Linköping studies in science and technology. Dissertations.
No. 1368

Sensor Fusion and Calibration of Inertial Sensors, Vision, Ultra-Wideband and GPS

Jeroen Hol

Department of Electrical Engineering
Linköping University, SE–581 83 Linköping, Sweden

Linköping 2011

Cover illustration: A selection of Xsens devices used in this theses. From left to right: an MTi OEM, an MTx, an MTi with UWB transmitter and an MTi-G.

Linköping studies in science and technology. Dissertations.
No. 1368

Sensor Fusion and Calibration of Inertial Sensors, Vision, Ultra-Wideband and GPS

Jeroen Hol

hol@isy.liu.se
www.control.isy.liu.se
Division of Automatic Control
Department of Electrical Engineering
Linköping University
SE–581 83 Linköping
Sweden

ISBN 978-91-7393-197-7 ISSN 0345-7524

Copyright © 2011 Jeroen Hol

Printed by LiU-Tryck, Linköping, Sweden 2011

To Nantje!

Abstract

The usage of inertial sensors has traditionally been confined primarily to the aviation and marine industry due to their associated cost and bulkiness. During the last decade, however, inertial sensors have undergone a rather dramatic reduction in both size and cost with the introduction of *micro-machined electromechanical system* (MEMS) technology. As a result of this trend, inertial sensors have become commonplace for many applications and can even be found in many consumer products, for instance smart phones, cameras and game consoles. Due to the drift inherent in inertial technology, inertial sensors are typically used in combination with aiding sensors to stabilize and improve the estimates. The need for aiding sensors becomes even more apparent due to the reduced accuracy of MEMS inertial sensors.

This thesis discusses two problems related to using inertial sensors in combination with aiding sensors. The first is the problem of *sensor fusion*: how to combine the information obtained from the different sensors and obtain a good estimate of position and orientation. The second problem, a prerequisite for sensor fusion, is that of *calibration*: the sensors themselves have to be calibrated and provide measurement in known units. Furthermore, whenever multiple sensors are combined additional calibration issues arise, since the measurements are seldom acquired in the same physical location and expressed in a common coordinate frame. Sensor fusion and calibration are discussed for the combination of inertial sensors with cameras, *ultra-wideband* (UWB) or *global positioning system* (GPS).

Two setups for estimating position and orientation in real-time are presented in this thesis. The first uses inertial sensors in combination with a camera; the second combines inertial sensors with UWB. Tightly coupled sensor fusion algorithms and experiments with performance evaluation are provided. Furthermore, this thesis contains ideas on using an optimization based sensor fusion method for a multi-segment inertial tracking system used for human motion capture as well as a sensor fusion method for combining inertial sensors with a dual GPS receiver.

The above sensor fusion applications give rise to a number of calibration problems. Novel and easy-to-use calibration algorithms have been developed and tested to determine the following parameters: the magnetic field distortion when an *inertial measurement unit* (IMU) containing magnetometers is mounted close to a ferro-magnetic object, the relative position and orientation of a rigidly connected camera and IMU, as well as the clock parameters and receiver positions of an indoor UWB positioning system.

Populärvetenskaplig sammanfattning

Användningen av tröghetssensorer har traditionellt varit begränsad främst till marin- och flygindustrin på grund av sensorernas kostnad som storlek. Under det senaste decenniet har tröghetssensorer storlek såväl som kostand genomgått en dramatisk minskning tack vara införandet av *micro-machined electromechanical system* (MEMS) teknik. Som ett resultat av denna trend, har de blivit vanliga i många ny tillämpningar och kan nu även hittas i många konsumentprodukter, till exempel smarta telefoner, kameror och spelkonsoler. På grund av sin drift används tröghetssensorer vanligtvis i kombination med andra sensorer för att stabilisera och förbättra resultatet. Behovet av andra sensorer blir ännu mer uppenbart på grund av minskad noggrannhet hos tröghetssensorer av MEMS-typ.

Denna avhandling diskuterar två problem relaterade till att använda tröghetssensorer i kombination med andra sensorer. Det första problemet är *sensorfusion*: hur kombinerar man information från olika sensorer för att få en bra skattning av position och orientering. Det andra problemet, en förutsättning för sensorfusion, är *kalibrering*: sensorerna själva måste vara kalibrerade och ge mätningar i kända enheter. När flera sensorer kombineras kan ytterligare kalibreringsproblem uppstå, eftersom mätningarna sällan utförs på samma fysiska position eller uttrycks i ett gemensam koordinatsystem. Sensorfusion och kalibrering diskuteras för kombinationen av tröghetssensorer med kameror, *ultra-wideband* (UWB) eller *global positioning system* (GPS).

Två tillämpningar för att skatta läge och orientering i realtid härleds och presenteras i den här avhandlingen. Den första använder tröghetssensorer i kombination med en kamera, den andra kombinerar tröghetssensorer med UWB. Vidare presenteras resultaten från ett flertal experiment för att kunna utvärdera prestanda. Dessutom innehåller denna avhandling idéer om hur man använder en optimeringsbaserad sensorfusionsmetod för att bestämma mänsklig rörelse, samt en sensorfusionsmetod för att kombinera tröghetssensorer med två GPS-mottagare.

Ovanstående sensorfusionstillämpningar ger upphov till ett flertal kalibreringsproblem. Nya och lättanvända kalibreringsalgoritmer har utvecklats och testats för att fastställa följande parametrar: distorsion av magnetfältet när en *inertial measurement unit* (IMU) som innehåller magnetometrar är monterad nära ett ferro-magnetiskt objekt, relativa läget och orienteringen för en kamera och IMU som är fast monterade i förhållande till varandra, samt klockparametrar och mottagarpositioner för ett UWB positioneringssystem för inomhusbruk.

Acknowledgments

I find it very hard to believe that I managed to get this far. It seems already a long time ago that I started my PhD studies at the automatic control group in Linköping. I received a warm welcome from Fredrik Gustafsson and Thomas Schön, my supervisors, who invited me to Sweden. Thank you for doing so! I really appreciate our many discussions and the frequent application of red pencil. I really hope to continue our fruitful collaboration in the future. Furthermore I would like to thank Lennart Ljung, Svante Gunnarsson, Ulla Salaneck, Åsa Karmelind and Ninna Stensgård for enabling me to work in such a pleasant atmosphere.

Furthermore I would like to thank Per Slycke, Henk Luinge and the rest of the Xsens team for the successful collaboration over the years and for giving me the opportunity to finish the second half of my PhD at Xsens. In my opinion, the combination of real world problems of industry with the theoretical approach from academia provided an excellent environment for my research.

Living in foreign country with an unknown language is always difficult in the beginning. I would like to thank the entire control group for making learning the Swedish language such fun. My special thanks go to David Törnqvist and Johan Sjöberg. You introduced me to the Swedish culture, listened to my stories and took care of distraction from work. At Xsens would like to thank my roommates of our not-so-silent room, Arun Vydhyanathan, Manon Kok, Martin Schepers and Raymond Zandbergen for our spontaneous research meetings about all the aspects of life.

This thesis has been proofread by Manon Kok, Miao Zhang, Arun Vydhyanathan and Zoran Sjanic. You kept an eye on the presentation of the material, asked difficult questions and improved the quality a lot, which is really appreciated. Writing the thesis would not have been so easy without the LaTeX support from Gustaf Hendeby, Henrik Tidefelt and David Törnqvist. Thank you!

Parts of this work have been supported by MATRIS (Markerless real-time Tracking for Augmented Reality Image), a sixth framework research program within the European Union, CADICS (Control, Autonomy, and Decision-making in Complex Systems), a Linneaus Center funded by the Swedish Research Council (VR) and the Strategic Research Center MOVIII, funded by the Swedish Foundation for Strategic Research (SSF). These are hereby gratefully acknowledged.

Finally, I am grateful to my parents for their never ending support and encouragement and to Nantje for being the love of my life. I finally have more time to spend with you.

Enschede, June 2011

Jeroen Hol

Contents

Notation

In this thesis, scalars are denoted with lowercase letters (u, ρ), geometric vectors with bold lowercase letters ($\boldsymbol{b}, \boldsymbol{\omega}$), state and parameter vectors with bold lowercase letters ($\boldsymbol{x}, \boldsymbol{\theta}$), quaternions with lowercase letters (q, e), and matrices with capitals (A, R). Superscripts denote rotations and in which frame a quantity is resolved ($q^{bn}, \boldsymbol{b}^n$). Subscripts are used for annotations and indexing ($\boldsymbol{x}_t, \boldsymbol{\omega}_{ie}, y_{\mathrm{u}}$).

Coordinate frames are used to denote the frame in which a quantity is resolved, as well as to denote the origin of the frame, e.g., $\boldsymbol{b}^n$ is the position of the body frame (b-frame) expressed in the navigation frame (n-frame) and $\boldsymbol{t}^b$ is the position of the transmitter resolved in the b-frame. Furthermore, q^{bn}, $\boldsymbol{\varphi}^{bn}$ and R^{bn} are the unit quaternion, the rotation vector and the rotation matrix, respectively. They parameterize the rotation from the n-frame to b-frame and can be used interchangeable. More details can be found in Chapter 4.

A complete list of used abbreviations and acronyms, a list of coordinate frames, together with a list of mathematical operators and sets can be found in the tables on the next pages.

Coordinate frames and locations

Notation	Meaning
b	Body frame.
c	Camera frame.
e	Earth frame.
ι	Image frame.
i	Inertial frame.
n	Navigation frame.
t	Transmitter location.

Some sets

Notation	Meaning
$\mathbb{Z}$	Integer numbers.
$\mathcal{Q}_s$	Scalar quaternions.
$\mathcal{Q}_1$	Unit-length quaternions.
$\mathcal{Q}_v$	Vector quaternions.
$\mathbb{R}$	Real numbers.
SO(3)	Special orthogonal group, 3-dimensional.
S(2)	Sphere, 2-dimensional.

Operators

Notation	Meaning
arg max	Maximizing argument.
arg min	Minimizing argument.
Cov	Covariance.
$[\cdot\times]$	Cross product matrix.
diag	Diagonal matrix.
$\mathrm{D}_x\,\cdot$	Jacobian matrix.
d	Differential operator.
$\cdot^{-1}$	Inverse.
$\cdot^{-T}$	Transposed inverse.
$\otimes$	Kronecker product.
$\|\cdot\|_2$	2-Norm.
$\|\cdot\|_W$	Weighted norm.
$\cdot^c$	Quaternion conjugate.
$\odot$	Quaternion multiplication.
$\cdot^L$	Left quaternion multiplication.
$\cdot^R$	Right quaternion multiplication.
tr	Trace.
$\cdot^T$	Transpose.
vec	Vectorize operator.

Abbreviations and acronyms

Abbreviation	Meaning
i.i.d.	Independently and identically distributed.
w.r.t.	With respect to.
2D	Two dimensional.
3D	Three dimensional.
AR	Augmented reality.
CAD	Computer aided design.
DGPS	Differential GPS.
DOF	Degrees of freedom.
EKF	Extended Kalman filter.
GNNS	Global navigation satellite system.
GPS	Global positioning system.
IMU	Inertial measurement unit.
KF	Kalman filter.
KKT	Karush-Kuhn-Tucker.
MAP	Maximum a posteriori.
MEMS	Micro-machined electromechanical system.
ML	Maximum likelihood.
NLOS	Non line of sight.
PDF	Probability density function.
PF	Particle filter.
PVT	Position, velocity and time.
RMSE	Root mean square error.
SBAS	Satellite-based augmentation system.
SLAM	Simultaneous localization and mapping.
TDOA	Time difference of arrival.
TOA	Time of arrival.
UWB	Ultra-wideband.

Part I

Background

1 Introduction

This chapter gives an introduction to the thesis by briefly explaining the setting in which the work has been carried out, presenting the contributions in view of a problem formulation and providing some reading directions.

1.1 Motivation

Inertial sensors have been around for decades. Traditionally they have been used in aviation and marine industry, where they are used for navigation and control purposes. That is, the inertial sensors are used to determine the instantaneous position and orientation of a platform. This hardware typically is very accurate as well as rather bulky and expensive.

The introduction of *micro-machined electromechanical system* (MEMS) technology has lead to smaller and cheaper inertial sensors, and this trend is still continuing. Nowadays inertial sensors have become more ubiquitous and are used in consumer applications. They can be found in cars, gaming consoles and in every self-respecting smart-phone. Some examples of platforms containing inertial sensors are shown in Figure 1.1.

