mathrm{g}$ triaxial accelerometer, ADXL335 [40].

- Type B – arms:

  IDG500 biaxial gyroscope with a measurement range of $\pm 500\,^{\circ}/\mathrm{s}$ [39].

- Type C – trunk:

  ADXL345 triaxial accelerometer with a programmable measurement range of $\pm 2/\pm 4/\pm 8/\pm 16\,\mathrm{g}$ [41], IMU3000 triaxial gyroscope with a programmable measurement range of $\pm 250/\pm 500/\pm 1000/\pm 3000^{\circ}/\mathrm{s}$ [42], Micromag3 triaxial magnetometer with a measurement range of ± 11 Gauss [43], AL-XAVRB board containing an AVR ATxmega processor [44].

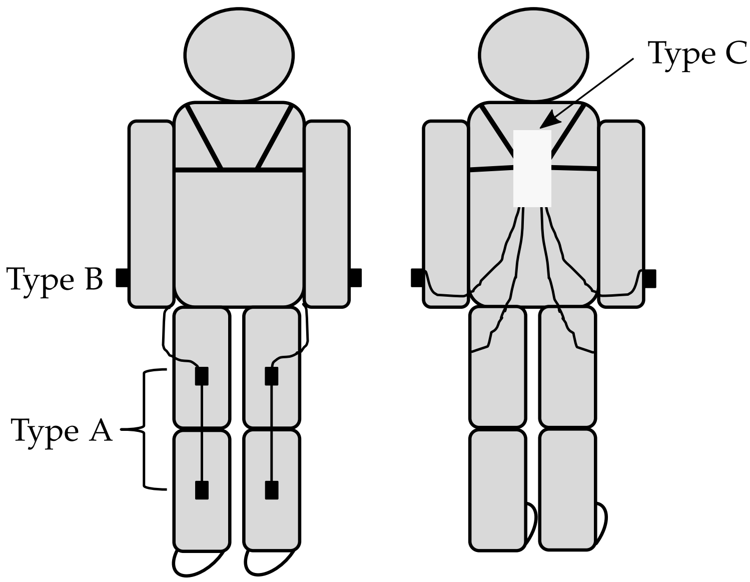

Figure 2.3: Placement of the GaitWatch components at the body, from [3].

3 ORIENTATION ESTIMATION

This chapter covers fundamentals of orientation estimation that are necessary for the implementation of the aforementioned system. In the motion monitoring field and especially in aircraft navigation the position of the coordinate frame of the body, with respect to a reference coordinate frame, is known as *attitude*, which is used as a synonym of orientation. There are different approaches to compute attitude estimates from magnetic and inertial data. This chapter explains their pros and cons and introduces sensor fusion as a means to mitigate the drawbacks of each approach. Beforehand, two main mathematical constructs used to express attitude – Euler angles and quaternions – are described.

3.1 EULER ANGLES

Euler angles are one of several ways to describe the orientation of an object and its associated *body frame* in three-dimensional Euclidean space, with respect to a *reference frame*. They represent a sequence of three elemental rotations about the axes of the coordinate system, defined as follows:

- The *roll* angle ϕ determines the rotation around the x-axis.
- The *pitch* angle θ determines the rotation around the y-axis.
- The *yaw* angle ψ determines the rotation around the z-axis.

Figure 3.1 depicts the rotation about the axes z, y', X by ψ, θ, ϕ, respectively, according to the Tait-Bryan convention. The colour blue indicates the orientation before and the colour red the orientation after the rotation. In contrast to *extrinsic rotations*, where the three elemental rotations may occur either about the axes of the original coordinate system, the

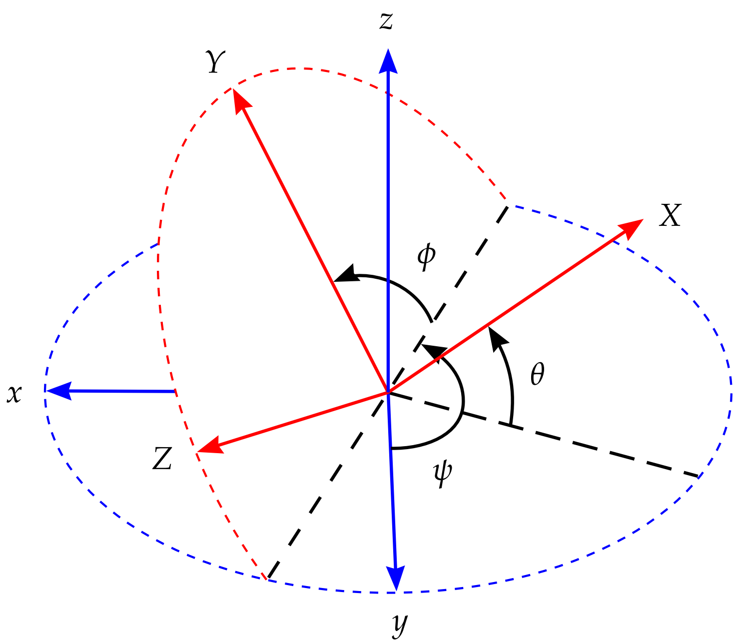

Figure 3.1: Representation of the body frame (red) with respect to the world frame (blue). The body frame was rotated, by the Euler angles ψ, θ, ϕ about the axes z, y', X, respectively. Adapted from [4].

Tait-Bryan rotations are *intrinsic rotations* that occur about the axes of the rotating coordinate system, which changes its orientation after each rotation.

Euler angles are a simple and intuitive means to represent rotations in three-dimensional space. However, for the above mentioned parameterisation they have singularities at values of $\theta = n\pi$, $n \in \mathbb{Z}$. At these points a rotation about the x-axis and the z-axis constitute the same motion, which results in the loss of one degree of freedom and makes changes in ϕ and ψ indistinguishable. This phenomenon is called *gimbal lock*. The usage of *quaternions* instead of Euler angles avoids the problem of gimbal locks.

3.1.1 *Transformation Matrix*

Coordinates representing a point in one coordinate system can be transformed to another. Such a transformations can be expressed as multiplication of a matrix with the coordinate vector that is to be transformed. Let $\mathbf{E}$ denote the orthonormal basis $\{x, y, z\} \in \mathbb{R}^3$ and let $\mathbf{E}'$ denote the orthonormal basis $\{X, Y, Z\} \in \mathbb{R}^3$. Furthermore, let $\mathbf{p}$ denote the position vector of an arbitrary point in three-dimensional Euclidean space. The coordinate transformation from $\mathbf{E}$ to $\mathbf{E}'$ is denoted $\boldsymbol{\Omega}_{\mathbf{E}\to\mathbf{E}'} : (p_1, p_2, p_3) \mapsto (p'_1, p'_2, p'_3)$. Then, the linear transformation from $\mathbf{p}$ to $\mathbf{p}'$ is given by

$$\mathbf{p}' = \boldsymbol{\Omega}_{\mathbf{E}\to\mathbf{E}'}(\mathbf{p}) = \mathbf{T}\mathbf{p}\,, \tag{3.1}$$

where $\mathbf{T}$ is the *transformation matrix*.

To transform the coordinate vector from the *world frame*, i. e. the reference frame, to the *body frame*, according to the common aerospace rotation sequence mentioned above and the Nort-East-Down system (NED), the transformation matrix $\mathbf{C}_{nb}$ is given by

$$\begin{aligned}
\mathbf{C}_{nb} &= \mathbf{T}_x(\phi)\mathbf{T}_y(\theta)\mathbf{T}_z(\psi) \\
&= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\
&= \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix}
\end{aligned} \tag{3.2}$$

Plugged in Equation 3.1 ($\mathbf{T} = \mathbf{C}_{nb}$) the left multiplications of the matrices $\mathbf{T}_x(\phi), \mathbf{T}_y(\theta), \mathbf{T}_z(\psi)$ to the vector $\mathbf{p}$ represent the coordinate rotations about the single axes x, y, z, respectively. That is, the orthogonal projection onto the axes of the coordinate system, which result from the respective two-dimensional rotation of ϕ, θ, ψ about the axes x, y, z, is computed. This is illustrated for a single rotation around the z-axis by the angle ψ in Figure 3.2. The matrices $\mathbf{T}_x(\phi), \mathbf{T}_y(\theta)$, and $\mathbf{T}_z(\psi)$ are also known as direction cosine matrices, since their elements are the cosines of the unsigned angles between the body-fixed axes and the axes of the world frame, as shown in [45]. The form stated here is already simplified. The matrix $\mathbf{C}_{bn}$ for transforming the coordinate vector from the body frame to the world frame is given by

$$\mathbf{C}_{bn} = \begin{bmatrix} \cos\theta\cos\psi & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi \\ \cos\theta\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{bmatrix} \tag{3.3}$$

Note that $\mathbf{C}_{bn} = \mathbf{C}_{nb}^T = \mathbf{C}_{nb}^{-1}$. Thus, $\mathbf{C}_{bn}$ and $\mathbf{C}_{nb}$ are orthogonal matrices so that $\mathbf{C}_{bn}\mathbf{C}_{nb} = \mathbf{I}$.

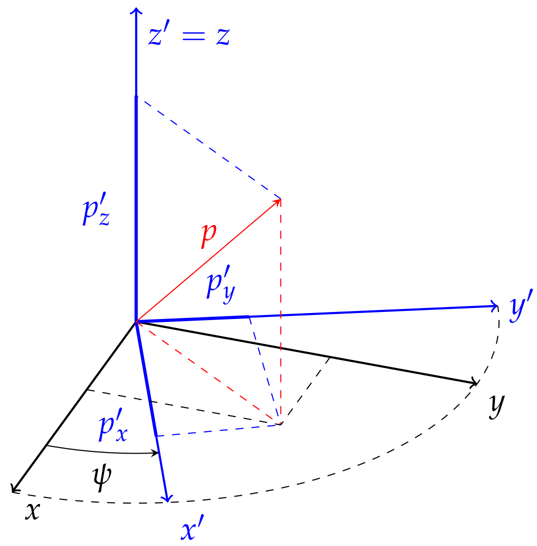

Figure 3.2: An exemplary coordinate rotation about the z-axis by an angle ψ, illustrating the orthogonal projection on the resulting axes x', y', z'.

3.2 QUATERNIONS

Quaternions are a number system that extends the complex numbers. They were first described by Irish mathematician William Rowan Hamilton in 1843. The set of quaternions is equal to $\mathbb{R}^4$ and denoted with $\mathbb{H}$. A quaternion $\mathbf{q} \in \mathbb{H}$ may be represented as

$$\mathbf{q} = q_0 + q_1 i + q_2 j + q_3 k = \begin{bmatrix} q_0, & q_1, & q_2, & q_3 \end{bmatrix}^T = \begin{bmatrix} q_0 \\ \mathbf{q}_{1:3} \end{bmatrix}, \tag{3.4}$$

whereby the following relations between the imaginary units i, j, k hold:

$$\begin{gathered} i^2 = j^2 = k^2 = ijk = -1\,, \\ \begin{aligned} ij &= k, & ji &= -k, \\ jk &= i, & kj &= -i, \\ ki &= j, & ik &= -j\,. \end{aligned} \end{gathered} \tag{3.5}$$

The *adjoint*, *norm* and *inverse* of a quaternion are defined as

$$\bar{\mathbf{q}} = \begin{bmatrix} q_0 \\ -\mathbf{q}_{1:3} \end{bmatrix}, \tag{3.6}$$

$$||\mathbf{q}|| = \sqrt{q_0^2 + q_1^2 + q_2^2 + q_3^2} \tag{3.7}$$

$$\mathbf{q}^{-1} = \frac{\bar{\mathbf{q}}}{||\mathbf{q}||} \tag{3.8}$$

The sum of two quaternions is defined as

$$\mathbf{q} + \mathbf{p} = \begin{bmatrix} q_0 + p_0 \\ \mathbf{q}_{1:3} + \mathbf{p}_{1:3} \end{bmatrix}. \tag{3.9}$$

Quaternion multiplication is defined as

$$\mathbf{q}\mathbf{p} = \begin{bmatrix} q_0 p_0 - \mathbf{q}_{1:3}^T \mathbf{p}_{1:3} \\ q_0 \mathbf{p}_{1:3} + p_0 \mathbf{q}_{1:3} - \mathbf{q}_{1:3} \times \mathbf{p}_{1:3} \end{bmatrix}, \tag{3.10}$$

and can be written as the second quaternion left-multiplied by a matrix-valued function Q of the first quaternion.

$$\mathbf{q}\mathbf{p} = Q(\mathbf{q})\mathbf{p} = \bar{Q}(\mathbf{p})\mathbf{q} \tag{3.11}$$

The *quaternion matrix function*, $Q : \mathbb{H} \to \mathbb{R}^{4\times 4}$ is defined by

$$Q(\mathbf{q}) = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & q_3 & -q_2 \\ q_2 & -q_3 & q_0 & q_1 \\ q_3 & q_2 & -q_1 & q_0 \end{bmatrix}, \tag{3.12}$$

and the *conjugate quaternion matrix function*, $\bar{Q} : \mathbb{H} \to \mathbb{R}^{4\times 4}$ is defined by

$$\bar{Q}(\mathbf{q}) = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix}. \tag{3.13}$$