Currently, the major disadvantage of MEMS components is the reduced performance in terms of accuracy and stability, impeding autonomous navigation. As a result, inertial sensors are typically used in combination with aiding sources. As there is no generic solution for *three dimensional* (3D) tracking in general (Welch and Foxlin, 2002), the aiding source is chosen depending on the application. Examples of aiding sources include actual sensors such as vision, *ultra-wideband* (UWB) and the *global positioning system* (GPS). Furthermore, constraints, for instance from bio-mechanical models, can also be used as aiding sources.

(a) Airbus A380

(b) Volvo V70

(c) iPhone

(d) Nintendo Wii

Figure 1.1: *Examples of platforms containing inertial sensors. By courtesy of Airbus, Volvo cars, Apple and Nintendo.*

Combined with a suitable aiding source, inertial sensors form the basis for a powerful tracking technology which has been successfully applied in a wide range of applications. Examples include navigation of autonomous vehicles, motion capture and *augmented reality* (AR), see Figure 1.2. Navigation of autonomous vehicles, for aerial, ground or marine applications, is a more traditional application of inertial sensors. They are used, typically in combination with GPS, to determine the real-time position and orientation of the platform. This is in turn used in the control loop to stabilize the platform and to make sure that it follows a predetermined path.

For motion capture applications, small *inertial measurement units* (IMUs) are attached to body segments. Such a system can measure the orientation and relative position of the individual segments and thereby the exact movement of a person or an animal. In a health-care setting, this allows clinical specialists to analyze, monitor and train the movements of a patient. Similarly, it allows athletes to study and improve their technique. In the movie and gaming industry, the recorded movements of an actor form the basis for special effects or game characters.

One of the main ideas of AR is to overlay a real scene with computer generated graphics in real-time. This can be accomplished by showing the virtual objects

(a) Autonomous vehicles

(b) Motion capture

(c) Augmented reality (AR)

Figure 1.2: *Examples of applications using inertial technology. By courtesy of Tartan Racing and Frauenhofer IGD.*

on see-through head-mounted displays or superimposing them on live video imagery. Inertial sensors combined with vision can be used to determine the position and orientation of the camera. This knowledge is required to position and align the virtual objects correctly on top of the real world and to ensure that they appear to stay in the same location regardless of the camera movement.

1.2 Problem formulation

Inertial sensors are small and unobtrusive devices which are used in a wide range of applications. However, they have to be used together with some kind of aiding sensor. This immediately brings us to the problem of sensor fusion: how can one combine the information from the sensors and models to obtain a good estimate and extract all the information from the available measurements.

The second problem is frequently overlooked. Whenever multiple sensors are combined one also has to deal with additional calibration issues. Quantities are seldom measured at the same position and in the same coordinate frame, implying that the alignment, the relative position and/or orientation of the sensors, has to be determined. Sometimes this can be taken from a technical drawing, but frequently this has to be determined in-situ from the sensor measurements. Such

a procedure is called a calibration procedure, and has to be designed for every combination of sensors. A good calibration is a prerequisite to do sensor fusion. Therefore, it is very important to have simple and efficient calibration procedures, preferable without additional hardware.

Both sensor fusion and calibration are discussed in the thesis. We focus on the combination of inertial sensors with vision, UWB or GPS. These applications are discussed individually, starting from a common theoretical background.

1.3 Motivating example

For outdoor applications, GPS is the standard choice of aiding source to combine with inertial sensors. However, for indoor applications this not a viable option due to problematic reception of GPS signals as well as reflections. As an alternative one can use a UWB setup. Such a system consists of a network of synchronized receivers that track a (potentially large) number of small transmitters. Integrating one of these transmitters with an inertial sensor results in a 6 gls-DOF general purpose tracking system for indoor applications. The sensor unit is shown in Figure 1.3.

Figure 1.3: *An Xsens prototype sensor unit, integrating an IMU and an UWB transmitter into a single housing.*

Figure 1.4 shows a comparison of the IMU, UWB and the combined IMU/UWB estimates. As is clearly seen, the IMU performance rapidly decreases due to drift, especially for position and velocity. UWB provides only position measurements; velocity and orientation are not available. The inertial/UWB combination provides superior estimates. It gives accurate and stable estimates of position, velocity as well as orientation. More details can be found in Chapter 7.

The above inertial/UWB technology is commercially available as a part of MVN Motiongrid by Xsens Technologies (Xsens Technologies, 2010).

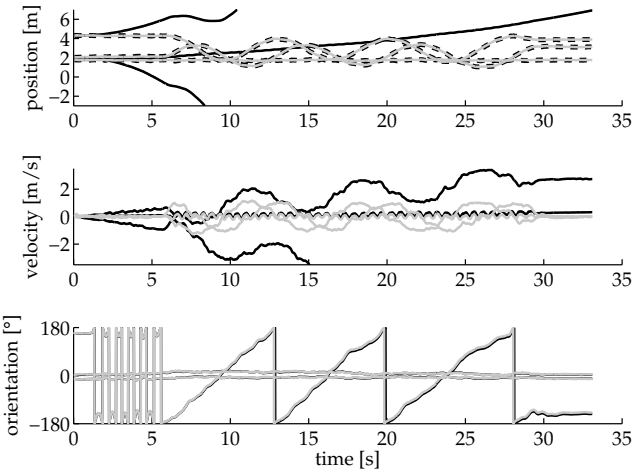


Figure 1.4: *Position, velocity and orientation estimates. Shown are trajectories using the IMU (–), UWB (- -) and the IMU/UWB combination (–).*

1.4 Contributions

The main contributions of this thesis are, listed in order of appearance:

- A concise overview of inertial sensors, vision, UWB and GPS and the kinematics linking their measurements.
- The idea of applying optimization based sensor fusion methods to multi-segment inertial tracking system.
- A calibration algorithm to calibrate the magnetic field of an IMU mounted close to a ferro-magnetic object.
- The development, testing and evaluation of a real-time pose estimation system based on sensor fusion of inertial sensors and vision.
- An easy-to-use calibration method to determine the relative position and orientation of a rigidly connected camera and IMU.
- A calibration method to determine the clock parameters and receiver positions of a UWB positioning system.
- The development, testing and evaluation of a tightly coupled 6 *degrees of freedom* (DOF) tracking system based on sensor fusion of inertial sensors and UWB.
- A sensor fusion method for combining inertial sensors and dual GPS.

1.5 Publications

Parts of this thesis have been previously published. Material on sensor fusion using inertial sensors and vision (Chapter 6) has been published in

> J. D. Hol, T. B. Schön, and F. Gustafsson. Modeling and calibration of inertial and vision sensors. *International Journal of Robotics Research*, 29(2):231–244, feb 2010b. doi: 10.1177/0278364909356812.
>
> J. D. Hol, T. B. Schön, and F. Gustafsson. A new algorithm for calibrating a combined camera and IMU sensor unit. In *Proceedings of 10th International Conference on Control, Automation, Robotics and Vision*, pages 1857–1862, Hanoi, Vietnam, Dec. 2008b. doi: 10.1109/ICARCV.2008.4795810.
>
> J. D. Hol, T. B. Schön, and F. Gustafsson. Relative pose calibration of a spherical camera and an IMU. In *Proceedings of 8th International Symposium on Mixed and Augmented Reality*, pages 21–24, Cambridge, UK, Sept. 2008a. doi: 10.1109/ISMAR.2008.4637318.
>
> F. Gustafsson, T. B. Schön, and J. D. Hol. Sensor fusion for augmented reality. In *Proceedings of 17th International Federation of Automatic Control World Congress*, Seoul, South Korea, July 2008. doi: 10.3182/20080706-5-KR-1001.3059.
>
> J. D. Hol. *Pose Estimation and Calibration Algorithms for Vision and Inertial Sensors*. Licentiate thesis no 1379, Department of Electrical Engineering, Linköping University, Sweden, May 2008. ISBN 978-91-7393-862-4.
>
> J. D. Hol, T. B. Schön, H. Luinge, P. J. Slycke, and F. Gustafsson. Robust real-time tracking by fusing measurements from inertial and vision sensors. *Journal of Real-Time Image Processing*, 2(2):149–160, Nov. 2007. doi: 10.1007/s11554-007-0040-2.
>
> J. Chandaria, G. Thomas, B. Bartczak, K. Koeser, R. Koch, M. Becker, G. Bleser, D. Stricker, C. Wohlleber, M. Felsberg, F. Gustafsson, J. D. Hol, T. B. Schön, J. Skoglund, P. J. Slycke, and S. Smeitz. Real-time camera tracking in the MATRIS project. *SMPTE Motion Imaging Journal*, 116(7–8):266–271, Aug. 2007a
>
> J. D. Hol, T. B. Schön, F. Gustafsson, and P. J. Slycke. Sensor fusion for augmented reality. In *Proceedings of 9th International Conference on Information Fusion*, Florence, Italy, July 2006b. doi: 10.1109/ICIF.2006.301604.

Inertial and UWB (Chapter 7) related publications are

> J. D. Hol, T. B. Schön, and F. Gustafsson. Ultra-wideband calibration for indoor positioning. In *Proceedings of IEEE International Confer-*

ence on Ultra-Wideband, volume 2, pages 1–4, Nanjing, China, Sept. 2010c. doi: 10.1109/ICUWB.2010.5616867.

J. D. Hol, H. J. Luinge, and P. J. Slycke. Positioning system calibration. Patent Application, Mar. 2010a. US 12/749494.

J. D. Hol, F. Dijkstra, H. Luinge, and T. B. Schön. Tightly coupled UWB/IMU pose estimation. In *Proceedings of IEEE International Conference on Ultra-Wideband*, pages 688–692, Vancouver, Canada, Sept. 2009. doi: 10.1109/ICUWB.2009.5288724. Best student paper award.

J. D. Hol, F. Dijkstra, H. J. Luinge, and P. J. Slycke. Tightly coupled UWB/IMU pose estimation system and method. Patent Application, Feb. 2011. US 2011/0025562 A1.

Outside the scope of this thesis fall the following conference papers.

G. Hendeby, J. D. Hol, R. Karlsson, and F. Gustafsson. A graphics processing unit implementation of the particle filter. In *Proceedings of European Signal Processing Conference*, Poznań, Poland, Sept. 2007.

J. D. Hol, T. B. Schön, and F. Gustafsson. On resampling algorithms for particle filters. In *Proceedings of Nonlinear Statistical Signal Processing Workshop*, Cambridge, UK, Sept. 2006a. doi: 10.1109/NSSPW.2006.4378824.

1.6 Outline

This thesis consists of two parts. Part I contains the background material to the applications presented in Part II.

Part I starts with a brief introduction to the field of estimation. It provides some notion about the concepts of parameters estimation, system identification and sensor fusion. Chapter 3 contains sensor specific introductions to inertial sensors, vision, UWB and GPS. The measurements, calibration techniques and standard usage for each kind of sensor are discussed. Chapter 4, on kinematics, provides the connection between these sensors.

Part II discusses sensor fusion for a number of applications, together with associated calibration problems. It starts with orientation estimation using standalone IMUs in Chapter 5. Chapter 6 is about estimating position and orientation using vision and inertial sensors. Pose estimation using UWB and inertial sensors is the topic of Chapter 7. The combination of GPS and inertial sensors is discussed in Chapter 8. Finally, Chapter 9 concludes this thesis and gives possible directions for future work.

2

Estimation theory

This chapter provides a brief introduction to the field of estimation. It provides some notion about the concepts of parameter estimation, system identification and sensor fusion.

2.1 Parameter estimation

In parameter estimation the goal is to infer knowledge about certain parameters of interest from a set of measurements. Mathematically speaking, one wants to find a function, the estimator, which maps the measurements to a parameter value, the estimate. In statistics literature (e.g. Trees, 1968; Hastie et al., 2009) many different estimators can be found, among which the *maximum likelihood* (ML) estimator and the *maximum a posteriori* (MAP) estimator are of particular interest.

The ML estimate is defined in the following way

$$\hat{\theta}_{\text{ML}} = \arg\max_{\theta} p(y|\theta). \tag{2.1}$$

That is, the estimate $\hat{\theta}_{\text{ML}}$ is the value of the parameter vector θ at which the likelihood function $p(y|\theta)$ attains its maximum. Here $p(y|\theta)$ is the *probability density function* (PDF) which specifies how the measurements y are distributed when the parameter vector θ is given. Example 2.1 shows how an ML estimate for the mean and the covariance of a Gaussian distribution is derived.

2.1 Example: Gaussian parameter estimation

The PDF and the corresponding log-likelihood of a Gaussian random variable $\boldsymbol{y}$ with mean $\boldsymbol{\mu} \in \mathbb{R}^m$ and covariance $\Sigma \in \mathbb{R}^{m \times m}$, $\Sigma \succ 0$, denoted as $\boldsymbol{y} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ is given by

$$p(\boldsymbol{y}|\boldsymbol{\mu}, \Sigma) = (2\pi)^{-\frac{m}{2}} (\det \Sigma)^{-\frac{1}{2}} \exp\left(-\tfrac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu})^T \Sigma^{-1}(\boldsymbol{y}-\boldsymbol{\mu})\right), \tag{2.2a}$$

$$\log p(\boldsymbol{y}|\boldsymbol{\mu}, \Sigma) = -\tfrac{m}{2}\log(2\pi) - \tfrac{1}{2}\log\det\Sigma - \tfrac{1}{2}(\boldsymbol{y}-\boldsymbol{\mu})^T \Sigma^{-1}(\boldsymbol{y}-\boldsymbol{\mu}), \tag{2.2b}$$

where the dependence on the parameters $\boldsymbol{\mu}$ and Σ has been denoted explicitly.

Given a set of N independent samples $\boldsymbol{y}_{1:N} = \{\boldsymbol{y}_n\}_{n=1}^N$ drawn from $\boldsymbol{y}$, the ML estimate of the parameters $\boldsymbol{\theta} = \{\boldsymbol{\mu}, \Sigma\}$ is now given as

$$\begin{aligned}
\hat{\boldsymbol{\theta}} &= \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{y}_{1:N}|\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \prod_{n=1}^N p(\boldsymbol{y}_n|\boldsymbol{\theta}) \\
&= \arg\min_{\boldsymbol{\theta}} \sum_{n=1}^N -\log p(\boldsymbol{y}_n|\boldsymbol{\theta}) \\
&= \arg\min_{\boldsymbol{\theta}} \frac{Nm}{2}\log 2\pi + \frac{N}{2}\log\det\Sigma + \frac{1}{2}\operatorname{tr}\left(\Sigma^{-1}\sum_{n=1}^N (\boldsymbol{y}_n-\boldsymbol{\mu})(\boldsymbol{y}_n-\boldsymbol{\mu})^T\right).
\end{aligned} \tag{2.3}$$

Minimizing *with respect to* (w.r.t.) $\boldsymbol{\theta}$ yields $\hat{\boldsymbol{\theta}} = \{\hat{\boldsymbol{\mu}}, \hat{\Sigma}\}$ with

$$\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{n=1}^N \boldsymbol{y}_n, \qquad \hat{\Sigma} = \frac{1}{N}\sum_{n=1}^N (\boldsymbol{y}_n - \hat{\boldsymbol{\mu}})(\boldsymbol{y}_n - \hat{\boldsymbol{\mu}})^T, \tag{2.4}$$

which are the well-known sample mean and sample covariance.

The MAP estimate is defined in the following way

$$\hat{\boldsymbol{\theta}}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{\theta}|\boldsymbol{y}). \tag{2.5}$$

That is, the estimate $\hat{\boldsymbol{\theta}}_{\text{MAP}}$ is the value of the parameter $\boldsymbol{\theta}$ at which the posterior density $p(\boldsymbol{\theta}|\boldsymbol{y})$ attains its maximum. Bayes' formula expands $p(\boldsymbol{\theta}|\boldsymbol{y})$ as

$$p(\boldsymbol{\theta}|\boldsymbol{y}) = \frac{p(\boldsymbol{\theta}, \boldsymbol{y})}{p(\boldsymbol{y})} = \frac{p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{p(\boldsymbol{y})}. \tag{2.6}$$

This implies that the MAP estimate can equivalently be defined as

$$\hat{\boldsymbol{\theta}}_{\text{MAP}} = \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta}). \tag{2.7}$$

Here, $p(\boldsymbol{\theta})$ is called the prior. It models the a priori distribution of the parameter vector $\boldsymbol{\theta}$. In case of an uninformative, uniform prior, the MAP estimate and the ML estimate become equivalent. Example 2.2 shows how a MAP estimate for an exponential prior is derived. Exponential priors are for instance used in the context of *ultra-wideband* (UWB) positioning, see Section 7.3.

2.2 Example: Range measurements with outliers

Consider a set of N independent range measurements $\boldsymbol{y}_{1:N} = \{y_n\}_{n=1}^N$ with Gaussian measurement noise. Due to the measurement technique, the measurements are occasionally corrupted with a positive offset. That is,

$$y_n = r + d_n + e_n, \tag{2.8}$$

where r is the range, $e_n \sim \mathcal{N}(0, \sigma^2)$ is the measurement noise and $d_n \geq 0$ is a possibly nonzero disturbance. The presence of the disturbances is taken into account using an exponential prior with parameter λ,

$$p(d_n) = \begin{cases} \lambda \exp(-\lambda d_n), & d_n \geq 0 \\ 0, & d_n < 0. \end{cases} \tag{2.9}$$

The MAP estimate of the parameter vector $\boldsymbol{\theta} = \{r, d_1, \dots d_N\}$ is now given as

$$\begin{aligned} \hat{\boldsymbol{\theta}} &= \arg\max_{\boldsymbol{\theta}} p(\boldsymbol{y}_{1:N}|\boldsymbol{\theta}) p(\boldsymbol{\theta}) = \arg\max_{\boldsymbol{\theta}} \prod_{n=1}^N p(y_n|r, d_n) p(d_n) \\ &= \arg\min_{\boldsymbol{\theta}} \sum_{n=1}^N \frac{(y_n - r - d_n)^2}{2\sigma^2} + \lambda \sum_{n=1}^N d_n \end{aligned} \tag{2.10}$$

The last step is obtained by taking the logarithm and removing some constants. There exists no closed form solution to (2.10). However, it can be solved using numerical optimization methods, see Section 2.3.

As an application of parameter estimation consider the topic of system identification (Ljung, 1999, 2008). Here, one wants to identify or estimate parameters of dynamical systems. These parameters are computed from information about the system in the form of input and output data. This data is denoted

$$\boldsymbol{z}_{1:N} = \{\boldsymbol{u}_1, \boldsymbol{u}_2, \dots, \boldsymbol{u}_N, \boldsymbol{y}_1, \boldsymbol{y}_2, \dots, \boldsymbol{y}_N\}, \tag{2.11}$$

where $\{\boldsymbol{u}_t\}_{t=1}^N$ denote the input signals and $\{\boldsymbol{y}_t\}_{t=1}^N$ denote the output signals (measurements). Note that the input signals are sometimes sampled at a higher frequency than the measurements, for instance in the case of inertial sensors and vision. In this case, some of the measurements are missing and they are removed from the dataset.

Besides a dataset, a predictor capable of predicting measurements is required. More formally, the predictor is a parameterized mapping $g(\cdot)$ from past input and output signals to the space of the model outputs,

$$\hat{\boldsymbol{y}}_{t|t-1}(\boldsymbol{\theta}) = g(\boldsymbol{\theta}, \boldsymbol{z}_{1:t-1}), \tag{2.12}$$

where $\boldsymbol{z}_{1:t-1}$ is used to denote all the input and output signals up to time $t-1$ and $\boldsymbol{\theta}$ denotes the parameters of the predictor. Many different predictor families are described in literature (see Ljung, 1999), including black-box models and gray-box models based on physical insight.

Finally, in order to compute an estimate of the parameters θ a method to determine which parameters are best at describing the data is needed. This is accomplished by posing and solving an optimization problem

$$\hat{\theta} = \arg\min_{\theta} V(\theta, z_{1:N}), \tag{2.13}$$

where V is a cost function measuring how well the predictor agrees with the measurements. Typically, the cost function $V(\theta, z_{1:N})$ is of the form

$$V(\theta, z_{1:N}) = \frac{1}{2} \sum_{t=1}^{N} \|y_t - \hat{y}_{t|t-1}(\theta)\|^2_{\Lambda_t}, \tag{2.14}$$

where Λ_t is a suitable weighting matrix. Introducing the stacked normalized prediction vector $e = \text{vec}\,[e_1, e_2, \ldots, e_N]$ of length Nn_y where

$$e_t = \Lambda_t^{-1/2}(y_t - \hat{y}_{t|t-1}(\theta)), \tag{2.15}$$

the optimization problem (2.13) can be simplified to

$$\hat{\theta} = \arg\min_{\theta} e^T e. \tag{2.16}$$

The covariance of the obtained estimate $\hat{\theta}$ can be approximated as (Ljung, 1999)

$$\text{Cov}\,\hat{\theta} = \frac{e^T e}{N n_y} \left([\mathrm{D}_\theta\, e]^T [\mathrm{D}_\theta\, e] \right)^{-1}, \tag{2.17}$$

where $[\mathrm{D}\cdot]$ is the Jacobian matrix, see Appendix A.1. The dataset (2.11) on which the estimate $\hat{\theta}$ is based does not have to satisfy any constraint other than that it should be informative, i.e., it should allow one to distinguish between different models and/or parameter values (Ljung, 1999). It is very hard to quantify this notion of informativeness, but in an uninformative experiment the predicted output will not be sensitive to certain parameters and this results in large variances of the obtained estimates.

2.2 Sensor fusion

Sensor fusion is about combining information from different sensor sources. It has become a synonym for state estimation, which can be interpreted as a special case of parameter estimation where the parameters are the state of the system under consideration. Depending on the application, the current state or the state evolution are of interest. The former case is called the filtering problem, the latter the smoothing problem.

2.2.1 Filtering

In filtering applications the goal is to find an estimate of the current state given all available measurements (Gustafsson, 2010; Jazwinski, 1970). Mathematically speaking, one is interested in the filtering density $p(x_t|y_{1:t})$, where x_t is the state at time t and the set $y_{1:t} = \{y_1, \ldots, y_t\}$ contains the measurement history up to

time t. The filtering density is typically calculated using the recursion

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) = \frac{p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})}{p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1})}, \tag{2.18a}$$

$$p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}) = \int p(\boldsymbol{y}_t|\boldsymbol{x}_t)p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})\,\mathrm{d}\boldsymbol{x}_t, \tag{2.18b}$$

$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1}) = \int p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1})\,\mathrm{d}\boldsymbol{x}_{t-1}. \tag{2.18c}$$

This filter recursion follows from Bayes' rule (2.6) in combination with the standard assumption that the underlying model has the Markov property

$$p(\boldsymbol{x}_t|\boldsymbol{x}_{1:t-1}) = p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}), \qquad p(\boldsymbol{y}_t|\boldsymbol{x}_{1:t}) = p(\boldsymbol{y}_t|\boldsymbol{x}_t). \tag{2.19}$$

The conditional densities $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ and $p(\boldsymbol{y}_t|\boldsymbol{x}_t)$ are implicitly defined by the process model and measurement model, respectively. For the nonlinear state space model with additive noise, we have

$$\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t) + \boldsymbol{w}_t \quad \Leftrightarrow \quad p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = p_{\boldsymbol{w}_t}\Big(\boldsymbol{x}_t - f(\boldsymbol{x}_{t-1})\Big), \tag{2.20a}$$

$$\boldsymbol{y}_t = h(\boldsymbol{x}_t) + \boldsymbol{v}_t \quad \Leftrightarrow \quad p(\boldsymbol{y}_t|\boldsymbol{x}_t) = p_{\boldsymbol{v}_t}\Big(\boldsymbol{y}_t - h(\boldsymbol{x}_t)\Big). \tag{2.20b}$$

Here $p_{\boldsymbol{w}_t}$ and $p_{\boldsymbol{v}_t}$ denote the PDFs of the process noise $\boldsymbol{w}_t$ and the measurement noise $\boldsymbol{v}_t$ respectively. In Example 2.3, the well-known *Kalman filter* (KF) is derived from the filter equations (2.18).

2.3 Example: The Kalman filter (KF)

Consider the following linear state-space model with additive noise,

$$\boldsymbol{x}_{t+1} = A_t\boldsymbol{x}_t + \boldsymbol{w}_t, \tag{2.21a}$$

$$\boldsymbol{y}_t = C_t\boldsymbol{x}_t + \boldsymbol{v}_t, \tag{2.21b}$$

where $\boldsymbol{w}_t \sim \mathcal{N}(\boldsymbol{0}, \Sigma_{\mathrm{w}})$ and $\boldsymbol{v}_t \sim \mathcal{N}(\boldsymbol{0}, \Sigma_{\mathrm{v}})$ are *independently and identically distributed* (i.i.d.) Gaussian variables. In this case (2.18) can be evaluated analytically, resulting in the KF (Kalman, 1960; Kailath et al., 2000; Gustafsson, 2010).