Unit quaternions, i.e. quaternions with unity norm, can be used to represent rotations in three dimensional space. Let $\mathbf{z} \in \mathbb{R}^3$ be an arbitrary vector in the world frame and let $\mathbf{z}' \in \mathbb{R}^3$ be the same vector in the body frame. Furthermore, let $\mathbf{r}$ be a quaternion with $||\mathbf{r}|| = 1$. Then the following relations hold:

$$\begin{bmatrix} 0 \\ \mathbf{z}' \end{bmatrix} = \mathbf{r} \begin{bmatrix} 0 \\ \mathbf{z} \end{bmatrix} \mathbf{r}^{-1}, \tag{3.14}$$

$$\begin{bmatrix} 0 \\ \mathbf{z} \end{bmatrix} = \mathbf{r}^{-1} \begin{bmatrix} 0 \\ \mathbf{z}' \end{bmatrix} \mathbf{r}. \tag{3.15}$$

Sequences of rotations can thus be represented by products of quaternions. The quaternion for a single rotation by α about the axis $\boldsymbol{\epsilon} \in \mathbb{R}^3$, $||\boldsymbol{\epsilon}|| = 1$, is given by

$$\begin{aligned} \mathbf{r}_\epsilon(\alpha, \boldsymbol{\epsilon}) &= \left[\cos\tfrac{\alpha}{2}, \quad \boldsymbol{\epsilon}\sin\tfrac{\alpha}{2}\right]^T \\ &= \left[\cos\tfrac{\alpha}{2}, \quad \epsilon_1\sin\tfrac{\alpha}{2}, \quad \epsilon_2\sin\tfrac{\alpha}{2}, \quad \epsilon_3\sin\tfrac{\alpha}{2}\right]^T . \end{aligned} \tag{3.16}$$

The reverse mapping from quaternions to Euler angles is given by

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \text{atan2}((2(q_0q_1 + q_2q_3), 1 - 2(q_1^2 + q_2^2)) \\ \arcsin(2(q_0q_2 - q_1q_3)) \\ \text{atan2}(2(q_0q_3 + q_1q_2), 1 - 2(q_2^2 + q_3^2)) \end{bmatrix} . \tag{3.17}$$

Quaternions are less intuitive than Euler angles. Nevertheless, they are very popular in attitude representation, since they are mathematically elegant and don't suffer from singularities.

3.3 PROJECTION OF THE GRAVITY VECTOR

As described in Chapter 2, accelerometers measure the linear acceleration they experience. Under static or quasi-static (steady, linear motion) conditions, or at low acceleration it can be assumed that the measured acceleration is mainly that of gravity. Be means of simple trigonometric functions estimates for the pitch and the roll angle can be obtained. Since the gravity vector is perpendicular to the xy-plane and thus a rotation around the z-axis will not cause any variation in the sensed acceleration, the yaw angle cannot be obtained by this method. To solve this problem a three-dimensional magnetometer is used, which measures the variation of Earth's magnetic field while rotating around the z-axis.

When the accelerometer is motionless, its measurements will be directly related to the angle of the sensor relative to gravity, as depicted in Figure 3.3 (a). In that case θ is given by

$$\theta = \text{atan2}(A_y, A_x) . \tag{3.18}$$

However, when the sensor is in motion, in addition to the gravity, there are radial and tangential acceleration components due to motion, as depicted in Figure 3.3 (b). Ignoring these components will cause incorrect angle estimates.

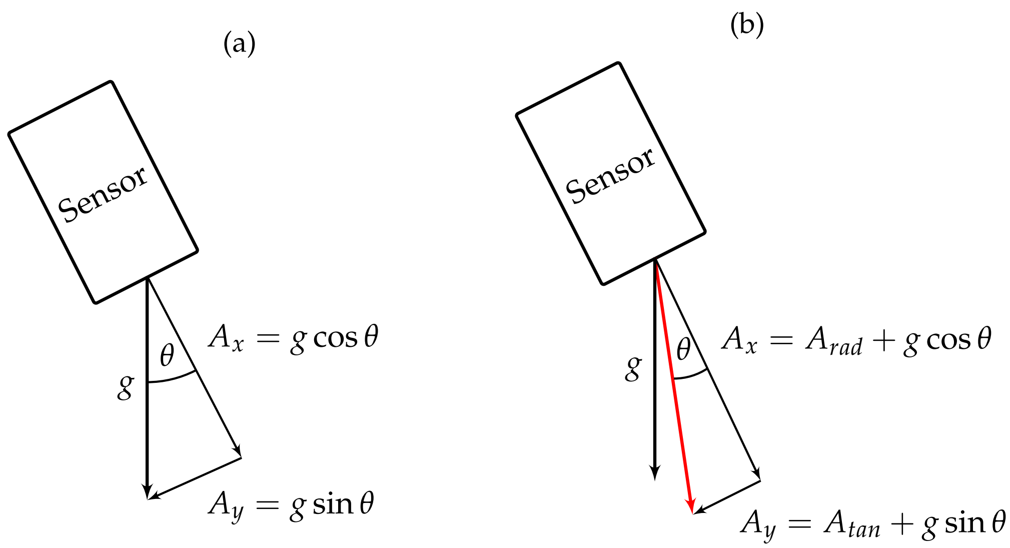

Figure 3.3: Acceleration seen by the sensor (b) with and (a) without motion, from [5].

3.4 INTEGRATION OF ANGULAR RATE

Another way to estimate the attitude of an object is the integration of the angular rate around the x, y and z-axis, respectively. Although this would theoretically lead to very accurate orientation estimates, they are impaired by angle random walk (ARW) and dynamical bias in practice. ARW is an effect caused by the integration of high-frequency, thermo-mechanical noise, which leads to a random additive angle in the orientation signal. An even greater impact than AWR has the gyroscopes dynamic bias, which has its origin in low-frequency flicker noise. Both effects cause a dramatical drift in the angle signal over time.

3.5 SENSOR FUSION

Since the projection of the gravity vector is only valid under static or quasi-static conditions, or at low acceleration, and the integration of the angular rate leads to non-reliable estimates due to ARW and dynamic bias, but is not affected by the intensity of motion, a means to combine the information of both sensors is desirable. The combination of information from multiple sensors to increase the overall precision of the estimation of a certain quantity of interest is termed *sensor fusion*. Raol [46] states the following advantages of sensor fusion:

- Robust functional and operational performance is given, in case of data loss from one sensor, due to redundancy provided by multiple sensors.
- Enhanced confidence in the results inferred from the measurement of one sensor, if they are confirmed by the measurement of another sensor.
- With sensor fusion an arbitrary fine time resolution of measurements is possible, whereas single sensors need a finite time to transmit measurements and so limit the frequency of measurements.
- One sensor might be, to some extent, better in a certain state of the measured process, e.g. low or high motion intensity in attitude estimation, and thus, by fusing multiple sensor signals, a satisfactory accuracy among all states of the process could be attained.

Sensor fusion can be realised by the use of a Kalman filter, which is described in detail in the next chapter.

4 DIGITAL FILTERS

Conceived in general terms, a filter is a physical device for removing unwanted components of a mixture. In the technical field a filter is a system designed to extract information from noisy measurements of a process. That is, the filter delivers an estimate of the variables of principal interest, which is why it may also be called an estimator. Filter theory is applied in diverse fields of science and technology, such as communications, radar, sonar, navigation, and biomedical engineering [6].

In contrast to *analogue filters* that consist of electronic circuits to attenuate unwanted frequencies in continuous-time signals and thus extract the useful signal, a *digital filter* is a set of mathematical operations applied to a discrete-time signal in order to extract information about the hidden quantity of interest. A *discrete-time* signal is a sequence of samples at equidistant time instants that represent the continuous-time signal with no loss, provided the sampling theorem is satisfied, according to which the sample frequency has to be greater than twice the highest frequency component of the continuous-time signal.

Digital filters can be classified as *linear* and *non-linear*. If the quantity at the output of the filter is a *linear* function of its input, that is, the filter function satisfies the superposition principle, the filter is said to be *linear*. Otherwise, the filter is *non-linear*.

4.1 THE FILTERING PROBLEM

Consider, as an example involving filter theory, the continuous-time dynamical system depicted in Figure 4.1. The desired state vector of the system, $\mathbf{x}(t)$, is usually hidden and can only be observed by indirect measurements $\mathbf{y}(t)$ that are a function of $\mathbf{x}(t)$ and subject to noise. Equally, the equation describing the evolution of the state $\mathbf{x}(t)$ is usually subject to system errors. These could be caused by, for instance, effects not ac-

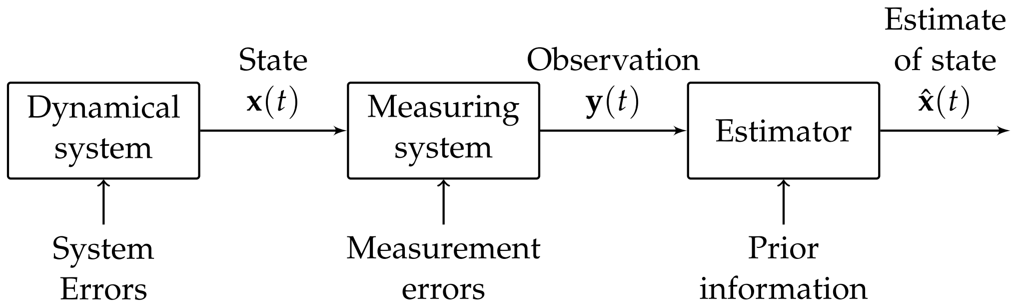

Figure 4.1: Block diagram depicting the components involved in state estimation, from [6].

counted for in the model. The dynamical system may be an aircraft in flight, in which case the elements of the state vector are constituted by its position and velocity. The measuring system may be a tracking radar producing the observation vector $\mathbf{y}(t)$ over an interval $[0, T]$. The requirement of the filter is to deliver a reliable estimate $\hat{\mathbf{x}}(t)$ of the actual state, by taking the measurement as well as prior information into account.

4.2 THE WIENER FILTER

A statistical criterion, according to which the performance of a filter can be measured, is the mean-squared error. Consider the linear dicrete-time filter with the impulse response $w_0, w_1, w_2, \dots$ depicted in Figure 4.2. At some discrete time n it produces an output designated by $\hat{x}(n)$, which provides an estimate of a desired response denoted by $d(n)$. According to Haykin [6] the essence of the filtering problem and the resulting requirement is summarised with the following statement:

> "Design a linear discrete-time filter whose output $\hat{x}(n)$ provides an estimate of the desired response $d(n)$, given a set of input samples $y(0), y(1), y(2), \dots$, such that the mean-square value of the estimation error $e(n)$, defined as the difference between the desired response $d(n)$ and the actual response $\hat{x}(n)$, is minimized."

Assume a stationary stochastic process with known statistical parameters as the mean and correlation functions of the useful signal and the unwanted additive noise. Then, the solution to this statistical optimisation problem is commonly known as the *Wiener filter*. Yet, since the

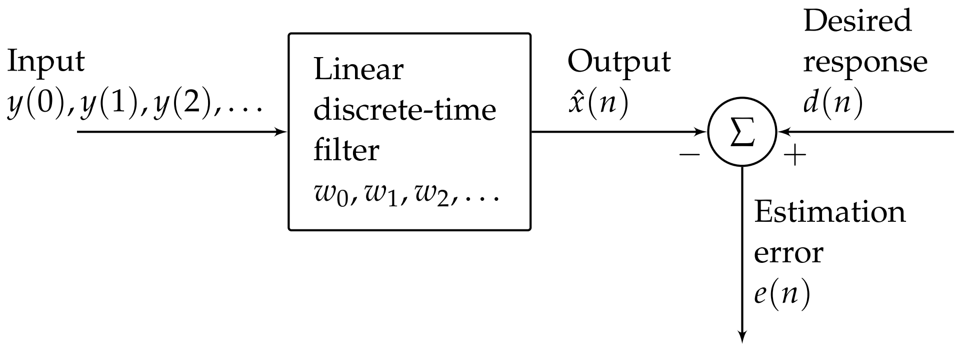

Figure 4.2: Block diagram representation of the statistical filtering problem, from [6].

Wiener filter requires a priori information about the statistics of the data to be processed, it may not be optimum for non-stationary processes. For such an environment, in which the statistics are time-varying, it needs a filter that constantly adapts its parameters to optimise its output.

4.3 ADAPTIVE FILTERS

A possible approach to mitigate the limitations of the Wiener filter for non-stationary processes is the 'estimate and plug' procedure. The filter 'estimates' the statistical parameters of the relevant signals and 'plugs' them into a *non-recursive* formula for computing the filter parameters. This procedure requires excessively elaborate and costly hardware for real-time operation [6]. To overcome this disadvantage one may use an *adaptive filter*, which is a self-designing system that relies, in contrast, on a *recursive* algorithm. This allows the filter to perform satisfactorily even if there is no complete knowledge of the relevant signal characteristics. Provided the variations in the statistics of the input data are sufficiently slow, the algorithm can track time variations and is thus suitable for non-stationary environments. The algorithm starts from some predetermined set of initial conditions respecting the knowledge about the system. In a stationary environment it converges to the optimum Wiener solution in some statistical sense after successive iterations. The *Kalman filter* is such an adaptive filter.

Due to the fact that the parameters of an adaptive filter are updated each iteration, they become data dependent. The system does not obey the principles of superposition which therefore makes the adaptive filter in reality a *non-linear* system. However, an adaptive filter is commonly said to be *linear* if its input-output map satisfies the superposition principle, as long as its parameters are held fixed. Otherwise it is said to be *non-linear*.