Starting from a Gaussian estimate $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) = \mathcal{N}(\hat{\boldsymbol{x}}_{t|t}, \Sigma_{t|t})$, the process model (2.21a) gives the following joint PDF

$$p(\boldsymbol{x}_t, \boldsymbol{x}_{t+1}|\boldsymbol{y}_{1:t}) = \mathcal{N}\left(\begin{pmatrix}\hat{\boldsymbol{x}}_{t|t}\\ A_t\hat{\boldsymbol{x}}_{t|t}\end{pmatrix}, \begin{bmatrix}\Sigma_{t|t} & \Sigma_{t|t}A_t^T\\ A_t\Sigma_{t|t} & A_t\Sigma_{t|t}A_t^T + \Sigma_{\mathrm{w}}\end{bmatrix}\right). \tag{2.22}$$

Evaluating (2.18c), i.e. marginalizing (2.22) w.r.t. $\boldsymbol{x}_t$, now yields

$$p(\boldsymbol{x}_{t+1}|\boldsymbol{y}_{1:t}) = \mathcal{N}\Big(A_t\hat{\boldsymbol{x}}_{t|t}, A_t\Sigma_{t|t}A_t^T + \Sigma_{\mathrm{w}}\Big) \triangleq \mathcal{N}\Big(\hat{\boldsymbol{x}}_{t+1|t}, \Sigma_{t+1|t}\Big). \tag{2.23}$$

Similarly, the measurement model (2.21b) gives the following joint PDF

$$p(\boldsymbol{x}_t, \boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}) = \mathcal{N}\left(\begin{pmatrix}\hat{\boldsymbol{x}}_{t|t-1}\\ C_t\hat{\boldsymbol{x}}_{t|t-1}\end{pmatrix}, \begin{bmatrix}\Sigma_{t|t-1} & \Sigma_{t|t-1}C_t^T\\ C_t\Sigma_{t|t-1} & C_t\Sigma_{t|t-1}C_t^T + \Sigma_{\mathrm{v}}\end{bmatrix}\right). \tag{2.24}$$

Evaluating (2.18b), i.e. marginalizing (2.24) w.r.t. $\boldsymbol{x}_t$, now yields

$$p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1}) = \mathcal{N}\Big(C_t\hat{\boldsymbol{x}}_{t|t-1}, S_t\Big), \tag{2.25}$$

with $S_t = C_t\Sigma_{t|t-1}C_t^T + \Sigma_{\mathrm{v}}$. Evaluating (2.18a), i.e. conditioning (2.24) on $\boldsymbol{y}_t$, gives

$$\begin{aligned} p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) &= \mathcal{N}\Big(\hat{\boldsymbol{x}}_{t|t-1} + K_t(\boldsymbol{y}_t - C_t\hat{\boldsymbol{x}}_{t|t-1}), \Sigma_{t|t-1} - K_tS_tK_t^T\Big) \\ &\triangleq \mathcal{N}\Big(\hat{\boldsymbol{x}}_{t|t}, \Sigma_{t|t}\Big), \end{aligned} \tag{2.26}$$

where $K_t = \Sigma_{t|t-1}C_t^TS_t^{-1}$.

Alternatively, the measurement update (2.26) can be interpreted as fusing two information sources: the prediction and the measurement. More formally, we have the equations

$$\begin{pmatrix}\hat{\boldsymbol{x}}_{t|t-1}\\ \boldsymbol{y}_t\end{pmatrix} = \begin{pmatrix}I\\ C_t\end{pmatrix}\boldsymbol{x}_t + \boldsymbol{e}_t, \qquad \boldsymbol{e}_t \sim \mathcal{N}\left(\begin{pmatrix}\boldsymbol{0}\\ \boldsymbol{0}\end{pmatrix}, \begin{bmatrix}\Sigma_{t|t-1} & 0\\ 0 & \Sigma_{\mathrm{v}}\end{bmatrix}\right), \tag{2.27}$$

from which we want to estimate $\boldsymbol{x}_t$. ML estimation boils down to solving a weighted least squares problem. The mean $\hat{\boldsymbol{x}}_{t|t}$ and covariance $\Sigma_{t|t}$ obtained this way can be shown to be identical to (2.26) (Humpherys and West, 2010).

To prevent the KF from being affected by faulty measurements or outliers, outlier rejection or gating is used to discard these. The standard approach is to apply hypothesis testing on the residuals, which are defined as

$$\boldsymbol{\varepsilon}_t = \boldsymbol{y}_t - \hat{\boldsymbol{y}}_{t|t-1}, \tag{2.28}$$

i.e. the difference between the observed measurement $\boldsymbol{y}_t$ and the one-step ahead prediction from the model $\hat{\boldsymbol{y}}_{t|t-1} = C_t\hat{\boldsymbol{x}}_{t|t-1}$. In absence of errors, the residuals are normal distributed as $\boldsymbol{\varepsilon}_t \sim \mathcal{N}(0, S_t)$ according to (2.25). This allows the calculation of confidence intervals for the individual predicted measurements and in case a measurement violates these, the measurement is considered an outlier and is ignored.

Although conceptually simple, the multidimensional integrals in (2.18) typically do not have an analytical solution. An important exception is the KF, discussed in Example 2.3. However, typically some kind of approximation is used resulting in a large family of filter types, including the *extended Kalman filter* (EKF) (Schmidt, 1966; Kailath et al., 2000; Gustafsson, 2010) and *particle filter* (PF) (Gordon et al., 1993; Arulampalam et al., 2002; Gustafsson, 2010).

2.2.2 Smoothing

In smoothing problems, the goal is to find the best estimate of the state trajectory given all the measurements (Jazwinski, 1970; Gustafsson, 2010). Mathematically speaking, the posterior density $p(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t})$ is the object of interest, where $\boldsymbol{x}_{1:t} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t\}$ is a complete state trajectory up to time t and the set $\boldsymbol{y}_{1:t} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_t\}$ contains the measurements up to time t. To obtain a point estimate from the posterior, the MAP estimate is the natural choice.

Using the Markov property of the underlying model (2.19), it follows that the MAP estimate can be expanded as

$$\begin{aligned}\hat{\boldsymbol{x}}_{1:t} &= \arg\max_{\boldsymbol{x}_{1:t}} p(\boldsymbol{x}_{1:t}, \boldsymbol{y}_{1:t}) = \arg\max_{\boldsymbol{x}_{1:t}} p(\boldsymbol{x}_1) \prod_{i=2}^{t} p(\boldsymbol{x}_i|\boldsymbol{x}_{i-1}) \prod_{i=1}^{t} p(\boldsymbol{y}_i|\boldsymbol{x}_i) \\ &= \arg\min_{\boldsymbol{x}_{1:t}} -\log p(\boldsymbol{x}_1) - \sum_{i=2}^{t} \log p(\boldsymbol{x}_i|\boldsymbol{x}_{i-1}) - \sum_{i=1}^{t} \log p(\boldsymbol{y}_i|\boldsymbol{x}_i). \end{aligned} \tag{2.29}$$

Including (2.20) the MAP estimate can be formulated as the optimization problem

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & -\log p_{\boldsymbol{x}_1}(\boldsymbol{x}_1) - \sum_{i=1}^{t-1} \log p_{\boldsymbol{w}}(\boldsymbol{w}_i) - \sum_{i=1}^{t} \log p_{\boldsymbol{v}}(\boldsymbol{v}_i) \\ \text{s.t.} \quad & \boldsymbol{x}_{i+1} = f(\boldsymbol{x}_i) + \boldsymbol{w}_i, \quad i = 1, \ldots, t-1 \\ & \boldsymbol{y}_i = h(\boldsymbol{x}_i) + \boldsymbol{v}_i, \quad i = 1, \ldots, t \end{aligned} \tag{2.30}$$

with variables $\boldsymbol{\theta} = \{\boldsymbol{x}_{1:t}, \boldsymbol{w}_{1:t-1}, \boldsymbol{v}_{1:t}\}$.

2.4 Example: Gaussian smoothing

Assuming $\boldsymbol{x}_1 \sim \mathcal{N}(\bar{\boldsymbol{x}}_1, \Sigma_{\mathrm{x}_1})$, $\boldsymbol{w}_i \sim \mathcal{N}(\boldsymbol{0}, \Sigma_{\mathrm{w}})$ and $\boldsymbol{v}_i \sim \mathcal{N}(\boldsymbol{0}, \Sigma_{\mathrm{w}})$, with known mean and covariance, the optimization problem (2.30) simplifies using (2.2) to

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \frac{1}{2}\|\tilde{\boldsymbol{e}}\|_2^2 + \frac{1}{2}\sum_{i=1}^{t-1} \|\tilde{\boldsymbol{w}}_i\|_2^2 + \frac{1}{2}\sum_{i=1}^{t} \|\tilde{\boldsymbol{v}}_i\|_2^2 \\ \text{s.t.} \quad & \tilde{\boldsymbol{e}} = \Sigma_{\mathrm{x}_1}^{-1/2}(\boldsymbol{x}_1 - \bar{\boldsymbol{x}}_1) \\ & \tilde{\boldsymbol{w}}_i = \Sigma_{\mathrm{w}}^{-1/2}(\boldsymbol{x}_{i+1} - f(\boldsymbol{x}_i)), \quad i = 1, \ldots, t-1 \\ & \tilde{\boldsymbol{v}}_i = \Sigma_{\mathrm{v}}^{-1/2}(\boldsymbol{y}_i - h(\boldsymbol{x}_i)), \quad i = 1, \ldots, t \end{aligned} \tag{2.31}$$

with variables $\boldsymbol{\theta} = \{\boldsymbol{x}_{1:t}, \tilde{\boldsymbol{e}}, \tilde{\boldsymbol{w}}_{1:t-1}, \tilde{\boldsymbol{v}}_{1:t}\}$. The constant terms present in (2.2b) do not affect the position of the optimum of (2.31) and have been left out. Introducing the normalized residual vector $\boldsymbol{\epsilon} = \{\tilde{\boldsymbol{e}}, \tilde{\boldsymbol{w}}_{1:t-1}, \tilde{\boldsymbol{v}}_{1:t}\}$, (2.31) becomes the nonlinear least squares problem

$$\min_{\boldsymbol{\theta}} \quad \tfrac{1}{2}\|\boldsymbol{\epsilon}(\boldsymbol{\theta})\|_2^2 \tag{2.32}$$

with variables $\boldsymbol{\theta} = \boldsymbol{x}_{1:t}$.

Since MAP estimation is used for the smoothing problem, the problem can be straightforwardly extended to include other parameters of interest as well. Examples include lever arms, sensor alignment, as well as standard deviations. That is, MAP estimation can be used to solve combined smoothing and calibration problems.

The filter density $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$ can be obtained from the posterior density $p(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t})$

by marginalization,

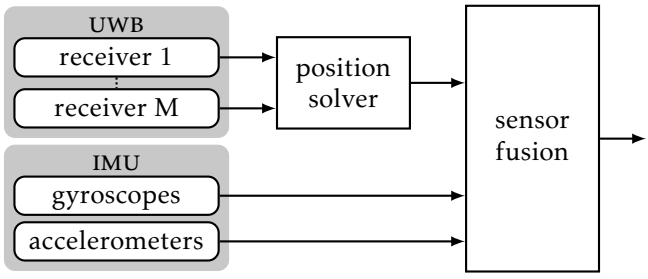
$$p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}) = \int p(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t})\,\mathrm{d}\boldsymbol{x}_{1:t-1}, \tag{2.33}$$

i.e., integrating away the variables that are not of interest, in this case the state history $\boldsymbol{x}_{1:t-1}$. This implies that the filter estimates are contained in the MAP estimate of the smoothing problem. Although the smoothing problem is inherently formulated as a batch problem, recursive versions are possible by adding new measurements to the problem and solving the modified problem starting it in the most recent estimate. Recent advancements in *simultaneous localization and mapping* (SLAM) literature (Grisetti et al., 2010; Kaess et al., 2008) show that such an approach is feasible in real-time.

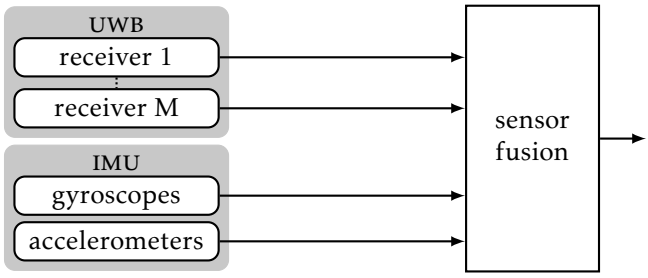
2.2.3 Loose and tight coupling

In the navigation community it is not uncommon to work with so-called loosely coupled systems. By loosely coupled approach we refer to a solution where the measurements from one or several of the individual sensors are pre-processed before they are used to compute the final result. A tightly coupled approach on the other hand refers to an approach where all the measurements are used directly to compute the final result.