4.4 THE KALMAN FILTER

The *Kalman filter* is a set of recursive mathematical equations that provides an efficient means to estimate the state of a linear dynamic system perturbed by additive white Gaussian noise, even when the precise nature of the modelled system is unknown. It incorporates knowledge of the system and measurement device dynamics, the statistical description of the system errors and measurement noise, and available information about initial conditions of the variables of interest, in order to produce an estimate of these variables, in a way that the mean of the squared error is minimised [7].

The filter is named after Rudolf E. Kalman who 1960 published his famous paper describing a recursive solution to the discrete-data linear filtering problem [47]. Since that time, the Kalman filter has been the subject of extensive research, due, to a large extent, to the advances in digital computing [48]. It finds applications in radar tracking, navigation, and orientation estimation, among others. Zarchan and Musoff [49] stated: "With the possible exception of the fast Fourier transform, Kalman filtering is probably the most important algorithmic technique ever devised."

4.4.1 *An Introductory Example*

The following introductory example from May [7] is an illustrative description of the determination of a one-dimensional position to understand how the Kalman filter works. Suppose you are lost at sea during the night and take a star sighting to determine your approximate position at time t_1 to be z_1. Your location estimate is, due to inherent measurement device inaccuracies and human error, somewhat uncertain, and thus assumed to be associated with a standard deviation σ_{z_1}. The conditional probability of $x(t_1)$, your actual position at time t_1, conditioned on the observed value z_1, is depicted in Figure 4.3. The best estimate of your position, based on this conditional probability density, is

$$\hat{x}(t_1) = z_1 \tag{4.1}$$

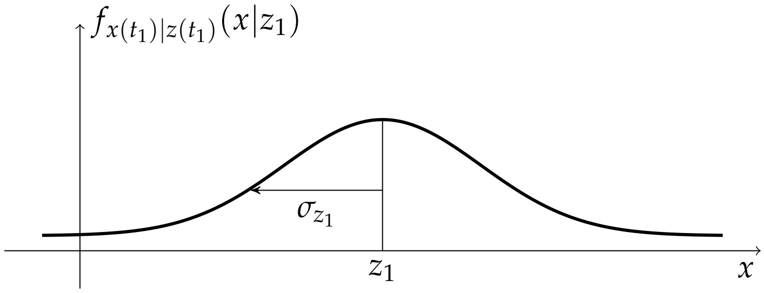

Figure 4.3: Conditional probability density of position based on measurement value z_1, from [7].

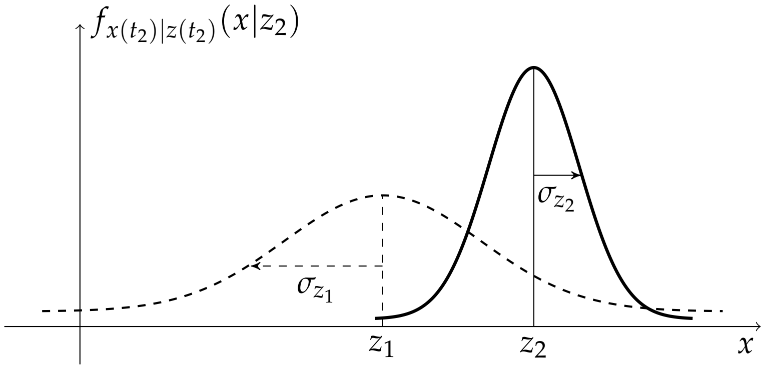

Figure 4.4: Conditional probability density of position based on measurement value z_2 alone, from [7].

and the variance of the error in the estimate is

$$\sigma_x^2(t_1) = \sigma_{z_1}^2 \,. \tag{4.2}$$

Right after you, say a trained navigator friend takes an independent fix at time $t_2 \cong t_1$, so that the true position has not changes at all. He obtains a measurement z_2 with a variance σ_{z_2}, which is somewhat smaller than yours, since he has a higher skill. Figure 4.4 depicts the conditional density of your position at time t_2, based only on the measurement value z_2. Combining these data, your position at time $t_2 \cong t_1$, $x(t_2)$, given both z_1 and z_2, is then a Gaussian density with mean μ and variance σ^2, as indicated in Figure 4.5, with

$$\mu = z_1 \frac{\sigma_{z_2}^2}{\sigma_{z_1}^2 + \sigma_{z_2}^2} + z_2 \frac{\sigma_{z_1}^2}{\sigma_{z_1}^2 + \sigma_{z_2}^2} \tag{4.3}$$

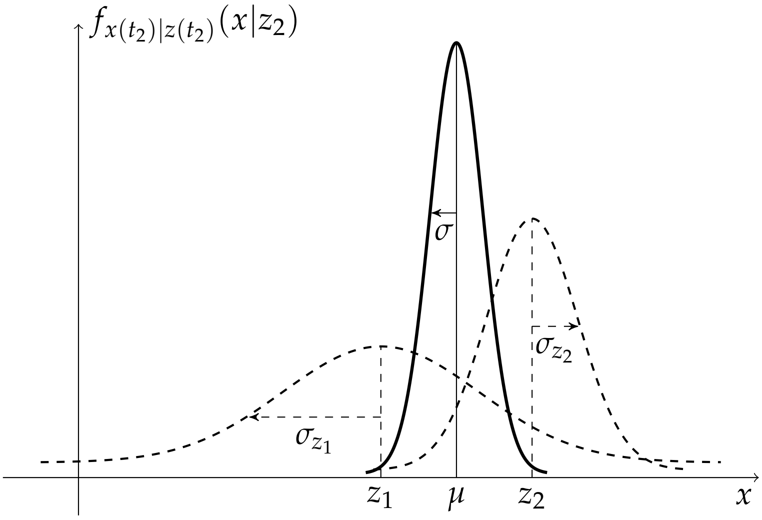

Figure 4.5: Conditional probability density of position based on data z_1 and z_2, from [7].

and

$$\frac{1}{\sigma^2} = \frac{1}{\sigma_{z_1}^2} + \frac{1}{\sigma_{z_2}^2} \,. \tag{4.4}$$

The uncertainty in your estimate of position has been decreased because σ is less than either $\sigma_{z_1}^2$ or $\sigma_{z_2}^2$. Even if σ_{z_1} was very large, the variance of the estimate is less than σ_{z_2}, which means that even poor quality data increases the precision of the filter output. The best estimate, given this density, is

$$\hat{x}(t_2) = \mu \,, \tag{4.5}$$

with an associated error variance σ^2.

Having a closer look at the form of μ in Equation 4.3, one notices that it makes good sense. If the measurements were of equal precision, meaning $\sigma_{z_1} = \sigma_{z_2}$, the optimal estimate is simply the average of both measurements, as would be expected. If σ_{z_1} is larger than σ_{z_2}, the equation weights z_2 more heavily than z_1.

Equation 4.5 for the filter output can be written as

$$\begin{aligned}\hat{x}(t_2) &= z_1\frac{\sigma_{z_2}^2}{\sigma_{z_1}^2+\sigma_{z_2}^2} + z_2\frac{\sigma_{z_1}^2}{\sigma_{z_1}^2+\sigma_{z_2}^2} \\ &= z_1 + \frac{\sigma_{z_1}^2}{\sigma_{z_1}^2+\sigma_{z_2}^2}[z_2 - z_1]\end{aligned} \tag{4.6}$$

or in a form that is used in Kalman filter implementations, with $\hat{x}(t_1) = z_1$, as

$$\hat{x}(t_2) = \hat{x}(t_1) + K(t_2)[z_2 - \hat{x}(t_1)]\,, \tag{4.7}$$

where

$$K(t_2) = \frac{\sigma_{z_1}^2}{\sigma_{z_1}^2+\sigma_{z_2}^2}\,. \tag{4.8}$$

These equation represent the 'predictor-corrector' structure of the Kalman filter. A prediction of the value that the desired variables and the measurements will have at the next measurement time is made, based on all previous information. Then the difference between the measurement and its predicted value is used to correct the prediction of the desired variables. According to Equation 4.7 the optimal estimate at time t_2, $\hat{x}(t_2)$, is equal to $\hat{x}(t_1)$, the best prediction of its value before z_2 is taken, plus a correction term of an optimal weighting value times the difference between z_2 and the best prediction of it before the measurement is actually taken.

To incorporate dynamics into the model, suppose you travel for some time before taking another measurement. The best model you have for your motion may be of the form

$$\frac{dx}{dt} = u + w\,, \tag{4.9}$$

where u is a nominal velocity and w is a noise term, representing the uncertainty in your knowledge of the actual velocity due to disturbances and effects not accounted for in the simple first order equation. It will be modelled as white Gaussian noise with a mean of zero and variance of σ_w^2.

The conditional density of the position at time t_2, given z_1 and z_2, was previously derived. Figure 4.6 shows graphically how the density travels along the x-axis as time progresses. It start at the best estimate

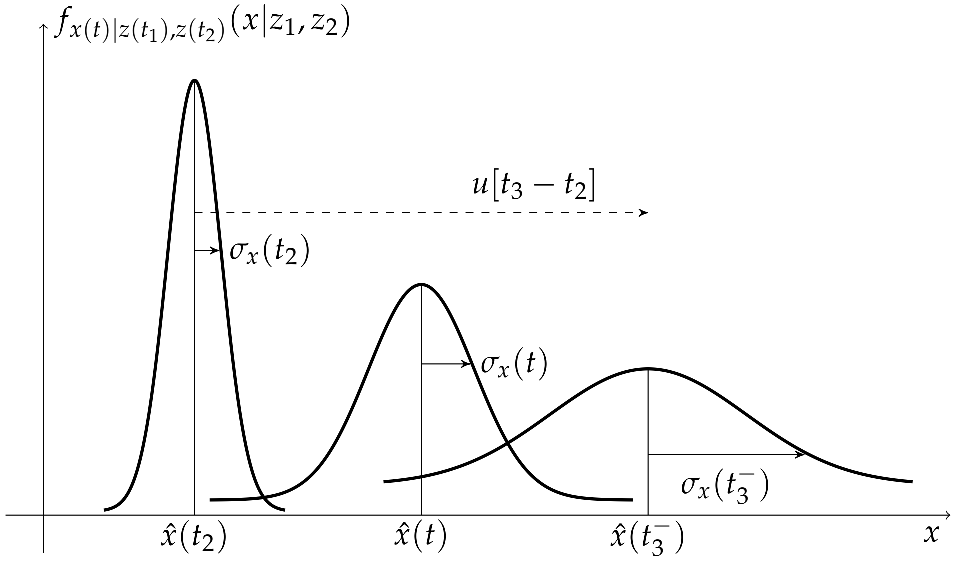

Figure 4.6: Propagation of conditional probability density, from [7].

and moves according to the above mentioned model of dynamics. Due to the constant addition of uncertainty over time it spreads out. As the variance becomes greater you become less sure of your position. The Gaussian density $f_{x(t_3)|z(t_1),z(t_2)}(x|z_1,z_2)$ can be expressed mathematically by its mean and variance given by

$$\hat{x}(t_3^-) = \hat{x}(t_2) + u[t_3 - t_2]\,, \tag{4.10}$$

$$\sigma_x^2(t_3^-) = \sigma_x^2(t_2) + \sigma_w^2[t_3 - t_2]\,. \tag{4.11}$$

Before the measurement is taken at t_3, $\hat{x}(t_3^-)$ is the optimal prediction of the location at t_3^-, associated with the variance $\sigma_x^2(t_3^-)$ in this prediction.

Now a measurement z_3 with an assumed variance $\sigma_{z_3}^2$ is taken. As before, its conditional probability density is combined with the density with mean $\hat{x}(t_3^-)$ and variance $\sigma_x^2(t_3^-)$, to yield a Gaussian density with mean

$$\hat{x}(t_3) = \hat{x}(t_3^-) + K(t_3)[z_3 - \hat{x}(t_3^-)] \tag{4.12}$$

and variance

$$\sigma_x^2(t_3) = \sigma_x^2(t_3^-) - K(t_3)\sigma_x^2(t_3^-)\,, \tag{4.13}$$

where the gain $K(t_3)$ is given by

$$K(t_3) = \frac{\sigma_x^2(t_3^-)}{\sigma_x^2(t_3^-) + \sigma_{z_3}^2}\,. \tag{4.14}$$

Observing the form of Equation 4.14 the reasonableness of the filter structure becomes obvious. If the variance of the measurement noise $\sigma_{z_3}^2$ is large, then $K(t_3)$ is small, meaning that little confidence is put in a very noisy measurement and that it is weighted lightly. For $\sigma_{z_3}^2 \to \infty$, $K(t_3)$ becomes zero, and $\hat{x}(t_3)$ equals $\hat{x}(t_3^-)$. Thus, an infinitely noisy measurement is totally ignored. Likewise, if the dynamical system noise variance σ_w^2 is large, then according to Equation 4.11, $\sigma_x^2(t_3^-)$ will be large, and so will be $K(t_3)$. Therefore, the measurement is weighted heavily, in case you are not very certain about the output of the system model within the filter structure. In the limit as $\sigma_w^2 \to \infty$, $\sigma_x^2(t_3^-) \to \infty$, and $K(t_3) \to 1$, so Equation 4.5 yields

$$\hat{x}(t_3) = \hat{x}(t_3^-) + 1 \cdot [z_3 - \hat{x}(t_3^-)] = z_3\,. \tag{4.15}$$

That means that in the limit of absolutely no confidence in the system model output, solely the new measurement is taken as the optimal estimate. Finally, if you are absolutely sure of your estimate before z_3 comes available, $\sigma_x^2(t_3^-)$ would become zero, and so would $K(t_3)$, which means that the measurements would be left disregarded.