As an example consider the sensor fusion of UWB and inertial sensors, see Figure 2.1. The loosely coupled approach processes the UWB measurements with a



(a) *Loosely coupled*



(b) *Tightly coupled*

Figure 2.1: *Two approaches to sensor fusion of UWB and inertial sensors.*

position solver. The positions obtained this way are used as measurements in the sensor fusion algorithm. In a tightly coupled setup the 'raw' UWB measurements are directly used in the sensor fusion algorithm.

The pre-processing step of loose coupling typically can be interpreted as some kind of estimator, which estimates a parameter from measurements. For instance, an ML estimator is typically used as the position solver in Figure 2.1a. The estimate is subsequently used as an artificial measurement by the sensor fusion algorithm. As long as the correct PDF of the estimate is used, the loose and the tight coupled approaches become equivalent. However, this PDF is in practice not available or approximated, resulting in a loss of information. Furthermore, the tightly coupled approach offers additional outlier rejection possibilities, resulting in more accurate and robust results. From a practical point of view loose coupling might be advantageous in some cases. However, the choice for doing so has to be made consciously and the intermediate estimates should be accompanied with realistic covariance values.

2.3 Optimization

In Section 2.1 and Section 2.2 a number of optimization problems have been formulated. This section aims at providing some background on how to solve these problems.

The general form of an optimization problem is

$$\min_{\theta} \quad f(\theta) \tag{2.34a}$$

$$\text{s.t.} \quad c_i(\theta) \leq \mathbf{0}, \tag{2.34b}$$

$$c_e(\theta) = \mathbf{0}, \tag{2.34c}$$

where f is the objective function, c_i are inequality constraints, c_e are equality constraints and θ are the variables. Standard algorithms for solving (2.34) exist in optimization literature (Boyd and Vandenberghe, 2004; Nocedal and Wright, 2006), including (infeasible start) Newton methods and trust-region methods.

For the optimization problem (2.34), the first order necessary optimality conditions, often known as the *Karush-Kuhn-Tucker* (KKT) conditions, are given as (Boyd and Vandenberghe, 2004; Nocedal and Wright, 2006)

$$\begin{aligned}
[\mathrm{D}_\theta f](\theta_*) + \lambda_*^T [\mathrm{D}_\theta c_i](\theta_*) + \nu_*^T [\mathrm{D}_\theta c_e](\theta_*) &= 0, \\
c_i(\theta_*) &\leq \mathbf{0}, \\
c_e(\theta_*) &= \mathbf{0}, \\
\lambda_* &\geq \mathbf{0}, \\
\operatorname{diag}(\lambda_*) c_i(\theta_*) &= \mathbf{0},
\end{aligned} \tag{2.35}$$

where $[\mathrm{D}\,\cdot\,]$ is the Jacobian matrix, see Appendix A.1, λ and ν are the Lagrange dual variables associated with the inequality constraints (2.34b) and with the equality constraints (2.34c), respectively, and $(\theta_*, \lambda_*, \nu_*)$ is a candidate optimal

point. Most optimization methods can be interpreted using these conditions; given an initial guess, a local approximation is made to the KKT conditions (2.35). Solving the resulting equations determines a search direction which is used to find a new and improved solution. The process is then repeated until convergence is obtained. Examples 2.5 and 2.6 provide examples relevant for this thesis.

2.5 Example: Equality constrained least squares

Consider the following equality constrained least squares problem

$$\min_{\boldsymbol{\theta}} \quad \tfrac{1}{2}\|\boldsymbol{\epsilon}(\boldsymbol{\theta})\|_2^2 \tag{2.36a}$$

$$\text{s.t.} \quad A\boldsymbol{\theta} + \boldsymbol{b} = \mathbf{0} \tag{2.36b}$$

For this problem, the first order linearization of the KKT conditions (2.35) results in the following system of equation

$$\underbrace{\begin{bmatrix} J^T J & A^T \\ A & 0 \end{bmatrix}}_{\triangleq K} \begin{pmatrix} \Delta\boldsymbol{\theta} \\ \boldsymbol{\nu} \end{pmatrix} = -\begin{pmatrix} J^T \boldsymbol{\epsilon} \\ A\boldsymbol{\theta} + \boldsymbol{b} \end{pmatrix}, \tag{2.37}$$

where $J = \mathrm{D}_{\boldsymbol{\theta}}\, \boldsymbol{\epsilon}$ is the Jacobian of the residual vector $\boldsymbol{\epsilon}$ w.r.t. the parameter vector $\boldsymbol{\theta}$, and $\boldsymbol{\nu}$ is the Lagrange dual variable associated with the constraint (2.36b). Solving for $(\Delta\boldsymbol{\theta}, \boldsymbol{\nu})$ yields a primal-dual search direction, which can be used in combination with an appropriately chosen step-size s to update the solution as $\boldsymbol{\theta} := \boldsymbol{\theta} + s\Delta\boldsymbol{\theta}$.

At the optimum $\boldsymbol{\theta}^*$, (2.37) can be used to obtain the Jacobian

$$[\mathrm{D}_{\boldsymbol{\epsilon}}\, \boldsymbol{\theta}] = [\mathrm{D}_{\boldsymbol{\epsilon}}\, \Delta\boldsymbol{\theta}] = -(K^{-1})_{11} J^T. \tag{2.38}$$

According the stochastic interpretation of (2.36), $\boldsymbol{\epsilon}$ are normalized residuals with a Gaussian distribution and $\mathrm{Cov}(\boldsymbol{\epsilon}) = I$. Therefore, application of Gauss' approximation formula (Ljung, 1999) yields

$$\begin{aligned} \mathrm{Cov}(\boldsymbol{\theta}) &= [\mathrm{D}_{\boldsymbol{\epsilon}}\, \boldsymbol{\theta}]\, \mathrm{Cov}(\boldsymbol{\epsilon}) [\mathrm{D}_{\boldsymbol{\epsilon}}\, \boldsymbol{\theta}]^T \\ &= (K^{-1})_{11} J^T J (K^{-1})_{11} = (K^{-1})_{11}. \end{aligned} \tag{2.39}$$

The last equality can be shown by expanding the $(1,1)$–block of K^{-1} using (A.11),

$$(K^{-1})_{11} = [I - X]\,(J^T J)^{-1}, \tag{2.40a}$$

$$X = (J^T J)^{-1} A^T \left(A (J^T J)^{-1} A^T\right)^{-1} A. \tag{2.40b}$$

Note that (2.40) shows that the covariance of the constrained problem is closely related to the covariance of the unconstrained problem $(J^T J)^{-1}$, see also Example 2.6.

2.6 Example: Constrained least squares covariance

To illustrate the covariance calculation (2.39), consider the example problem

$$\begin{aligned} \min_{\boldsymbol{\theta}} \quad & \frac{1}{2}\sum_{k=1}^{3}(\theta_k - y_k)^2 \\ \text{s.t.} \quad & \theta_1 - \theta_2 = 0 \end{aligned}$$

for which the KKT matrix K and its inverse are given by

$$K = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ \hline 1 & -1 & 0 & 0 \end{array}\right], \qquad K^{-1} = \left[\begin{array}{ccc|c} .5 & .5 & 0 & .5 \\ .5 & .5 & 0 & -.5 \\ 0 & 0 & 1 & 0 \\ \hline .5 & -.5 & 0 & -.5 \end{array}\right].$$

The covariance of the solution is now given by the top-left 3×3 block of K^{-1}. It is indeed correct, as can be verified by elimination of θ_1 or θ_2 and calculating the covariance of the corresponding unconstrained problem. Furthermore, note that the presence of the constraint results in a reduced variance compared to the unconstrained covariance $(J^T J)^{-1} = I_3$.

For an equality constrained problem, the KKT conditions (2.35) can be more compactly written as

$$[\mathrm{D}_{(\boldsymbol{\theta},\boldsymbol{\nu})}\,\mathcal{L}](\boldsymbol{\theta}_*, \boldsymbol{\nu}_*) = \mathbf{0}, \tag{2.41}$$

where the Lagrangian $\mathcal{L}$ is defined as

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\nu}) = f(\boldsymbol{\theta}) + \boldsymbol{\nu}^T c_e(\boldsymbol{\theta}). \tag{2.42}$$

Note that this interpretation can also be used for inequality constrained problems, once the set of active constraints has been determined. Around an optimal point $(\boldsymbol{\theta}_*, \boldsymbol{\nu}_*)$ the Lagrangian $\mathcal{L}$ can be approximated as

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\nu}) \approx \mathcal{L}(\boldsymbol{\theta}_*, \boldsymbol{\nu}_*) + \frac{1}{2}\begin{pmatrix} \boldsymbol{\theta} - \boldsymbol{\theta}_* \\ \boldsymbol{\nu} - \boldsymbol{\nu}_* \end{pmatrix}^T \underbrace{\begin{bmatrix} [\mathrm{D}_{\boldsymbol{\theta}\boldsymbol{\theta}}\, f](\boldsymbol{\theta}_*) & [\mathrm{D}_{\boldsymbol{\theta}}\, c_e]^T(\boldsymbol{\theta}_*) \\ [\mathrm{D}_{\boldsymbol{\theta}}\, c_e](\boldsymbol{\theta}_*) & 0 \end{bmatrix}}_{\triangleq K} \begin{pmatrix} \boldsymbol{\theta} - \boldsymbol{\theta}_* \\ \boldsymbol{\nu} - \boldsymbol{\nu}_* \end{pmatrix}, \tag{2.43}$$

where the linear term is not present because of (2.41). This result can be used to obtain the Laplace approximation (Bishop, 2006) of the PDF of $(\boldsymbol{\theta}, \boldsymbol{\nu})$. This method determines a Gaussian approximation of distribution $p(\boldsymbol{y}) = f(\boldsymbol{y})/Z$ centered on the mode $\boldsymbol{y}_0$. The normalization constant Z is typically unknown. Taking the Taylor expansion of $-\log p(\boldsymbol{y})$ around $\boldsymbol{y}_0$ gives

$$\begin{aligned} -\log p(\boldsymbol{y}) &\approx -\log f(\boldsymbol{y}_0) + \log Z + \tfrac{1}{2}(\boldsymbol{y} - \boldsymbol{y}_0)^T [\mathrm{D}_{\boldsymbol{y}\boldsymbol{y}} \log f(\boldsymbol{y})](\boldsymbol{y} - \boldsymbol{y}_0) \\ &= \log a + \tfrac{1}{2}(\boldsymbol{y} - \boldsymbol{y}_0)^T A(\boldsymbol{y} - \boldsymbol{y}_0). \end{aligned} \tag{2.44}$$

The first order term in the expansion vanishes since $\boldsymbol{y}_0$ is a stationary point of $p(\boldsymbol{y})$ and therefore $[\mathrm{D}_{\boldsymbol{y}} \log f(\boldsymbol{y})](\boldsymbol{y}_0) = \mathbf{0}$. Furthermore, since f is unnormalized, a

cannot be determined from $f(\boldsymbol{y})$. Taking the exponential of both sides, we obtain

$$\begin{aligned} p(\boldsymbol{y}) &\approx a^{-1} \exp\left(-\tfrac{1}{2}(\boldsymbol{y}-\boldsymbol{y}_0)^T A(\boldsymbol{y}-\boldsymbol{y}_0)\right) \\ &= (2\pi)^{-\frac{n}{2}} (\det A)^{\frac{1}{2}} \exp\left(-\tfrac{1}{2}(\boldsymbol{y}-\boldsymbol{y}_0)^T A(\boldsymbol{y}-\boldsymbol{y}_0)\right) = \mathcal{N}(\boldsymbol{y}_0, A^{-1}). \end{aligned} \tag{2.45}$$

In the last step the normalization constant a is determined by inspection. Using (2.43), the Laplace approximation of the PDF of $(\boldsymbol{\theta}, \boldsymbol{\nu})$ is given by

$$p(\boldsymbol{\theta}, \boldsymbol{\nu}) \approx \mathcal{N}\left(\begin{pmatrix} \boldsymbol{\theta}_* \\ \boldsymbol{\nu}_* \end{pmatrix}, K^{-1}\right). \tag{2.46}$$

Marginalizing this result shows that the PDF of $\boldsymbol{\theta}$ can be approximated as

$$p(\boldsymbol{\theta}) \approx \mathcal{N}\left(\boldsymbol{\theta}_*, (K^{-1})_{11}\right). \tag{2.47}$$

This result coincides with (2.39). As the size of the problem increases, the inversion of K can become infeasible. However, given a sparse matrix factorization of K it is possible to efficiently calculate parts of its inverse, see Appendix A.4.

The linearized KKT system for the smoothing problem (2.32) is very sparse and contains a lot of structure. By exploiting this structure with for instance sparse matrix factorizations, see Appendix A.3, efficient implementations are obtained. The work of Grisetti et al. (2010) is also of interest in this context. They exploit the predominantly time connectivity of the smoothing problem and achieve real-time implementations using concepts of subgraphs and hierarchies.

3
Sensors

In this thesis, inertial sensors are used in combination with aiding sensors such as vision, ultra-wideband (UWB) or global positioning system (GPS). This chapter aims to discuss the measurements, calibration techniques and standard usage for each kind of sensor.

3.1 Inertial measurement unit

Inertial measurement units (IMUs) are devices containing *three dimensional* (3D) rate gyroscopes and 3D accelerometers. In many cases, 3D magnetometers are included as well. IMUs are typically used for navigation purposes where the position and the orientation of a device is of interest. Nowadays, many gyroscopes and accelerometers are based on *micro-machined electromechanical system* (MEMS) technology, see Figure 3.1 and Figure 3.2.

Compared to traditional technology, MEMS components are small, light, inexpensive, have low power consumption and short start-up times. Currently, their major disadvantage is the reduced performance in terms of accuracy and bias stability. This is the main cause for the drift in standalone MEMS inertial navigation systems (Woodman, 2007). An overview of the available inertial sensor grades is shown in Figure 3.3.

The functionality of the MEMS sensors is based upon simple mechanical principles. Angular velocity can be measured by exploiting the Coriolis effect of a vibrating structure; when a vibrating structure is rotated, a secondary vibration is induced from which the angular velocity can be calculated. Acceleration can be measured with a spring suspended mass; when subjected to acceleration the mass will be displaced. Using MEMS technology the necessary mechanical structures

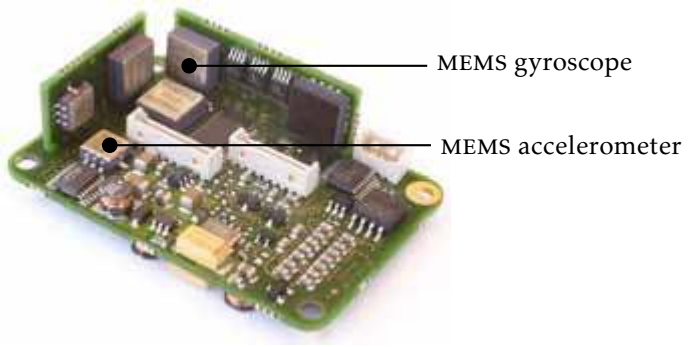


Figure 3.1: *A circuit board of an IMU containing MEMS components.*

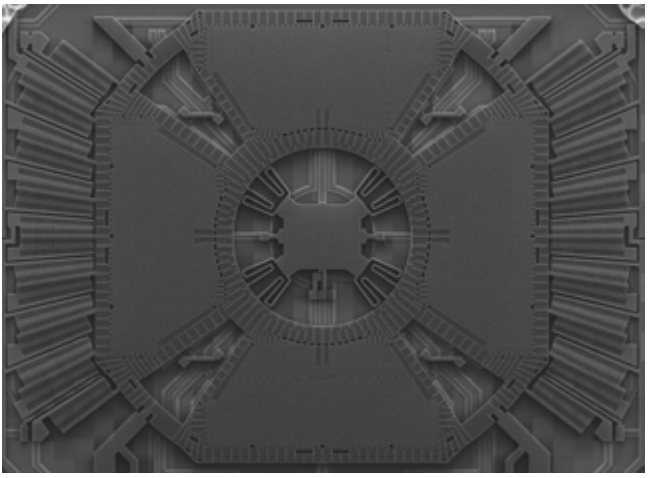


Figure 3.2: *Mechanical structure of a 3D MEMS gyroscope. By courtesy of STMicroelectronics.*

can be manufactured on silicon chips in combination with capacitive displacement pickups and electronic circuitry (Analog Devices, 2010; STMicroelectronics, 2011).

3.1.1 Measurement model

The sensing components have one or more sensitive axes along which a physical quantity (e.g. specific force, angular velocity or magnetic field) is converted into an output voltage. A typical sensor shows almost linear behavior in the working area. Based on this observation, the following (simplified) relation between the output voltage $\boldsymbol{u}$ and the physical signal $\boldsymbol{y}$ is postulated for multiple sensors with their sensitive axis aligned in a suitable configuration,

$$\boldsymbol{u}_t = GR\boldsymbol{y}_t + \boldsymbol{c}. \tag{3.1}$$

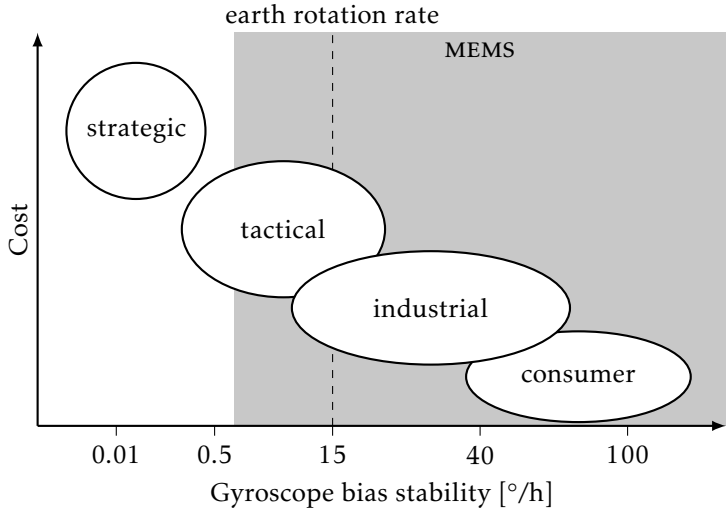


Figure 3.3: *Inertial sensors grades.*

Here the time dependency of the signals $\boldsymbol{y}$ and $\boldsymbol{u}$ is denoted using the subscript t, G is a diagonal matrix containing the gain of every sensitive axis, R is the alignment matrix specifying the direction of the sensitive axis *with respect to* (w.r.t.) the sensor housing and $\boldsymbol{c}$ is the offset vector. Note that the gain and the offset are typically temperature dependent. The calibrated measurement signal is obtained from the measured voltages by inverting (3.1).

The sensing components in an IMU are rate gyroscopes, accelerometers and magnetometers. The gyroscopes measure angular velocity or rate-of-turn $\boldsymbol{\omega}$. The accelerometers do not measure accelerations directly, but rather the external specific force $\boldsymbol{f}$. Linear acceleration $\ddot{\boldsymbol{b}}$ as well as the earth's gravitational field $\boldsymbol{g}$ contribute to the specific force. The magnetometers measure the local magnetic field $\boldsymbol{m}$. Both gyroscopes and accelerometers suffer from slowly time-varying biases.

In order to discuss the measurement model in more detail, a number of coordinate frames needs to be introduced:

The body frame (b-frame) is the coordinate frame of the moving IMU. Its origin is located in the center of the accelerometer triad and it is aligned to the casing. All the inertial measurements are resolved in this frame.

The navigation frame (n-frame) is a local geographic frame in which we want to navigate. That is, we want to know the position and orientation of the b-frame w.r.t. this frame. For most applications it is defined stationary w.r.t. the earth. However, in the context of navigation over large distances it is customary to move and rotate the n-frame along the surface of the earth. The first definition is used throughout this thesis, unless mentioned explicitly.

The inertial frame (i-frame) is a stationary, non-rotating frame. The IMU measures linear acceleration and angular velocity w.r.t. this frame. Its origin is located at the center of the earth and its axis are aligned w.r.t. the stars.

The earth frame (e-frame) coincides with the i-frame, but rotates with the earth. That is, it has its origin at the center of the earth and axes which are fixed w.r.t. the earth.

These coordinate frames are illustrated in Figure 3.4.

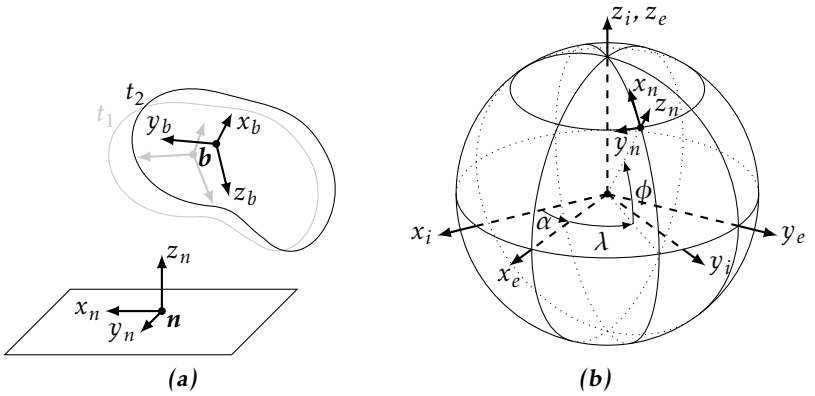


Figure 3.4: *An illustration of the coordinate frames: (a) shows the b-frame at time instants t_1 and t_2 moving w.r.t. the n-frame. (b) shows the n-frame at latitude ϕ and longitude λ, the e-frame at angle $\alpha(t) = \omega_{ie}t$ and the i-frame.*

Using the notation of page xv and suppressing the time dependency of the quantities involved, the gyroscope measurements $\boldsymbol{y}_\omega$ are modeled as (Titterton and Weston, 1997)

$$\boldsymbol{y}_\omega = \boldsymbol{\omega}_{ib}^b + \boldsymbol{\delta}_\omega^b + \boldsymbol{e}_\omega^b. \tag{3.2a}$$

Here, $\boldsymbol{\omega}_{ib}$ is the angular velocity of the b-frame as observed in the i-frame, $\boldsymbol{\delta}_\omega$ is a slowly time-varying bias and $\boldsymbol{e}_\omega$ is *independently and identically distributed* (i.i.d.) Gaussian noise. The angular velocity can be expanded as

$$\boldsymbol{\omega}_{ib}^b = R^{bn}(\boldsymbol{\omega}_{ie}^n + \boldsymbol{\omega}_{en}^n) + \boldsymbol{\omega}_{nb}^b, \tag{3.2b}$$

where R is a rotation matrix, $\boldsymbol{\omega}_{ie}$ is the earth rate, $\boldsymbol{\omega}_{en}$ is the transport rate and $\boldsymbol{\omega}_{nb}$ is the angular velocity required for navigation purposes.

The accelerometer measurements $\boldsymbol{y}_\mathrm{a}$ are modeled as (Titterton and Weston, 1997)

$$\boldsymbol{y}_\mathrm{a} = \boldsymbol{f}^b + \boldsymbol{\delta}_\mathrm{a}^b + \boldsymbol{e}_\mathrm{a}^b = R^{bn}(\ddot{\boldsymbol{b}}_{ii}^n - \boldsymbol{g}^n) + \boldsymbol{\delta}_\mathrm{a}^b + \boldsymbol{e}_\mathrm{a}^b, \tag{3.3a}$$

where $\boldsymbol{f}$ is the external specific force, $\boldsymbol{\delta}_\mathrm{a}$ is a slowly time-varying bias and $\boldsymbol{e}_\mathrm{a}$ is i.i.d. Gaussian noise. The second expression in (3.3a) splits the specific force into its contributions from the linear acceleration of the body as observed in the i-

frame $\ddot{\boldsymbol{b}}_{ii}$ and the gravity vector $\boldsymbol{g}$. The linear acceleration can be expanded as (see Section 4.3 and (4.28)),

$$\ddot{\boldsymbol{b}}_{ii}^{n} = \boldsymbol{\omega}_{ie}^{n} \times \boldsymbol{\omega}_{ie}^{n} \times R^{ni}\boldsymbol{b}^{i} + 2\boldsymbol{\omega}_{ie}^{n} \times \dot{\boldsymbol{b}}_{n}^{n} + \ddot{\boldsymbol{b}}_{nn}^{n}, \tag{3.3b}$$

where $\ddot{\boldsymbol{b}}_{nn}$ is the acceleration of the body as observed in the n-frame required for navigation purposes. In the context of navigation over large distances, it is more common to work with the ground speed w.r.t. the earth $\dot{\boldsymbol{b}}_{e}$ instead of $\dot{\boldsymbol{b}}_{n}$. In this case one is interested in $\ddot{\boldsymbol{b}}_{en}$ which gives rise to the following alternative expansion of the linear acceleration (see (4.31)),

$$\ddot{\boldsymbol{b}}_{ii}^{n} = \boldsymbol{\omega}_{ie}^{n} \times \boldsymbol{\omega}_{ie}^{n} \times R^{ni}\boldsymbol{b}^{i} + (2\boldsymbol{\omega}_{ie}^{n} + \boldsymbol{\omega}_{en}^{n}) \times \dot{\boldsymbol{b}}_{e}^{n} + \ddot{\boldsymbol{b}}_{en}^{n}. \tag{3.3c}$$

The centripetal acceleration terms $\boldsymbol{\omega}_{ie}^{n} \times \boldsymbol{\omega}_{ie}^{n} \times R^{ni}\boldsymbol{b}^{i}$ in (3.3b) and (3.3c) are typically absorbed in the (local) gravity vector.