Extending Equations 4.10, 4.11, 4.12, 4.13, and 4.14 to the vector case, and allowing time varying parameters in the system and noise description leads to the general Kalman filter equations. A complete mathematical derivation is found in Haykin [6].

4.4.2 *Formulation of the Kalman Filter Equations*

Let $\mathbf{x}_k \in \mathbb{R}^n$ be the state vector of a discrete-time controlled process governed by the linear stochastic difference equation

$$\mathbf{x}_k = \mathbf{\Phi}_{k-1}\mathbf{x}_{k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1} + \mathbf{w}_{k-1} \tag{4.16}$$

and $\mathbf{z}_k \in \mathbb{R}^m$ the observation or measurement vector of this process, given by

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k\,, \tag{4.17}$$

where the index $k \in \mathbb{N}^0$ denotes discrete time normalised to the sampling interval. The $n \times 1$ vector w_k and the $m \times 1$ vector v_k represent the process noise and the measurement noise, respectively, modelled as zero-mean, Gaussian white noise

$$\mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q}_k)\,, \tag{4.18}$$

$$\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R}_k)\,, \tag{4.19}$$

with the process noise covariance matrix $\mathbf{Q}_k$ and the measurement noise covariance matrix $\mathbf{R}_k$. The $n \times n$ transition matrix $\mathbf{\Phi}_{k-1}$ in 4.16 relates the state at the previous time step $k-1$ to the state at the current step k. The $n \times l$ matrix $\mathbf{B}_{k-1}$ relates the known, optional control input $\mathbf{u}_{k-1} \in \mathbb{R}^l$ to the state $\mathbf{x}_k$. Finally, the $m \times n$ measurement matrix $\mathbf{H}_k$ in 4.17 relates the state $\mathbf{x}_k$ to the measurement $\mathbf{z}_k$. Both noise processes are assumed to be uncorrelated. The process noise might not always have a physical meaning. However, it represents the fact that the model of the real world is not precise. The process and measurement noise covariance matrices are related to the respective noise vectors according to

$$\mathbf{Q}_k = \mathrm{E}[\mathbf{w}_k, \mathbf{w}_k^T]\,, \tag{4.20}$$

$$\mathbf{R}_k = \mathrm{E}[\mathbf{v}_k, \mathbf{v}_k^T]\,, \tag{4.21}$$

where E denotes the expected value.

The Kalman filter solves the problem of estimating the state $\mathbf{x}_k$ of the given linear stochastic system, minimising the weighted mean-squarred error. The state estimate is denoted with $\hat{\mathbf{x}}_k$, which is also a linear function of the measurement $\mathbf{z}_k$. This problem is called the *linear quadratic Gaussian* estimation problem; the dynamic system is linear, the performance cost function is quadratic, and the random process is Gaussian.

We define the vector $\hat{\mathbf{x}}_k^- \in \mathbb{R}^n$ as the *a priori* state estimate representing knowledge of the process prior to step k and $\hat{\mathbf{x}}_k \in \mathbb{R}^n$ as the *a posteriori* state estimate at step k given the measurement $\mathbf{z}_k$:

$$\hat{\mathbf{x}}_k^- = \mathbf{\Phi}_{k-1}\hat{\mathbf{x}}_{k-1} + \mathbf{B}_{k-1}\mathbf{u}_{k-1}\,, \tag{4.22}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k[\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-]\,. \tag{4.23}$$

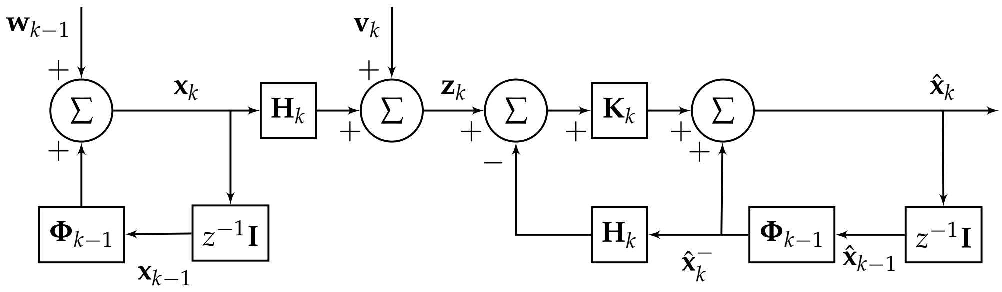

Figure 4.7: Block diagram depicting the relation between a discrete-time dynamical system, its observation, and the Kalman filter.

The term $[\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-]$ is called the measurement *innovation* or *residual*. It reflects the discordance between the predicted measurement $\mathbf{H}_k\hat{\mathbf{x}}_k^-$ and the actual measurement $\mathbf{z}_k$. The $n \times m$ matrix $\mathbf{K}_k$ is termed the Kalman gain and is given by

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^T[\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k]^{-1}\,, \tag{4.24}$$

with

$$\mathbf{P}_k^- = \mathbf{\Phi}_{k-1}\mathbf{P}_{k-1}\mathbf{\Phi}_{k-1}^T + \mathbf{Q}_{k-1} \tag{4.25}$$

and

$$\mathbf{P}_k = [\mathbf{I} - \mathbf{K}_k\mathbf{H}_k]\mathbf{P}_k^-\,. \tag{4.26}$$

Figure 4.7 illustrates the relation of the Kalman filter to the discrete-time dynamical system, where z^{-1} denotes the unit-delay and $\mathbf{I}$ the $n \times n$ identity matrix. For the sake of simplicity the control input is not depicted.

The Kalman filter equations can be divided into two groups: *time update* Equations 4.22, 4.25 and *measurement update* Equations 4.23 , 4.24, and 4.26, as seen in Figure 4.8, which shows the 'predict and correct' behaviour of the filter algorithm. After an initialisation of the parameters, the *time update* and *measurement update* steps are repeated recursively every time step.

4.4.3 *The Extended Kalman Filter*

Up to this point the Kalman filter has solved the filtering problem for *linear* time-dynamical systems. One may extend the Kalman filter to systems with state dynamics governed by *non-linear* state transformations

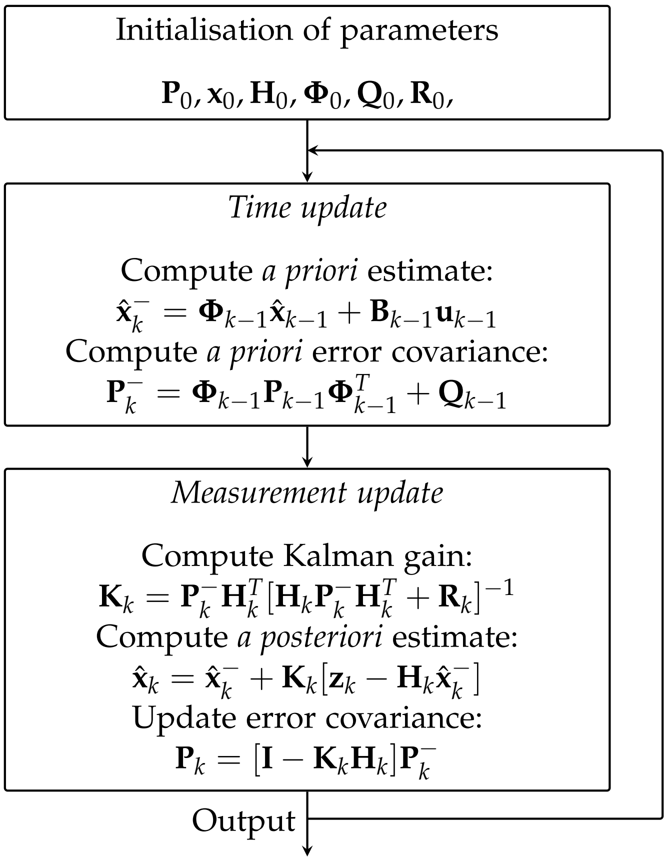

Figure 4.8: Operation cycle of the Kalman filter algorithm illustrating 'predict and correct' behaviour.

$$\mathbf{x}_k = \boldsymbol{\phi}_{k-1}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}) + \mathbf{w}_{k-1}, \quad \mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q}_k)\,, \tag{4.27}$$

and/or a *non-linear* transformation from state variables to measurement variables

$$\mathbf{z}_k = \mathbf{h}_k(\mathbf{x}_k) + \mathbf{v}_k, \quad \mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R}_k)\,. \tag{4.28}$$

The *functional* $\boldsymbol{\phi}_{k-1}$ denotes the *non-linear* transition matrix function that may be time varying. It relates the state at the previous time step $k-1$ to the current time step k. The vector $\mathbf{u}_{k-1}$ is again the exogenous control input. The functional $\mathbf{h}_k$ denotes a *non-linear* measurement matrix function that relates the state $\mathbf{x}_k$ to the measurement $\mathbf{z}_k$ and is possibly time varying, too.

Some non-linear problems can be deemed *quasilinear*, which means that the variation of the non-linear functionals $\boldsymbol{\phi}$ and $\mathbf{h}$ are predominantly linear about the value $\mathbf{x}_0$. That is,

$$\boldsymbol{\phi}_k(\mathbf{x}_0 + d\mathbf{x}, \mathbf{u}) \approx \boldsymbol{\phi}_k(\mathbf{x}_0, \mathbf{u}) + d\mathbf{x} \left.\frac{\partial \boldsymbol{\phi}_k(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}}\right|_{\mathbf{x}_0, \mathbf{u}}, \tag{4.29}$$

$$\mathbf{h}_k(\mathbf{x}_0 + d\mathbf{x}) \approx \mathbf{h}_k(\mathbf{x}_0) + d\mathbf{x} \left.\frac{\partial \mathbf{h}_k(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}_0}, \tag{4.30}$$

which requires that $\boldsymbol{\phi}$ and $\mathbf{h}$ are differentiable at $\mathbf{x}$.

Through a *linearisation* of the state-space model of Equations 4.27 and 4.28 at each time instant around the most recent state estimate, the standard Kalman filter equation from Section 4.4.2 can be applied. The filter resulting from a *linear approximation* of the state transitions and the relation of the measurement to the respective state is referred to as the *extended Kalman filter* (EKF).

Similar to Equation 4.22, the predicted state estimate is given by

$$\hat{\mathbf{x}}_k^- = \boldsymbol{\phi}_{k-1}(\mathbf{x}_{k-1}, \mathbf{u}_{k-1}), \tag{4.31}$$

and the predicted measurement by

$$\hat{\mathbf{z}}_k = \mathbf{h}_k(\hat{\mathbf{x}}_k^-). \tag{4.32}$$

The *a posteriori* estimate is then, conditioned on the actual measurement,

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k[\mathbf{z}_k - \hat{\mathbf{z}}_k]. \tag{4.33}$$

The corresponding *a priori* covariance matrix $\mathbf{P}_k^-$, the Kalman gain $\mathbf{K}_k$, and the *a posteriori* covariance matrix $\mathbf{P}_k$ are equal to Equations 4.24, 4.25, and 4.26 in Section 4.4.2. They are reproduced here with the linearised state transition and measurement matrices for convenience of presentation:

$$\mathbf{P}_k^- = \mathbf{\Phi}_{k-1}^{[1]} \mathbf{P}_{k-1} \mathbf{\Phi}_{k-1}^{[1]T} + \mathbf{Q}_{k-1}, \tag{4.34}$$

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^{[1]T} [\mathbf{H}_k^{[1]} \mathbf{P}_k^- \mathbf{H}_k^{[1]T} + \mathbf{R}_k]^{-1}, \tag{4.35}$$

$$\mathbf{P}_k = [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k^{[1]}] \mathbf{P}_k^-. \tag{4.36}$$

The Jacobian matrices of the functionals $\boldsymbol{\phi}$ and $\mathbf{h}$ are given by

$$\mathbf{\Phi}_{k-1}^{[1]} = \left. \frac{\partial \boldsymbol{\phi}_{k-1}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k-1}^{-}, \mathbf{u}_{k-1}} , \tag{4.37}$$

and

$$\mathbf{H}_{k}^{[1]} = \left. \frac{\partial \mathbf{h}_{k}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k}^{-}} . \tag{4.38}$$

The ij^{th} entry of $\mathbf{\Phi}_{k-1}^{[1]}$ is equal to the partial derivative of the i^{th} component of $\boldsymbol{\phi}_{k-1}(\mathbf{x})$ with respect to the j^{th} component of $\mathbf{x}$. The derivatives are evaluated at $\mathbf{x} = \hat{\mathbf{x}}_{k-1}^{-}$. Likewise, the ij^{th} entry of $\mathbf{H}_{k-1}^{[1]}$ is equal to the partial derivative of the i^{th} component of $\mathbf{h}_k(\mathbf{x})$ with respect to the j^{th} component of $\mathbf{x}$. The derivatives are evaluated at $\mathbf{x} = \hat{\mathbf{x}}_{k}^{-}$. The superscript $^{[1]}$ denotes the *first-order* approximation.

Figure 4.9 illustrates the 'predict and correct' behaviour of the extended Kalman filter algorithm. Likewise, after an initialisation of the parameters, the *time update* and *measurement update* steps are repeated recursively every time step. In addition to the standard Kalman filter equations, the Jacobian matrices have to be computed, in order to linearise the state-space model at each time instant around the most recent state estimate.

Extended Kalman filtering is commonly used. In fact, it was the first successful application of the Kalman filter [50]. Unlike its linear counterpart, the extended Kalman filter may not necessarily be an optimal estimator. Owing to its linearisation the EKF may quickly diverge, if the process is modelled incorrectly or the initial state estimate is wrong.

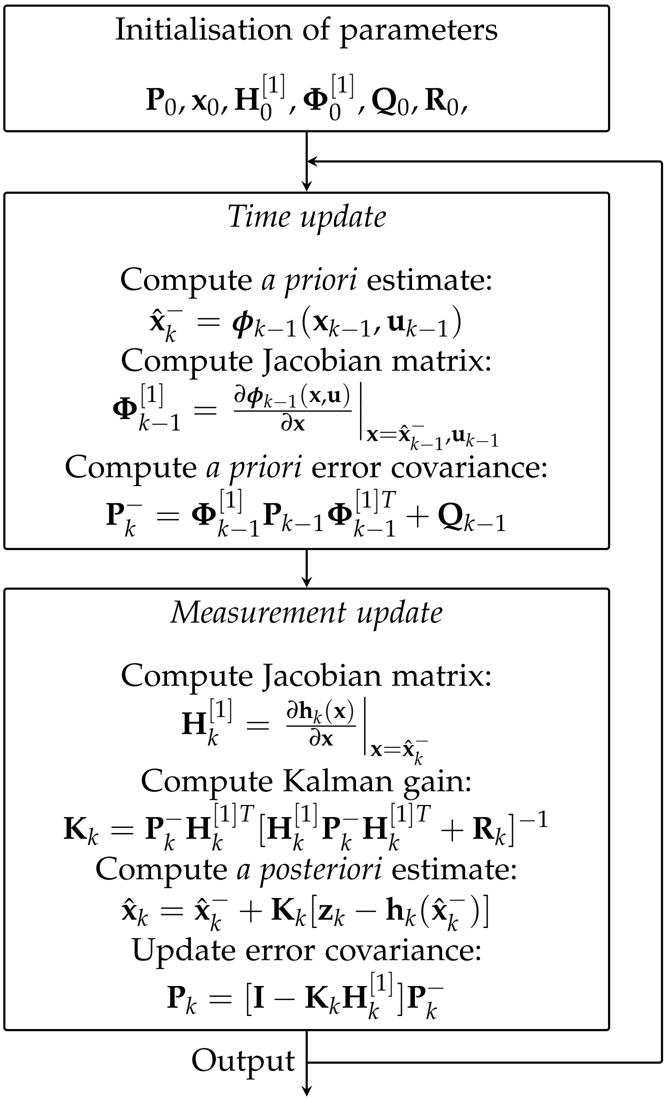

Figure 4.9: Operation cycle of the extended Kalman filter algorithm illustrating 'predict and correct' behaviour.

5 IMPLEMENTATION

This chapter describes the theoretical design of the filter algorithm proposed in [5], as well as its implementation based on the fundamentals acquired in the previous chapters. Other than Bennett et al. in [5] who tested the filter algorithm by hand, we used movement data from real subjects. After outlining the initial situation, i. e. stating existing orientation algorithms, the theoretical design of the filter is described in detail. Subsequently, the software implementation and experiments are given, followed by the results and its discussion.

5.1 INITIAL SITUATION

There were already a variety of Kalman filter algorithms implemented. In tandem with an orientation estimation based on a Qualisys motion capture system using cameras in combination with optical markers, they served as a reference. The placement of the markers on the leg are depicted in Figure 5.1. From the recorded motion of the markers in space the motion capture system computed the orientation angles of the thighs and shanks.

(Any algorithms that are already implemented for comparison that I can cite or state and describe how they differ from the one that I will implement??)

5.2 THEORETICAL DESIGN

This section maps the theoretical design of the system proposed by Bennett et al. in [5] to the existing GaitWatch system. It states the assigned coordinate frames and the conventions with respect to rotation about them. Furthermore, it describes the kinematic model and the extended Kalman filter in detail.

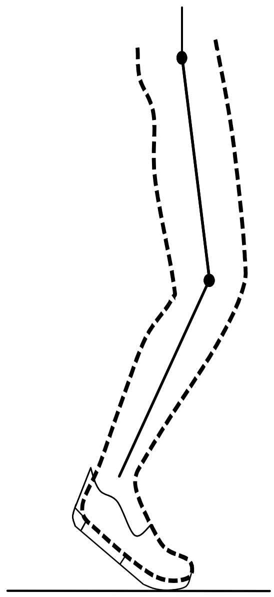

Figure 5.1: Human leg with optical markers, from [8].

5.2.1 *Kinematic Model*

The *kinematic model* relates the respective angles of the thigh and shank about the hip and knee joint to the acceleration seen by the wearable sensors. When walking in a straight line, the human leg can be modelled as a two-link planar revolute robot [5]. Then, thighs and shanks remain in a single plane which is approximately parallel to the direction of motion. As depicted in Figure 5.2, the revolute joints of the pendulum robot (Pendubot) represent the hip und knee joint, and the two links the thigh and shank, respectively. The origin of the inertial world frame is located at the base of link 1, the upper of both links. The x-axis points forward, the y-axis points out from the hip to the right, and the z-axis points down. Thus, since the figure depicts the right leg from lateral, the y-axis points out of the page. This configuration follows the right-hand rule, which can also be used to determine the sense of rotation around the axes. The pitch angle θ_1 is measured with respect to the x-axis, and the pitch angle θ_2 of link 2 with respect to link 1.

The IMUs placed on the thighs and shanks measured the angular velocity and linear acceleration of the thighs and shanks, respectively. According to Spong and Hutchinson [51], the x and z displacement and its derivatives in the world frame are as follows:

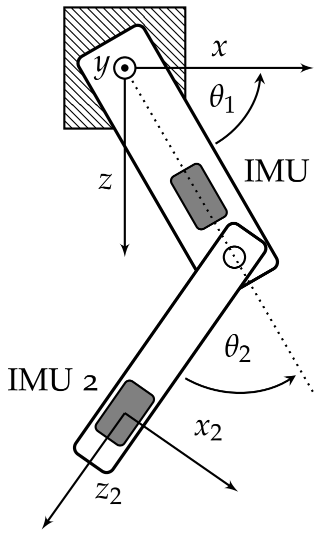

Figure 5.2: Kinematic model of the human leg, from [5].

$$x = l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) \tag{5.1}$$

$$\dot{x} = -l_1\dot{\theta}_1 \sin\theta_1 - l_2(\dot{\theta}_1 + \dot{\theta}_2)\sin(\theta_1 + \theta_2) \tag{5.2}$$

$$\begin{aligned}\ddot{x} = &-l_1[\dot{\theta}_1^2 \cos\theta_1 + \ddot{\theta}_1 \sin\theta_1] - l_2[(\dot{\theta}_1 + \dot{\theta}_2)^2 \cos(\theta_1 + \theta_2) \\ &+(\ddot{\theta}_1 + \ddot{\theta}_2)\sin(\theta_1 + \theta_2)]\end{aligned} \tag{5.3}$$

$$z = -l_1 \sin\theta_1 - l_2 \sin(\theta_1 + \theta_2) \tag{5.4}$$

$$\dot{z} = -l_1\dot{\theta}_1 \cos\theta_1 - l_2(\dot{\theta}_1 + \dot{\theta}_2)\cos(\theta_1 + \theta_2) \tag{5.5}$$

$$\begin{aligned}\ddot{z} = &-l_1[\ddot{\theta}_1 \cos\theta_1 - \dot{\theta}_1^2 \sin\theta_1] - l_2[(\ddot{\theta}_1 + \ddot{\theta}_2)\cos(\theta_1 + \theta_2) \\ &+(\dot{\theta}_1 + \dot{\theta}_2)^2 \sin(\theta_1 + \theta_2)]\end{aligned} \tag{5.6}$$

in which l_1 and l_2 are the lengths of the two links, respectively. Using Equations 5.3 and 5.6, and the estimates of the angles θ_1 and θ_2 and their derivatives obtained with the EKF described in Section 5.2.2 we can estimate the motion based acceleration in x and z direction that sensor 2 will see in the world coordinate frame.

The orientation of the sensor frames at rest are different from the world frame and dynamic when the pendulum is in motion. In order to transform the values from the world frame to the dynamic body frame of IMU 2, which is depicted in Figure 5.2, we used the transformation matrix $\mathbf{T}_y(\theta)$ from Equation 3.2 in transposed form. The positive sense of rotation according the right-hand rule is opposite to the mathematically positive sense, which is why we use the transpose of the transformation matrix. The body frame of sensor two is not aligned with the world frame for $\theta_1 = \theta_2 = 0$. Thus, an offset of $-\frac{\pi}{2}$ is necessary. With $\theta = \theta_1 + \theta_2 - \frac{\pi}{2}$, this yields

$$\mathbf{T}_y^T(\theta_1 + \theta_2 - \frac{\pi}{2}) = \begin{bmatrix} \cos(\theta_1 + \theta_2 - \frac{\pi}{2}) & 0 & \sin(\theta_1 + \theta_2 - \frac{\pi}{2}) \\ 0 & 1 & 0 \\ -\sin(\theta_1 + \theta_2 - \frac{\pi}{2}) & 0 & \cos(\theta_1 + \theta_2 - \frac{\pi}{2}) \end{bmatrix} . \tag{5.7}$$

The rotated tangential and radial components of the motion based acceleration estimates, A_{rad} and A_{tan} are found using the tranformation matrix to rotate the results of Equations 5.3 and 5.6, respectively, according to Equation 3.1.

$$\begin{bmatrix} A_{rad} \\ 0 \\ A_{tan} \end{bmatrix} = \mathbf{T}_y^T(\theta_1 + \theta_2 - \frac{\pi}{2}) \begin{bmatrix} \ddot{x} \\ 0 \\ \ddot{z} \end{bmatrix} \tag{5.8}$$

Then, the radial and tangential acceleration estimates are subtracted from the sensor readings A_x and A_y, which leaves an estimate of the gravity based acceleration $\mathbf{g}$ that acts on the sensor:

$$\mathbf{g} = \begin{bmatrix} g_x \\ g_y \\ g_z \end{bmatrix} = \begin{bmatrix} A_x \\ 0 \\ A_z \end{bmatrix} - \begin{bmatrix} A_{rad} \\ 0 \\ A_{tan} \end{bmatrix} . \tag{5.9}$$

According to Equation 3.18 the angle estimate is

$$\theta_1 + \theta_2 = \mathrm{atan2}(g_z, g_x) . \tag{5.10}$$

This improved angle estimate is then fused with the estimate based on the integration of the angular rate measured with the gyroscope, in order to reduce the estimation error due to gyroscope drift.

5.2.2 *Extended Kalman Filter Model*

The state-space model of the extended Kalman filter is given by the state vector $\mathbf{x} \in \mathbb{R}^{n=10}$

$$\mathbf{x} = \begin{bmatrix} x, & z, & \theta_1, & \omega_1, & \alpha_1, & \theta_2, & \omega_2, & \alpha_2, & \beta_1, & \beta_2 \end{bmatrix}^T \tag{5.11}$$

where x and y correspond to the horizontal and vertical position of the end of link 2 with respect to the origin of the world frame, i. e. the hip joint. θ_1 is the angle, ω_1 the angular velocity, and α_1 the angular acceleration of the first joint, respectively. The corresponding values for the second link are θ_2, ω_2, and α_2. The biases from the gyroscope on the first and second sensor are β_1 and β_2, respectively. They are assumed to be constant or slowly time varying.

The measurement vector $\mathbf{z} \in \mathbb{R}^{m=3}$ is given by

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} = \begin{bmatrix} \omega_1 + \beta_1, & \omega_1 + \omega_2 + \beta_2, & \theta_1 + \theta_2 \end{bmatrix}^T + \mathbf{v}\,. \tag{5.12}$$

where $\mathbf{v}$ is the random measurement noise process, modelled as zero-mean, Gaussian white noise. The element z_1 represents the measurement of the first link angular velocity, which is the sum of the first link rotation and the gyroscope 1 bias. Equally, the element z_2 represents the measurement of the second link angular velocity, which is the sum of the first and second link rotation and the bias of gyroscope 2. Finally, the element z_3 is the angle estimate of the second accelerometer, which will see the angular displacement of both links.

According to Rowell [52], the plant dynamics of a system can be expressed as a set of n coupled first-order ordinary differential equations, known as the *state equations*. The modelled system is governed by the *non-linear* ordinary differential equations

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t) + \mathbf{w} = \begin{bmatrix} -l_1\omega_1 \sin\theta_1 - l_2(\omega_1+\omega_2)\sin(\theta_1+\theta_2) \\ -l_1\omega_1 \cos\theta_1 - l_2(\omega_1+\omega_2)\cos(\theta_1+\theta_2) \\ \omega_1 \\ \alpha_1 \\ 0 \\ \omega_2 \\ \alpha_2 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \mathbf{w}\,. \tag{5.13}$$

where $\dot{\mathbf{x}}$ consists of the component-wise time derivatives of the state vector $\mathbf{x}$,

$$\dot{x}_i = f_i(\mathbf{x}, t) = \frac{dx_i}{dt}, \quad i = 1, \ldots, n\,, \tag{5.14}$$

expressed in terms of the state variables $x_1(t), \ldots, x_n(t)$. Given this *state-space representation*, the system state at any instant may be interpreted as a point in an n-dimensional state space whose axes are the state variables. The dynamic state response $\mathbf{x}(t)$ can be interpreted as a trajectory traced out in the state space. The system described by Equation 5.13 is *time-invariant* since it does not depend explicitly on time. Thus, we may leave out the t and write from now on $\mathbf{f}(\mathbf{x}) = \mathbf{f}(\mathbf{x}, t)$.