The magnetometer measurements $\boldsymbol{y}_{\mathrm{m}}$ are modeled as

$$\boldsymbol{y}_{\mathrm{m}} = \boldsymbol{m}^{b} + \boldsymbol{e}_{\mathrm{m}}^{b} = R^{bn}\boldsymbol{m}^{n} + \boldsymbol{e}_{\mathrm{m}}^{b}. \tag{3.4}$$

Here, $\boldsymbol{m}$ is the local magnetic field vector and $\boldsymbol{e}_{\mathrm{m}}$ is i.i.d. Gaussian noise. In absence of ferromagnetic objects $\boldsymbol{m}$ is the earth magnetic field and the magnetometer measurements can be used as a compass to give the direction of the magnetic north.

3.1.2 Calibration

Calibrating the IMU boils down to finding the gain G, the alignment R and the offset $\boldsymbol{c}$ in (3.1) for the accelerometer and the gyroscope. The calibration principle is to subject the IMU to a known acceleration or angular velocity and choose the calibration parameters such that the observed sensor output is as likely as possible. Ignoring the time variability of the biases and using the standard assumptions of i.i.d. Gaussian noise, the corresponding *maximum likelihood* (ML) problem can be formulated as

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \sum_{t} \tfrac{1}{2}\|\boldsymbol{u}_t - h(\boldsymbol{y}_t, \boldsymbol{\theta})\|^2_{\Sigma_{\mathrm{y}}^{-1}}, \tag{3.5}$$

where the parameter vector $\boldsymbol{\theta} = \{G, R, \boldsymbol{c}\}$ consists of the parameters in (3.1) and Σ_{y} is the covariance of $\boldsymbol{y}$. Traditionally, known excitations are obtained using special manipulators such as turntables (Titterton and Weston, 1997). Alternatively, the IMU can be placed in several static orientations (Ferraris et al., 1995; Skog, 2009).

Commercially available IMUs are typically calibrated at production. Besides gain, alignment and offset, temperature dependency and g-sensitivity are sometimes also accounted for (Grewal and Andrews, 2011). The latter term describes the effect of the specific force on the gyroscope output. Recalibration is not necessary for the gyroscopes and the accelerometers unless the housing is opened or the sensor is subjected to a large shock. When the IMU is mounted on a ferromagnetic object, the magnetometers do have to be recalibrated as discussed in Section 5.4.

3.1.3 Strapdown navigation

The measurements from the accelerometers and the gyroscopes can be used to compute the position and orientation of an object relative to a known starting point using inertial navigation (Chatfield, 1997; Titterton and Weston, 1997). In a strapdown configuration such as the sensor unit, the measurements are resolved in the b-frame, rather than in the i-frame. Hence, the orientation q^{nb}, see Section 4.2, can be calculated by integrating the angular velocity $\boldsymbol{\omega}_{nb}^{b}$. The position $\boldsymbol{b}^{n}$ can be obtained by double integration of the linear acceleration $\ddot{\boldsymbol{b}}^{n}$. This acceleration is obtained from the external specific force $\boldsymbol{f}^{b}$, after it has been rotated to the n-frame and corrected for gravity and Coriolis effects. This procedure is illustrated in Figure 3.5. For good integration results, the initial conditions have to be determined accurately. This is a challenging problem by itself. In practice, the

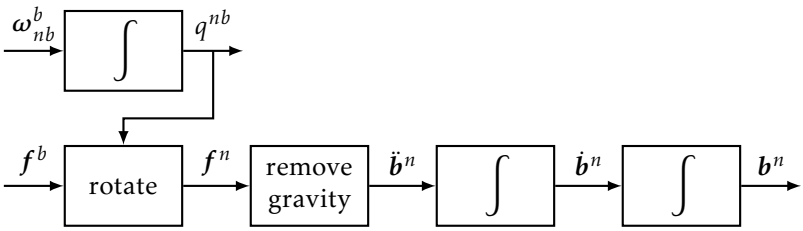


Figure 3.5: *Strapdown inertial navigation algorithm for calculating position and orientation from angular velocity and specific force.*

angular velocity $\boldsymbol{\omega}_{nb}^{b}$ and the external specific force $\boldsymbol{f}^{b}$ are obtained from the gyroscope and the accelerometer measurements. These include bias and noise terms which cause errors in the calculated position and orientation. This so-called integration drift is inherent to all inertial navigation. Moreover, using MEMS inertial sensors, the integration drift is relatively large. Hence, the orientation estimates and especially the position estimates are only accurate and reliable for a short period of time.

3.1.4 Allan variance

The performance of IMUs is typically specified in terms of their Allan variance (IEEE Std 1559, 2009; El-Sheimy et al., 2008). This is a time-domain analysis method of the stochastic process of the sensor measurements which investigates the errors as a function of averaging times.

Assume that $y_{1:K} = \{y_k\}_{k=1}^{K}$ is a dataset of K consecutive measurements recorded with sample time T. The measurement is performed in a stable climate without exciting the system. Averaging over clusters of n samples, with a cluster time $T_c = nT$, we obtain $\bar{y}_{1:L}(n) = \{\bar{y}_l\}_{l=1}^{L}$, where $L = \lfloor \frac{K}{n} \rfloor$ and the cluster average $\bar{y}_l$ is

defined as

$$\bar{y}_l = \frac{1}{n} \sum_{k=1}^{n} y_{(l-1)n+k}. \tag{3.6}$$

The Allan variance for cluster time $T_c = nT$ is now defined as

$$\sigma^2(T_c) = \frac{1}{2(L-1)} \sum_{l=1}^{L-1} (\bar{y}_{l+1} - \bar{y}_l)^2. \tag{3.7}$$

That is, it calculates the mean square cluster average error.

The Allan variance can be related to the power spectral density S of the measurements y using (El-Sheimy et al., 2008)

$$\sigma^2(T_c) = 4 \int_0^\infty \frac{\sin^4(\pi f T_c)}{(\pi f T_c)^2} S(f)\, \mathrm{d}f. \tag{3.8}$$

Hence, the Allan variance can be interpreted as the energy of the spectral density passed through a filter. The bandwidth of the filter depends on the cluster time T_c. The Allan variance can therefore be used to identify various noise sources present in the measurements. Typically it is presented as the Allan standard deviation $\sigma(T_c)$ versus cluster time T_c in a log–log plot.

An example of an Allan variance plot is shown in Figure 3.6. It shows the charac-

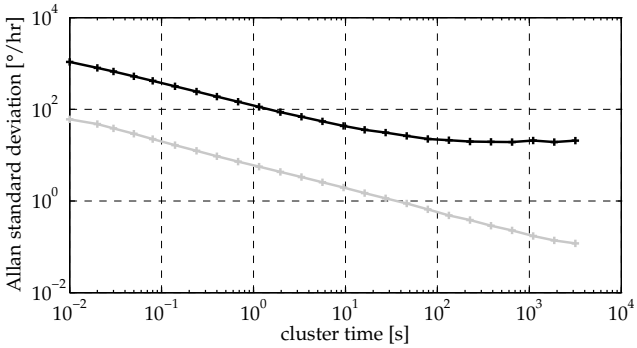


Figure 3.6: *Allan variance for gyroscopes. Shown are results from an industrial grade MEMS sensor (—) and a fiber optic sensor (—).*

teristic behavior for the Allan variance for inertial sensors, where a sensor typically has a large white noise component. This exhibits itself in the Allan variance with a slope of $-\frac{1}{2}$, since averaging over a twice as long cluster will halve the variance of the mean. However, if the cluster times increase, the bias instability will cancel this effect and the Allan variance becomes a flat line and eventually increases again. The quality differences between IMUs are mainly found in the noise magnitude and the bias instability. Higher quality sensors, such as fiber

optic gyroscopes, have less noise as well as improved bias stability. Less noise results in lower Allan variance curves and improved stability results in larger cluster times at which the Allan variance curve reaches a plateau.

The Allan variance is an easy tool to study and compare the noise characteristics of inertial sensors. However, it does not consider factors such as linearity, temperature stability and other calibration parameters related to dynamic accuracy. These effects are also very relevant for sensor quality and price. Therefore, the Allan variance should never be relied on exclusively when deciding which sensor to use in an application.

3.2 Vision

A vision sensor is a rather complex device composed of a number of sub-systems, as illustrated in Figure 3.7. The optical system bundles incident rays of light and forms an analog image, which is digitized by the image sensor. Computer vision then extracts the relevant information from the image. In this thesis we use computer vision to extract *two dimensional* (2D) features in the image and associate them to 3D points in the scene. These correspondences are considered as the measurements from the vision system.

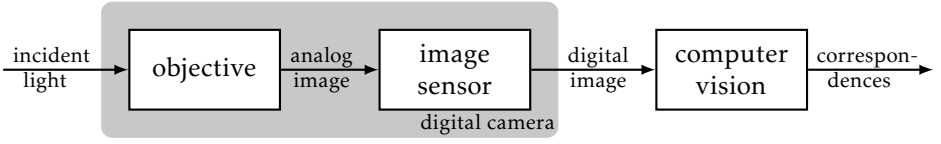


Figure 3.7: *The pipeline of a vision sensor.*

Correspondences, i.e. 2D features in the image and their associated 3D location in the scene, are the cornerstones of many computer vision applications. They can be classified as 2D bearings-only measurements: from the feature coordinates it is possible to find a line on which the corresponding 3D point lies. However, information about the distance or depth of the point is not provided. When three or more correspondences have been obtained in a single image, they can be used to calculate the position and orientation of the camera (Hartley and Zisserman, 2004; Ma et al., 2006).

3.2.1 Measurement model

The optical system or objective projects incident rays of light to points in the sensor plane. Various types of objectives exist, each with a specific application area. Examples include standard perspective lenses, wide angle lenses, zoom lenses, macro lenses and fish-eye lenses. In general, objectives are rather complex composite devices composed of a number of functional elements, see Figure 3.8. The most important elements are the so-called lens elements and stops. The lens elements have refractive surfaces which bend the light, whereas the stops limit

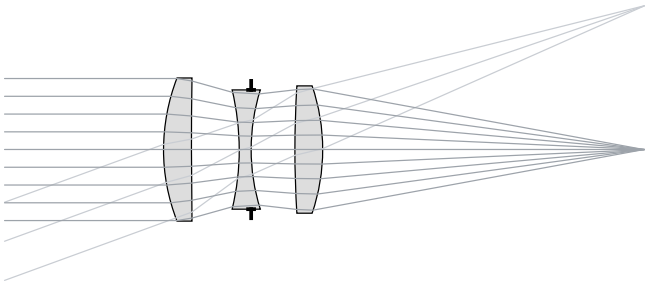


Figure 3.8: *Cross section of a low-cost objective. The triplet of lens elements bundles the parallel rays of light entering the system from the left and forms an image on the right.*

the bundle of light propagating through the system. Combining a number of elements an optical system can be designed in such a way that the desired image formation is achieved with minimal optical aberration.