For a *linear* system in state space form given by

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x}\,, \tag{5.15}$$

with a time-invariant *system dynamics matrix* $\mathbf{F}$ there is a *state transition matrix* $\mathbf{\Phi}(t - t_0)$ that propagates the state of the system forward from any time t_0 to a time t, according to

$$\mathbf{x}(t) = \mathbf{\Phi}(t - t_0)\mathbf{x}(t_0)\,. \tag{5.16}$$

The solution to the system described by Equation 5.15 is

$$\mathbf{x}(t) = e^{\mathbf{F}(t-t_0)}\mathbf{x}(t_0)\,. \tag{5.17}$$

As outlined in [49], the state transition matrix can be found by a Taylor-series expansion of $e^{\mathbf{F}(t-t_0)}$

$$\begin{aligned} \mathbf{\Phi}(t - t_0) = e^{\mathbf{F}(t-t_0)} &= \sum_{k=0}^{\infty} \frac{\mathbf{F}^n [t - t_0]^n}{k!} \\ &= \mathbf{I}_n + \mathbf{F}[t - t_0] \\ &\quad + \frac{\mathbf{F}^2 [t - t_0]^2}{2!} + \frac{\mathbf{F}^3 [t - t_0]^3}{3!} + \cdots\,, \end{aligned} \tag{5.18}$$

where $\mathbf{I}_n \in \mathbb{R}^{n \times n}$ is the identity matrix. Truncating the Taylor series after the first order terms yields the linear approximation of the fundamental matrix:

$$\mathbf{\Phi}(t - t_0) \approx \mathbf{I}_n + \mathbf{F}[t - t_0]\,. \tag{5.19}$$

The discrete fundamental matrix that propagates the state from time stem k to $k + 1$ can be found by substituting T_s for $t - t_0$, which yields

$$\mathbf{\Phi}_{k \to k+1} = \mathbf{\Phi}(T_s) \approx \mathbf{I}_n + \mathbf{F}T_s\,, \tag{5.20}$$

where T_s is the sampling period.

Because the state equations of our system are *non-linear*, a first-order approximation of the system dynamics matrix $\mathbf{F}$ is used, given by the Jacobian of $\mathbf{f}(\mathbf{x})$

$$
\begin{aligned}
\mathbf{F}_k^{[1]} = \mathbf{J}_\mathbf{f} &= \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix} \\
&= \begin{bmatrix} 0 & 0 & A & C & 0 & E & G & 0 & 0 & 0 \\ 0 & 0 & B & D & 0 & F & H & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{\mathbf{x}=\hat{\mathbf{x}}_k} ,
\end{aligned}
\tag{5.21}
$$

with

$$
\begin{aligned}
A &= -l_1\omega_1\cos(\theta_1) - l_2(\omega_1+\omega_2)\cos(\theta_1+\theta_2)\,, \\
B &= +l_1\omega_1\sin(\theta_1) + l_2(\omega_1+\omega_2)\sin(\theta_1+\theta_2)\,, \\
C &= -l_1\sin(\theta_1) - l_2\sin(\theta_1+\theta_2)\,, \\
D &= -l_1\cos(\theta_1) - l_2\cos(\theta_1+\theta_2)\,, \\
E &= -l_2(\omega_1+\omega_2)\cos(\theta_1+\theta_2)\,, \\
F &= +l_2(\omega_1+\omega_2)\sin(\theta_1+\theta_2)\,, \\
G &= -a_2\sin(\theta_1+\theta_2)\,, \\
H &= +a_2\cos(\theta_1+\theta_2)\,.
\end{aligned}
$$

The partial derivatives are evaluated at the state estimate $\hat{\mathbf{x}}_k$. The discrete state transition matrix must be recomputed every time step k. It is given by

$$
\mathbf{\Phi}_k^{[1]} \approx \mathbf{I}_n + \mathbf{F}_k^{[1]} T_s\,. \tag{5.22}
$$

The estimate $\hat{\mathbf{x}}_{k-1}$ can be propagated forward to the *a priori* estimate $\hat{\mathbf{x}}_k^-$ by integrating the non-linear differential equation at each sampling interval. Applying Euler integration Equation 4.31 yields

$$\begin{aligned} \hat{\mathbf{x}}_k^- &= \boldsymbol{\phi}_{k-1}(\hat{\mathbf{x}}_{k-1}, 0) \\ &= \hat{\mathbf{x}}_{k-1} + \mathbf{f}(\hat{\mathbf{x}}_{k-1}) T_s \,, \end{aligned} \tag{5.23}$$

where T_s is the integration interval. The control input $\mathbf{u}_k$ is equal to zero since the system does not have any inputs. A higher-order numerical integration procedure would not improve the *a priori* estimate since the function $\mathbf{f}(\mathbf{x}, t)$ is constant over time.

The relation between the states and the measurements is linear. The measurement matrix $\mathbf{H} \in \mathbb{R}^{3\times 10}$ according to Equation 4.17 is given by

$$\mathbf{H} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} . \tag{5.24}$$

The process and measurement covariance matrices $\mathbf{Q} \in \mathbb{R}^{10\times 10}$ and $\mathbf{R} \in \mathbb{R}^{3\times 3}$, respectively, are given by

$$\mathbf{Q} = \begin{bmatrix} \sigma_d & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \sigma_d & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\sigma_{\theta 1}^9}{9} & \frac{\sigma_{\theta 1}^4}{4} & \frac{\sigma_{\theta 1}^5}{5} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\sigma_{\theta 1}^4}{4} & \frac{\sigma_{\theta 1}^3}{3} & \frac{\sigma_{\theta 1}^2}{2} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\sigma_{\theta 1}^5}{5} & \frac{\sigma_{\theta 1}^2}{2} & \sigma_{\theta 1} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{\sigma_{\theta 2}^9}{9} & \frac{\sigma_{\theta 2}^4}{4} & \frac{\sigma_{\theta 2}^5}{5} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{\sigma_{\theta 2}^4}{4} & \frac{\sigma_{\theta 2}^3}{3} & \frac{\sigma_{\theta 2}^2}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{\sigma_{\theta 2}^5}{5} & \frac{\sigma_{\theta 2}^2}{2} & \sigma_{\theta 2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_\beta & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \sigma_\beta \end{bmatrix} . \tag{5.25}$$

$$\mathbf{R} = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} , \tag{5.26}$$

The process noise matrix $\mathbf{Q}$ is constant. The parameters $\sigma_d, \sigma_{\theta 1}, \sigma_{\theta 2}$ and σ_β are determined by tuning for the best performance. The elements σ_1 and σ_2 of $\mathbf{R}$ are constant. They are determined by computing the sample standard deviation of the measurement data during an initialisation stage while the subject stands still. The standard deviation of a finite data set with n samples $x_1, x_2, \ldots, x_n$ is

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2}, \text{ where } \mu = \frac{1}{n}\sum_{i=1}^{n} x_i \,. \tag{5.27}$$

The standard deviation σ_3 is set dynamically based on the motion intensity. It toggles between σ_s and σ_f according to slow or fast motion, distinguished by a threshold δ:

$$\sigma_3 = \begin{pmatrix} \sigma_s & < \delta \\ \sigma_f & \text{else} \end{pmatrix} \tag{5.28}$$

In order to determine the motion intensity we used the LTSD developed in [3]. The LTSD computes the long term spectral envelope of the signal.

5.2.3 *Summary of the Entire Filter Algorithm*

The state estimates are computed recursively according to the *time update* Equations 4.34, 5.23 and the *measurement update* Equations 4.23, 4.24, and 4.26. The entire computation steps of the recursive filter algorithm are summarised in Figure 5.3.

5.3 MATLAB® IMPLEMENTATION

The filter algorithm was implemented in MATLAB®. Listing 6.2 shows the source code of the filter function. Listing 6.2 shows the script used to run the experiments.

5.4 EXPERIMENTS

The movement data was gathered at the Department of Neurology of the Klinikum Großhadern, Munich.

5.4.1 *Test Sequence*

The subject wore the GaitWatch system on its body. Then, the following sequence was carried out by the patient: The subject stood in front of the force plate. Then, the GaitWatch and force plate record was started and

Initialisation of parameters

$$\mathbf{x}_0, \mathbf{P}_0, \mathbf{H}, \mathbf{Q}, \mathbf{R}_0,$$

Time update

Compute fundamental matrix:

$$\mathbf{\Phi}_{k-1}^{[1]} \approx \mathbf{I}_n + \mathbf{F}_{k-1} T_s$$

Compute *a priori* estimate:

$$\hat{\mathbf{x}}_k^- = \mathbf{\Phi}_{k-1}^{[1]} \hat{\mathbf{x}}_{k-1}$$

Compute *a priori* error covariance:

$$\mathbf{P}_k^- = \mathbf{\Phi}_{k-1}^{[1]} \mathbf{P}_{k-1} \mathbf{\Phi}_{k-1}^{[1]T} + \mathbf{Q}_{k-1}$$

Correct sensor readings

Compute acceleration due to motion:

$$\ddot{x} = -l_1[\omega_1^2 \cos\theta_1 + \alpha_1 \sin\theta_1] - l_2[(\omega_1 + \omega_2)^2 \cdot \cos(\theta_1 + \theta_2) + (\alpha_1 + \alpha_2)\sin(\theta_1 + \theta_2)]$$

$$\ddot{z} = -l_1[\alpha_1 \cos\theta_1 - \omega_1^2 \sin\theta_1] - l_2[(\alpha_1 + \alpha_2) \cdot \cos(\theta_1 + \theta_2) + (\omega_1 + \omega_2)^2 \sin(\theta_1 + \theta_2)]$$

Rotate acceleration to body frame:

$$A_{rad} = \mathbf{T}_y^T(\theta_1 + \theta_2 - \tfrac{\pi}{2})\ddot{x}$$

$$A_{tan} = \mathbf{T}_y^T(\theta_1 + \theta_2 - \tfrac{\pi}{2})\ddot{z}$$

Compute gravity estimate:

$$\mathbf{g} = \begin{bmatrix} g_x \\ g_y \end{bmatrix} = \begin{bmatrix} A_x \\ A_z \end{bmatrix} - \begin{bmatrix} A_{rad} \\ A_{tan} \end{bmatrix}$$

Compute corrected angle estimate:

$$\theta_1 + \theta_2 = \text{atan2}(g_y, g_x)$$

Measurement update

Compute Kalman gain:

$$\mathbf{K}_k = \mathbf{P}_k^- \mathbf{H}_k^T [\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k]^{-1}$$

Compute *a posteriori* estimate:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k[\mathbf{z}_k - \mathbf{H}_k \hat{\mathbf{x}}_k^-]$$

Update error covariance:

$$\mathbf{P}_k = [\mathbf{I} - \mathbf{K}_k \mathbf{H}_k]\mathbf{P}_k^-$$

Output

Figure 5.3: Entire computation steps of the recursive filter algorithm.

the subject made a step onto the force plate. After standing upright for a variable time of two to ten seconds the subject left the force plate, made a few steps, turned left, and stopped in front of it again. This sequence was repeated ten times.

5.4.2 *Initial Conditions*

Each trial began with the subject standing still and it was assumed that the subjects legs were fully stretched. This leads to the following initial state estimates:

$$\begin{aligned} x &= 0\,m\,, & z &= -(l_1 + l_2)\,m\,, \\ \theta_1 &= -\frac{\pi}{2}\,\mathrm{rad}\,, & \theta_2 &= 0\,\mathrm{rad}\,, \\ \omega_1 &= 0\,\frac{\mathrm{rad}}{\mathrm{s}}\,, & \omega_2 &= 0\,\frac{\mathrm{rad}}{\mathrm{s}}\,, \\ \beta_1 &= \mu_1\,\frac{\mathrm{rad}}{\mathrm{s}}\,, & \beta_2 &= \mu_2\,\frac{\mathrm{rad}}{\mathrm{s}}\,, \\ \alpha_1 &= 0\,\frac{\mathrm{rad}}{\mathrm{s}^2}\,, & \alpha_2 &= 0\,\frac{\mathrm{rad}}{\mathrm{s}^2}\,, \end{aligned} \tag{5.29}$$

where μ_1 and μ_2 are the mean values of the gyroscope signals collected during the first $t_{init} = 2\,\mathrm{s}$ of the rest period before the subject started moving. The mean value is given by

$$\mu = \frac{1}{n}\sum_{i=1}^{n} \omega_i\,, \quad n = t_{init} f_s\,, \tag{5.30}$$

where f_s is the sampling frequency.

The initial covariance matrix $\mathbf{P}_0$ was given by

$$\mathbf{P}_0 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5.31}$$

5.4.3 *Parameterisation*

The above-mentioned parameters were The parameters $\sigma_d, \sigma_{\theta 1}, \sigma_{\theta 2}$ and σ_β in the process noise covariance matrix $\mathbf{Q}$ were determined by tuning for the best performance:

$$\begin{aligned} \sigma_d &= , \\ \sigma_{\theta 1} &= , \\ \sigma_{\theta 2} &= , \\ \sigma_\beta &= , \\ \delta &= , \end{aligned} \qquad \begin{aligned} \sigma_1 &= , \\ \sigma_2 &= , \\ \sigma_s &= , \\ \sigma_f &= , \end{aligned} \tag{5.32}$$

5.5 RESULTS

5.6 DISCUSSION

6 CONCLUSION AND FUTURE WORK

6.1 CONCLUSIONS

Gait analysis is a useful tool both in clinical practice and biomechanical research. In order to replace motion capture systems with low cost wearable Magnetic, angular rate, and gravity sensors (MARG sensors) sensors detailed technical matters still need to be improved [8].

Summarising the above, I can say that I have learned a lot in the four month that I spent in Granada. Amongst others I have come to know many new work methods, not only due to being exposed to people from a different culture, but also due to the fact that scientific research differs strongly from the work as a student at university. I gained a deeper understanding of orientation estimation and how Kalman filtering improves those. Therefore I had to study the principles of Kalman filters as well as the basics of MARG sensors. I was able to improve my MATLAB® skills and have realised how important it is to write understandable and well commented code, if it is for a larger project and not only for a coursework. I am now familiar with tools such as GitHub and Pivotal Tracker which make working in a team much easier and significantly more efficient. Beside my work at the research centre, where I obtained a valuable insight into scientific research, I read a book about scientific writing that helped me to improve my oral and written English skills during my stay. Furthermore I now know the fundamentals of LaTeX.

All in all it was a great experience, professionally as well as personally. I truly and unreservedly recommend such a stay to *every* university student.

6.2 FUTURE WORK

Medical engineering is a very interesting blend of both my major interests, that is, working in the medical field as a paramedic and in the technical field as an electrical engineer. There is a variety of possible future work. One related topic would be the validation of the pitch angles measured with the gyroscopes of the GaitWatch by means of cameras that record the trace of visual markers. From these markers one could compute the pitch angels and compare them to the those of the GaitWatch.

BIBLIOGRAPHY

[1] Jussi Collin, Pavel Davidson, Martti Kirkko-Jaakkola, and Helena Leppäkoski. Inertial sensors and their applications. In Shuvra S. Bhattacharyya, Ed F. Deprettere, Rainer Leupers, and Jarmo Takala, editors, *Handbook of Signal Processing Systems*, pages 69–96. Springer New York, 2013. ISBN 978-1-4614-6858-5. doi: 10.1007/978-1-4614-6859-2_3. URL http://dx.doi.org/10.1007/978-1-4614-6859-2_3.

[2] M.N. Armenise, C. Ciminelli, F. Dell'Olio, and V.M.N. Passaro. *Advances in Gyroscope Technologies*. Springer, 2010. ISBN 9783642154942. URL https://books.google.de/books?id=lJUiyigJRBgC.

[3] Alberto Olivares Vicente and Kai Bötzel. *GaitWatch User Manual*.

[4] Juansempere. Wikimedia Commons File: Taitbrianzyx.svg. URL http://commons.wikimedia.org/wiki/File:Taitbrianzyx.svg. [Accessed 31 March, 2015].

[5] T.R. Bennett, R. Jafari, and N. Gans. Motion Based Acceleration Correction for Improved Sensor Orientation Estimates. In *2014 11th International Conference on Wearable and Implantable Body Sensor Networks (BSN)*, pages 109–114, June 2014. doi: 10.1109/BSN.2014.17.

[6] Simon Haykin. *Adaptive filter theory*. Prentice Hall, Upper Saddle River, N.J, 2002. ISBN 9780130484345.

[7] Chapter 1 introduction. In Peter S. Maybeck, editor, *Stochastic Models, Estimation and Control: Volume 1*, Mathematics in Science and Engineering, pages 1 – 24. Elsevier, 1979. doi: 10.1016/S0076-5392(08)62166-9. URL http://www.sciencedirect.com/science/article/pii/S0076539208621669.

[8] Weijun Tao, Tao Liu, Rencheng Zheng, and Hutian Feng. Gait analysis using wearable sensors. *Sensors*, 12(2):2255–2283, February 2012. doi: 10.3390/s120202255. URL http://www.mdpi.com/1424-8220/12/2/2255.

[9] Paolo Bonato. Advances in wearable technology and applications in physical medicine and rehabilitation. *Journal of NeuroEngineering and Rehabilitation*, 2:2–2, 2005. ISSN 1743-0003. doi: 10.1186/1743-0003-2-2. URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC552335/.

[10] A. Godfrey, R. Conway, D. Meagher, and G. ÓLaighin. Direct measurement of human movement by accelerometry. *Medical Engineering & Physics*, 30(10):1364–1386, December 2008. ISSN 1350-4533. doi: 10.1016/j.medengphy.2008.09.005. URL http://www.sciencedirect.com/science/article/pii/S1350453308001653.

[11] Wee-Soon Yeoh, I. Pek, Yi-Han Yong, Xiang Chen, and A.B. Waluyo. Ambulatory monitoring of human posture and walking speed using wearable accelerometer sensors. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pages 5184–5187, Aug 2008. doi: 10.1109/IEMBS.2008.4650382.

[12] Terrell Bennett, Roozbeh Jafari, and Nicholas Raphael Gans. An extended Kalman filter to estimate human gait parameters and walking distance. September 2013. URL http://www.essp.utdallas.edu/uploads/Main/Publications/ACC13.pdf.

[13] Wai Yin Wong, Man Sang Wong, and Kam Ho Lo. Clinical Applications of Sensors for Human Posture and Movement Analysis: A Review. *Prosthetics and Orthotics International*, 31(1):62–75, March 2007. ISSN 0309-3646, 1746-1553. doi: 10.1080/03093640600983949. URL http://poi.sagepub.com/content/31/1/62.

[14] H.J. Luinge, P.H. Veltink, and C.T.M. Baten. Estimation of orientation with gyroscopes and accelerometers. In *[Engineering in Medicine and Biology, 1999. 21st Annual Conference and the 1999 Annual Fall Meetring of the Biomedical Engineering Society] BMES/EMBS Conference, 1999. Proceedings of the First Joint*, volume 2, pages 844 vol.2–, Oct 1999. doi: 10.1109/IEMBS.1999.803999.

[15] Alberto Olivares Vicente. *Signal Processing of Magnetic and Inertial Sensor's Signals applied to Human Body Motion Monitoring*. PhD thesis, University of Granada, Granada, January 2013. URL `http://hera.ugr.es/tesisugr/21910947.pdf`.

[16] P.H. Veltink, H.B.J. Bussmann, W. de Vries, Wim L.J. Martens, and R.C. van Lummel. Detection of static and dynamic activities using uniaxial accelerometers. *Rehabilitation Engineering, IEEE Transactions on*, 4(4):375–385, Dec 1996. ISSN 1063-6528. doi: 10.1109/86.547939.

[17] B. Najafi, K. Aminian, A. Paraschiv-Ionescu, F. Loew, C.J. Bula, and P. Robert. Ambulatory system for human motion analysis using a kinematic sensor: monitoring of daily physical activity in the elderly. *Biomedical Engineering, IEEE Transactions on*, 50(6):711–723, June 2003. ISSN 0018-9294. doi: 10.1109/TBME.2003.812189.

[18] M. Ermes, J. Parkka, J. Mantyjarvi, and I. Korhonen. Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions. *Information Technology in Biomedicine, IEEE Transactions on*, 12(1):20–26, Jan 2008.

[19] O. Giggins, D. Kelly, and B. Caulfield. Evaluating rehabilitation exercise performance using a single inertial measurement unit. In *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2013 7th International Conference on*, pages 49–56, May 2013.

[20] J. Lupinski, V. Menon, P. Roh, S. Yuen, A. Valdevit, and J. Andrish. Measuring knee compliance to facilitate post-op ligament rehabilitation. In *Bioengineering Conference (NEBEC), 2011 IEEE 37th Annual Northeast*, pages 1–2, April 2011. doi: 10.1109/NEBC.2011.5778529.

[21] V. Bonnet, C. Mazza, P. Fraisse, and A. Cappozzo. Real-time estimate of body kinematics during a planar squat task using a single inertial measurement unit. *Biomedical Engineering, IEEE Transactions on*, 60 (7):1920–1926, July 2013. ISSN 0018-9294. doi: 10.1109/TBME.2013.2245131.

[22] Rui Zhang, F. Hoflinger, and L. Reindl. Inertial sensor based indoor localization and monitoring system for emergency responders. *Sensors Journal, IEEE*, 13(2):838–848, Feb 2013. ISSN 1530-437X. doi: 10.1109/JSEN.2012.2227593.

[23] T. Bennett, R. Jafari, and N. Gans. An extended kalman filter to estimate human gait parameters and walking distance. In *American Control Conference (ACC), 2013*, pages 752–757, June 2013. doi: 10.1109/ACC.2013.6579926.

[24] M. Mancini, C. Zampieri, P. Carlson-Kuhta, L. Chiari, and F. B. Horak. Anticipatory postural adjustments prior to step initiation are hypometric in untreated Parkinson's disease: an accelerometer-based approach. *European Journal of Neurology: The Official Journal of the European Federation of Neurological Societies*, 16(9):1028–1034, September 2009. ISSN 1468-1331. doi: 10.1111/j.1468-1331.2009.02641.x.

[25] L. Palmerini, L. Rocchi, S. Mellone, F. Valzania, and L. Chiari. Feature selection for accelerometer-based posture analysis in Parkinson's disease. *IEEE Transactions on Information Technology in Biomedicine*, 15(3):481–490, May 2011. ISSN 1089-7771. doi: 10.1109/TITB.2011.2107916.

[26] D. G. M. de Klerk, J. P. P. van Vugt, J. a. G. Geelen, and T. Heida. A long-term monitor including activity classification for motor assessment in Parkinson's disease patients. In Jos Vander Sloten, Pascal Verdonck, Marc Nyssen, and Jens Haueisen, editors, *4th European Conference of the International Federation for Medical and Biological Engineering*, number 22 in IFMBE Proceedings, pages 1706–1709. Springer Berlin Heidelberg, January 2009. ISBN 978-3-540-89207-6, 978-3-540-89208-3. URL http://link.springer.com/chapter/10.1007/978-3-540-89208-3_406.

[27] Kaveh Saremi, Jon Marehbian, Xiaohong Yan, Jean-Philippe Regnaux, Robert Elashoff, Bernard Bussel, and Bruce H. Dobkin. Reliability and Validity of Bilateral Thigh and Foot Accelerometry Measures of Walking in Healthy and Hemiparetic Subjects. *Neurorehabilitation and Neural Repair*, 20(2):297–305, June 2006. ISSN 1545-9683, 1552-6844. doi: 10.1177/1545968306287171. URL http://nnr.sagepub.com/content/20/2/297.

[28] Joe Verghese, Richard B. Lipton, Charles B. Hall, Gail Kuslansky, Mindy J. Katz, and Herman Buschke. Abnormality of gait as a predictor of non-alzheimer's dementia. *New England Journal of Medicine*, 347(22):1761–1768, 2002. doi: 10.1056/NEJMoa020441. URL http://dx.doi.org/10.1056/NEJMoa020441. PMID: 12456852.

[29] A. K. Bourke and G. M. Lyons. A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor. *Medical Engineering & Physics*, 30(1):84–90, January 2008. ISSN 1350-4533. doi: 10.1016/j.medengphy.2006.12.001. URL http://www.sciencedirect.com/science/article/pii/S1350453306002657.

[30] A.K. Bourke, P. van de Ven, M. Gamble, R. O'Connor, K. Murphy, E. Bogan, E. McQuade, P. Finucane, G. OLaighin, and J. Nelson. Assessment of waist-worn tri-axial accelerometer based fall-detection algorithms using continuous unsupervised activities. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pages 2782–2785, Aug 2010. doi: 10.1109/IEMBS.2010.5626364.

[31] A.K. Bourke, P.W.J. van de Ven, A. Chaya, G. Olaighin, and J. Nelson. Design and test of a long-term fall detection system incorporated into a custom vest for the elderly. In *Signals and Systems Conference, 208. (ISSC 2008). IET Irish*, pages 307–312, June 2008.

[32] eMarketer. 2 Billion Consumers Worldwide to Get Smartphones by 2016. URL http://www.emarketer.com/Article/2-Billion-Consumers-Worldwide-Smartphones-by-2016/1011694. [Accessed 5 April, 2015].

[33] The Michael J. Fox Foundation for Parkinson's Research (MJFF) and Intel Corporation. The Michael J. Fox Foundation and Intel Join Forces to Improve Parkinson's Disease Monitoring and Treatment through Advanced Technologies. URL http://goo.gl/PxQAof. [Accessed 20 March, 2015].

[34] Apple Inc. ResearchKit. URL http://www.apple.com/researchkit. [Accessed 20 March, 2015].

[35] Sage Bionetworks. mPower: Mobile Parkinson Disease Study. URL http://parkinsonmpower.org.

[36] J. Lenz and Alan S. Edelstein. Magnetic sensors and their applications. *Sensors Journal, IEEE*, 6(3):631–649, June 2006. ISSN 1530-437X. doi: 10.1109/JSEN.2006.874493.

[37] M.J. Thompson, M. Li, and D.A. Horsley. Low power 3-axis Lorentz force navigation magnetometer. In *Micro Electro Mechanical Systems*

(MEMS), 2011 IEEE 24th International Conference on, pages 593–596, Jan 2011. doi: 10.1109/MEMSYS.2011.5734494.

[38] SparkFun Electronics. *IMU Analog Combo Board 5DOF*, . URL https://www.sparkfun.com/products/retired/9268. [Accessed 17 March 2015].

[39] SparkFun Electronics. *Gyro Breakout Board IDG500*, . URL https://www.sparkfun.com/products/retired/9094. [Accessed 17 March 2015].

[40] Analog Devices. *ADXL335: Small, Low Power 3-Axis ±3G Accelerometer*, . URL http://www.analog.com/en/mems-sensors/mems-inertial-sensors/adxl335/products/product.html. [Accessed 17 March 2015].

[41] Analog Devices. *ADXL345: 3-Axis, ±2g/±4g/±8g/±16g Digital Accelerometer*, . URL http://www.analog.com/en/mems-sensors/mems-inertial-sensors/adxl345/products/product.html. [Accessed 17 March 2015].

[42] InvenSense. *IMU-3000 Triple Axis MotionProcessor™Gyroscope*. URL http://www.invensense.com/mems/gyro/imu3000.html. [Accessed 17 March 2015].

[43] SparkFun Electronics. *MicroMag 3-Axis Magnetometer*, . URL https://www.sparkfun.com/products/retired/244. [Accessed 17 March 2015].

[44] ALVIDI. *AVR ATxmega – Development Module*. URL http://www.alvidi.de/avr_xmodul_en.html. [Accessed 17 March 2015].

[45] James Diebel. Representing attitude: Euler angles, unit quaternions, and rotation vectors. Technical report, Stanford University, 2006.

[46] J.R. Raol. *Multi-Sensor Data Fusion with MATLAB®*. CRC Press, 2009. ISBN 9781439800058. URL https://books.google.de/books?id=7s8xpR-5r0UC.

[47] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME–Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[48] Gregory F. Welch. Kalman filter. In Katsushi Ikeuchi, editor, *Computer Vision*, pages 435–437. Springer US, 2014. ISBN 978-0-387-30771-8. doi: 10.1007/978-0-387-31439-6_716. URL http://dx.doi.org/10.1007/978-0-387-31439-6_716.

[49] Paul Zarchan and Howard Musoff. *Fundamentals of Kalman Filtering: A Practical Approach*. Progress in astronautics and aeronautics. Academic Press, 2009. ISBN 9781600867187. URL https://books.google.es/books?id=LUDKrQEACAAJ.

[50] M.S. Grewal and A.P. Andrews. *Kalman Filtering: Theory and Practice Using MATLAB*. Wiley, 2008. ISBN 9780470377802. URL http://books.google.es/books?id=J_fqMHOCzB8C.

[51] M.W. Spong and S. Hutchinson. *Robot Modeling and Control*. Wiley, 2005. ISBN 9780471649908. URL http://books.google.de/books?id=wGapQAAACAAJ.

[52] Derek Rowell. State-space representation of LTI systems. 2002. URL http://web.mit.edu/2.14/www/Handouts/StateSpace.pdf.

APPENDIX

MATLAB CODE

Listing 1: MATLAB® code file fusion_EKF.m

```
function [theta1, theta2] = fusion_EKF(gyro_thigh_y, ...
            gyro_shank_y, acc_thigh_x, acc_thigh_z, ...
            acc_shank_x, acc_shank_z, fs, l1, l2)

% FUNCTION fusion_EKF applies an extended Kalman filter
% in order to fuse the accelerometer and gyroscope data
% and thus obtain an accurate orientation estimate of
% the thighs and shanks.
%
% Input arguments:
% |_ 'gyro_thigh_y':  Row vector containing the angular
%                     rate of the thigh about the
%                     y-axis in radians per second.
% |_ 'gyro_shank_y':  Row vector containing the angular
%                     rate of the shank about the
%                     y-axis in radians per second.
% |_ 'acc_thigh_x':   Row vector containing the linear
%                     acceleration of the thigh along
%                     the x-axis in g.
% |_ 'acc_thigh_z':   Row vector containing the linear
%                     acceleration of the thigh along
%                     the z-axis in g.
% |_ 'acc_shank_x':   Row vector containing the linear
%                     acceleration of the shank along
%                     the x-axis in g.
% |_ 'acc_shank_z':   Row vector containing the linear
%                     acceleration of the shank along
%                     the z-axis in g.
```

```
% |_ 'frec':          Sampling frecuency in Hertz. Must
%                     be real positive.
% |_ 'l1':            Length of the thigh in m. Must be
%                     real positive.
% |_ 'l2':            Length of the shank in m. Must be
%                     real positive.
%
% Output:
% |_ 'theta1':        Row vector containing the thigh
%                     angle with respect to the x-axis
%                     of the world frame in radians.
% |_ 'theta2':        Row vector containing the shank
%                     angle with respect to the thigh
%                     in radians.
%
% IMPORTANT NOTE:     gyro_thigh_y, gyro_shank_y,
%                     acc_thigh_x, acc_thigh_z,
%                     acc_shank_x, and acc_shank_z
%                     must have the same length.
%                     Otherwise, an error will
%                     be returned.
% -----------------------------------------------------
% Authors:            Robin Weiss
% Entity:             University of Applied Sciences
%                     Munster, Munster, Germany
% Last modification:  01/05/2015
% -----------------------------------------------------

% 1) Check input arguments.
if ~isequal(length(gyro_thigh_y), ...
            length(gyro_shank_y), ...
            length(acc_thigh_x), ...
            length(acc_thigh_z), ...
            length(acc_shank_x), ...
            length(acc_shank_z))
    error(['Input arguments ''gyro_thigh_y'', ', ...
           '''acc_thigh_x'', ''acc_thigh_z'', ', ...
           '''acc_shank_x'', ''acc_shank_z'', ', ...
           'must have the same length.']);
end
```

```
if (fs <= 0 || ~isreal(fs))
    error(['Input␣argument␣''fs''␣must␣be␣real', ...
           'positive.']);
end

if (l1 <= 0 || ~isreal(l1))
    error(['Input␣argument␣''a1''␣must␣be␣real', ...
          '␣positive.']);
end

if (l2 <= 0 || ~isreal(l2))
    error(['Input␣parameter␣''a2''␣must␣be␣real', ...
           '␣positive.']);
end

% 2) Import GaitWatch functions library. All existing
%    functions have to be called using
%    'gw.functionName'.
gw = gwLibrary;

% 3) Compute the sampling period and the length of the
%    signal vectors.
Ts = 1 / fs;
len = length(gyro_thigh_y);

% % 5) Compute intensity level.
% lwin_fsd = 20;
% threshold_fsd = 3;
% shift_fsd = 19;
% input_signal = sqrt(acc_shank_x.^2+acc_shank_z.^2);
% [V_fsd, T_fsd] = gw.fsd(input_signal, lwin_fsd, ...
%                         shift_fsd, 512, threshold_fsd);
%
% % Determine marker signal.
% [marker, ~] = gw.compEstMark(V_fsd, T_fsd, ...
%                              input_signal, lwin_fsd, ...
%                              shift_fsd);
marker = ones(1, len);

% INITIALISATION OF PARAMETERS %

```

```
% 6) Compute mean of the first two seconds of the
%    gyroscope signals.
mu1 = mean(gyro_thigh_y(1:2*fs));
mu2 = mean(gyro_shank_y(1:2*fs));

% 7) Initialise the state vector.
x = [0, -(l1+l2), -pi/2, 0, 0, 0, 0, 0, mu1, mu2]';

% 8) Map gyroscope signals to measurement vector.
%    Column k represents the measurement vector at time
%    step k.
z = [gyro_thigh_y; gyro_shank_y; zeros(1, len)];

% 9) Initialise the error covariance matrix.
P = diag(ones(1, 10)/3);

% 10) Define the measurement matrix.
H = [0 0 0 1 0 0 0 0 1 0; ...
     0 0 0 1 0 0 1 0 0 1; ...
     0 0 1 0 0 1 0 0 0 0];

% 11) Define process noise covariance matrix.
sigma_d = 0.9;
sigma_t1 = 0.9;
sigma_t2 = 0.9;
sigma_b = 0.9;
Q = [...
sigma_d 0 0           0           0           0 0 0 0 0; ...
0 sigma_d 0           0           0           0 0 0 0 0; ...
0 0 sigma_t1^9/9 sigma_t1^4/4 sigma_t1^5/5 0 0 0 0 0; ...
0 0 sigma_t1^4/4 sigma_t1^3/3 sigma_t1^2/2 0 0 0 0 0; ...
0 0 sigma_t1^5/5 sigma_t1^2/2   sigma_t1   0 0 0 0 0; ...
0 0 0 0 0 sigma_t2^9/9 sigma_t2^4/4 sigma_t2^5/5 0 0; ...
0 0 0 0 0 sigma_t2^4/4 sigma_t2^3/3 sigma_t2^2/2 0 0; ...
0 0 0 0 0 sigma_t2^5/5 sigma_t2^2/2   sigma_t2   0 0; ...
0 0 0 0 0       0             0          0 sigma_b 0; ...
0 0 0 0 0       0             0          0 0 sigma_b];

% 12) Compute sample variance of the first
% two seconds of the gyroscope signals.
sigma_1 = var(gyro_thigh_y(1:2*fs));
```

```
sigma_2 = var(gyro_shank_y(1:2*fs));

% 12) Define measurement noise covariance matrix.
sigma_f = 0.9;
sigma_s = 0.9;
R = [sigma_1    0       0; ...
        0    sigma_2    0; ...
        0       0    sigma_s];

% 13) Define matrix function F.
    function F_k = F
        F_k = ...
[0, 0, 0, -l2 * sin(x(3)) - l2 * sin(x(3) + x(6)), ...
 0, 0, -l2 * sin(x(3) + x(6)), 0, 0, 0; ...
 0, 0, 0, l1 * cos(x(3)) + l2 * cos(x(3) + x(6)), ...
 0, 0, l2 * cos(x(3) + x(6)), 0, 0, 0; ...
 0, 0, 0, 1, 0, 0, 0, 0, 0, 0; ...
 0, 0, 0, 0, 1, 0, 0, 0, 0, 0; ...
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0; ...
 0, 0, 0, 0, 0, 0, 1, 0, 0, 0; ...
 0, 0, 0, 0, 0, 0, 0, 1, 0, 0; ...
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0; ...
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0; ...
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0];
    end

% 14) Initialise output vectors.
theta1 = zeros(1, len);
theta2 = zeros(1, len);

% 15) Filter loop.
for i=1:1:len

    % Set sigma3 in measurement noise covariance matrix
    % according to motion intensity.
    if marker(i)==1
       R(3, 3) = sigma_f;
    end

    if marker(i)==0
       R(3, 3) = sigma_s;
```

```
    end

    % TIME UPDATE %

    % Compute fundamental matrix.
    Phi = eye(10) + F * Ts;

    % Compute a priori state estimate.
    x = Phi * x;

    % Compute a priori error covariance matrix.
    P = Phi * P * Phi' + Q;

    % CORRECT SENSOR READINGS %

    % Compute acceleration due to motion.
    ax = -l1 * (x(4)^2 * cos(x(3)) + x(5) * sin(x(3))) ...
         - l2 * ((x(4) + x(7))^2 * cos(x(3) + x(6)) ...
         + (x(5) + x(8)) * sin(x(3) + x(6)));
    az = -l1 * (x(5) * cos(x(3)) - x(4)^2 * sin(x(3))) ...
         - l2 * (x(5) + x(8)) * cos(x(3) + x(6)) ...
         + ((x(4) + x(7))^2 * sin(x(3) + x(6)));

    % Compute transformation matrix
    Tz = [cos(x(3) + x(6) - 2 * pi), 0, ...
          sin(x(3) + x(6) - 2 * pi); 0, 1, 0; ...
         -sin(x(3) + x(6) - 2 * pi), 0, ...
          cos(x(3) + x(6) - 2 * pi)];

    % Rotate acceleration to body frame.
    a = Tz * [ax; 0; az];

    % Compute gravity estimate.
    g = [acc_shank_x(i); 0; acc_shank_z(i)] ...
         - a;

    % Compute corrected angle estimate and update
    % measurement vector.
    z(3, i) = atan2(g(3), g(1));

    % MEASUREMENT UPDATE %
```

```

    % Compute Kalman gain.
    K = P * H' / (H * P * H' + R);

    % Compute a posteriori estimate.
    x = x + K * (z(:, i) - H * x);

    % Update error covariance matrix.
    P = (eye(10) - K * H) / P;

    % Map internal states to output vector.
    theta1(i) = x(3);
    theta2(i) = x(6);

end

end
```

Listing 2: MATLAB® code file EKF_test.m

```
clear all; close all; clc;

load('data.mat');

[pitch_EKF_right_thigh, pitch_EKF_right_shank] = ...
            fusion_EKF(g_Y_right_thigh_1_C', ...
                       g_Y_right_shank_1_C', ...
                       a_X_right_thigh_1_C', ...
                       a_Z_right_thigh_1_C', ...
                       a_X_right_shank_1_C', ...
                       a_Z_right_shank_1_C', ...
                       f, 0.5, 0.5);

n = 500;

figure();
hold on
plot(time(1:n), pitch_acc_right_shank(1:n), ...
     time(1:n), pitch_gyro_right_shank(1:n), ...
     time(1:n), pitch_KF_right_shank(1:n), ...
     time(1:n), pitch_GKF_right_shank(1:n), ...
     time(1:n), pitch_EKF_right_shank(1:n));
```

```
title('Pitch␣angle␣of␣the␣right␣shank␣-␣Comparison');
xlabel('Time␣in␣seconds');
ylabel('Pitch␣in␣degrees');
legend('Accelerometer-based', ...
       'Integration␣of␣angular␣rate', ...
       'Kalman␣filter', 'Gated␣Kalman␣filter', ...
       'Extended␣Kalman␣filter');
```