From a pure geometric perspective, ignoring effects such as focus and lens thickness, the process of image formation can be described as a central projection (Hartley and Zisserman, 2004). In this projection, a ray is drawn from a point in space toward the camera center. This ray propagates through the optical system and intersects with the image plane where it forms an image of the point. The perhaps best known example of a central projection is the pinhole camera, see Figure 3.9. Its widespread use in computer vision literature can be explained

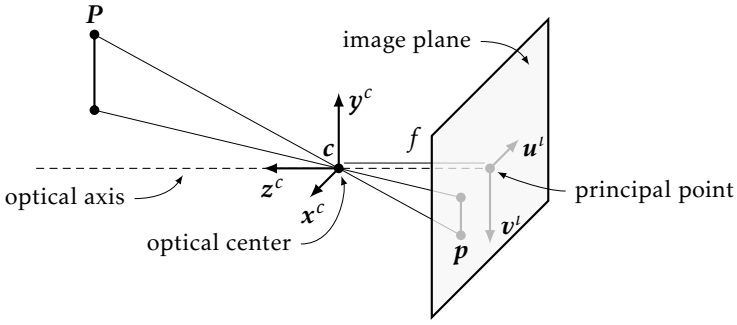


Figure 3.9: *Pinhole camera projection. The image* $\mathbf{p}$ *of a point* $\mathbf{P}$ *is the intersection point of the image plane and the line trough point* $\mathbf{P}$ *and the optical center* $\mathbf{c}$*.*

by noting that a perfect perspective objective is equivalent to a pinhole camera. To discuss the measurement model the following coordinate frames are required:

The camera frame (*c*-frame) is the coordinate frame of the moving camera. Its origin is located in the optical center of the camera, with the z-axis pointing along the optical axis.

The image frame (ι-frame) is the 2D coordinate frame of the camera images. It is located on the image plane, which is perpendicular to the optical axis.

According to the pinhole model the relation between an image point $\boldsymbol{p}_{\mathrm{a}}^{\iota} = (u, v)^T$ of the analog image and its corresponding scene point $\boldsymbol{p}^c = (X, Y, Z)^T$ is given by

$$\lambda \begin{pmatrix} u \\ v \\ f \end{pmatrix} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}, \tag{3.9}$$

where $\lambda > 0$ is a scale factor and f is the focal length. Changing from homogeneous coordinates to Euclidean coordinates yields

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{f}{Z} \begin{pmatrix} X \\ Y \end{pmatrix}. \tag{3.10}$$

This well known expression is obtained by eliminating λ from (3.9). The pinhole projection is an ideal projection and the image formation process of a real perspective objective will deviate from it, especially for low-quality objectives. An example is shown in Figure 3.10. This explains why the pinhole model is typi-

(a) *Observed image.*

(b) *Corrected image.*

Figure 3.10: *Camera images frequently suffer from radial distortion.*

cally used in combination with a compensation for radial distortion (Hartley and Zisserman, 2004; Zhang, 2000).

Although widely used in computer vision, the pinhole camera model is only suitable for perspective objectives with limited field of view. A more generic model, also suitable for wide angle lenses and omnidirectional cameras, is given by (Scaramuzza et al., 2006)

$$\lambda \begin{pmatrix} u \\ v \\ f(\rho) \end{pmatrix} = \begin{pmatrix} X \\ Y \\ Z \end{pmatrix}, \tag{3.11a}$$

with $\lambda > 0$. The constant focal length has been replaced with an n-th order poly-

nomial of the radius $\rho \triangleq \sqrt{u^2 + v^2}$,

$$f(\rho) \triangleq \sum_{i=0}^{n} \alpha_i \rho^i. \tag{3.11b}$$

Note that this model (3.11) includes the pinhole model (3.9) as a special case. Furthermore, the radial distortion is already included in the model, removing the need for a distortion compensation. To change from homogeneous coordinates to Euclidean coordinates, we solve for λ using the last line in (3.11a). After some algebra, one obtains

$$\begin{pmatrix} u \\ v \end{pmatrix} = \frac{\beta}{r} \begin{pmatrix} X \\ Y \end{pmatrix}, \tag{3.12a}$$

where $r \triangleq \sqrt{X^2 + Y^2}$ and β is the positive real root of the equation

$$\sum_{i=0}^{n} \alpha_i \beta^i - \frac{Z}{r}\beta = 0. \tag{3.12b}$$

Finding a closed form expression for this root can be very hard and is even impossible when $n > 4$. However, numerical evaluation is straightforward. A closed form expression for the derivative is given by

$$\frac{\partial \boldsymbol{p}_{\mathrm{a}}^{\imath}}{\partial \boldsymbol{p}^{c}} = \frac{\beta}{\gamma r^3} \begin{pmatrix} X \\ Y \end{pmatrix} \begin{pmatrix} XZ & YZ & -r^2 \end{pmatrix} + \frac{\beta}{r^3} \begin{bmatrix} Y^2 & -XY & 0 \\ -XY & X^2 & 0 \end{bmatrix}, \tag{3.13a}$$

with γ defined as

$$\gamma \triangleq Z - r \sum_{i=1}^{n} i \alpha_i \beta^{i-1}. \tag{3.13b}$$

This derivative is used extensively in for instance camera calibration and sensor fusion which require gradient information to minimize the error between predicted and measured feature positions.

Cameras deliver digital images with coordinates typically specified in pixels and indexed from the top left corner. Furthermore, there is the possibility of non-square as well as non-orthogonal pixels. These effects can be accounted for by an affine transformation which transforms the analog image coordinates $\boldsymbol{p}_{\mathrm{a}}^{\imath} = (u, v)^T$ into pixel coordinates $\boldsymbol{p}^{\imath} = (x, y)^T$,

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{bmatrix} s_x & s_\theta \\ 0 & s_y \end{bmatrix} \begin{pmatrix} u \\ v \end{pmatrix} + \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}. \tag{3.14}$$

Here, the transformation is composed of the pixel sizes s_x, s_y, the principal point coordinates x_0, y_0 and a skew parameter s_θ.

The camera measurements $\boldsymbol{y}_{\mathrm{c}}$ consist of the K correspondences $\{\boldsymbol{p}_k^{\imath} \leftrightarrow \boldsymbol{p}_k^{n}\}_{k=1}^{K}$ between a 2D feature in the image $\boldsymbol{p}_k^{\imath}$ and its corresponding 3D position in the n-frame $\boldsymbol{p}_k^{n}$ obtained using computer vision. Summarizing the previous paragraphs,

we model the correspondences as

$$\boldsymbol{p}_k^i = \mathcal{P}(\boldsymbol{p}_k^c) + \boldsymbol{e}_k^i, \tag{3.15}$$

where $\boldsymbol{e}_k^i$ is i.i.d. Gaussian noise. For a standard camera, the projection function $\mathcal{P}$ is composed of (3.10) and (3.14), whereas for a wide angle camera $\mathcal{P}$ is composed of (3.12) and (3.14). Note that $\mathcal{P}$ operates on $\boldsymbol{p}_k^c$. As this frame is moving, the 3D points are stored in the n-frame and the correspondences contain $\boldsymbol{p}_k^n$. To express these in the c-frame we use

$$\boldsymbol{p}_k^c = R^{cn}(\boldsymbol{p}_k^n - \boldsymbol{c}^n), \tag{3.16}$$

where R^{cn} and $\boldsymbol{c}^n$ parameterize the position and the orientation of the camera and are called the extrinsic parameters. In contrast, the parameters in (3.9)–(3.14) are called the intrinsic or internal parameters.

Although the models (3.9)–(3.14) are sufficient for many applications, they are simplifications of the imaging process and have some limitations. One of these is that the physical location of the optical center, which determines the position of the c-frame, is not well defined. Clearly, the optical center has to lie somewhere on the optical axis, but exactly where, or even whether it lies behind, inside, or in front of the objective depends highly on the typically unknown detailed design of all the elements in an objective. The location of the optical center is not of interest in pure computer vision setting. However, it becomes important when combining vision with inertial sensors in Chapter 6.

3.2.2 Calibration

Equations (3.9)–(3.14) contain a number of parameters which have to be determined individually for every camera. The process of doing so is referred to as camera calibration and is a well known problem in computer vision for which a number of toolboxes have been developed (Zhang, 2000; Bouguet, 2003; Scaramuzza et al., 2006). Typically, these require images at several angles and distances of a known calibration object. A planar checkerboard pattern, see Figure 3.11, is a frequently used calibration object because it is very simple to produce and it can be printed with a standard printer and has distinctive corners which are easy to detect. From the images of the calibration pattern correspondences $\boldsymbol{p}_k^i \leftrightarrow \boldsymbol{p}_k^n$ are constructed. In general this is a difficult problem, but due to the strong corners and simple planar geometry it is relatively easy to obtain the correspondences. The required image processing is typically implemented in off-the-shelf camera calibration software.

The goal of the calibration problem is to choose the intrinsic parameters such that the obtained correspondences are as likely as possible. This cannot be done without determining the extrinsic parameters of the calibration images as well. Under the standard assumptions of i.i.d. Gaussian noise, the corresponding ML

Figure 3.11: *A typical image used for camera calibration. The calibration object is a planar checkerboard pattern with known dimensions.*

problem can be formulated as

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \sum_{k=1}^{K} \tfrac{1}{2} \| \boldsymbol{p}_k^i - \mathcal{P}'(\boldsymbol{p}_k^n, \boldsymbol{\theta}) \|_2^2, \tag{3.17}$$

where $\mathcal{P}'$ is the projection function combining (3.15) and (3.16) and the parameter vector $\boldsymbol{\theta}$ consists of the intrinsic and extrinsic parameters. This optimization problem can be solved using standard algorithms from nonlinear optimization as discussed in Section 2.3. An initial guess of the parameters can be obtained from the homographies, the one-to-one relations that exist between the images and the planar calibration pattern (Zhang, 2000). In the remainder of this thesis, we assume that the camera has been calibrated.

3.2.3 Correspondence generation

In general, finding the correspondences is a difficult image processing problem where two tasks have to be solved. The first task consists of detecting points of interest or features in the image. Here, features are distinctive elements in the camera image, for instance, corners, edges, or textured areas. Common algorithms include the gradient based Harris detector (Harris and Stephens, 1988), the Laplace detector (Mikolajczyk et al., 2005), and the correlation based Kanade-Lucas-Tomasi tracker (Shi and Tomasi, 1994).

Once a feature has been found, it needs to be associated to a known 3D point in the scene in order to form a correspondence. This is the second task, which can be solved using probabilistic methods such as RANSAC (Fischler and Bolles, 1981). However, it can be drastically simplified by making use of some kind of descriptor of the feature which uniquely identifies it by providing information of

the local image such as image patches or local histograms. This descriptor should preferably be invariant to scale changes and affine transformations. Common examples are SIFT (Lowe, 2004) and more recently SURF (Bay et al., 2008) and FERNS (Ozuysal et al., 2007).

Once three or more correspondences have been obtained in a single image, they can be used to calculate the position and orientation of the camera (Hartley and Zisserman, 2004; Ma et al., 2006). This is actually a reduced version of the camera calibration problem (3.17), where in this case only the extrinsic parameters are sought as the intrinsic parameters are already known.

Correspondences can also be used to find the 3D position of a feature. In the simplest case this can be done using the epipolar geometry of a correspondence which is observed in two images taken from different locations. Extensions to multiple images exist as well (Hartley and Zisserman, 2004). These are the basis for structure from motion or bundle adjustment algorithms (Triggs et al., 2000) as well as visual *simultaneous localization and mapping* (SLAM) applications (Davison et al., 2007; Klein and Murray, 2007).

3.3 Ultra-wideband

Ultra-wideband (UWB) is a relatively new and promising radio technology with applications in radar, communication and localization. It occupies a very large frequency band, at least 0.5 GHz, as opposed to more traditional radio communications which operate in much smaller frequency bands. Since UWB is only allowed to transmit at very low power, see Figure 3.12, its spectrum becomes very noise-like and it can coexist with other services without influencing them.

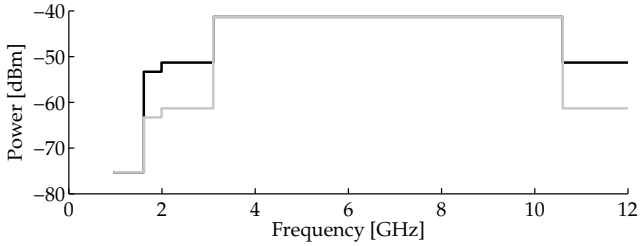


Figure 3.12: *US UWB emission masks for indoor (–) and handheld (··) usage.*

To obtain the aforementioned spectrum, UWB technology typically makes use of impulse radio using very short pulses, typically in the order of 1 ns, resulting in a high spatial resolution. This characteristic makes UWB very suitable for localization applications. The localization technologies can roughly be subdivided into three categories: systems using time, systems using angle-of-arrival and systems using signal strength (Gezici et al., 2005; Sahinoglu et al., 2008). In this thesis we focus on time-based methods, where position is inferred from the time it takes

for a signal to travel from the transmitter to the receiver.

Among the more mature UWB applications for indoor positioning are the so-called asset tracking systems in for instance health-care or manufacturing. Commercially available systems (e.g. Time Domain, 2010; Ubisense, 2010) typically consist of a network of synchronized UWB receivers which track a large number of small, battery powered and inexpensive UWB transmitters, shown in Figure 3.13. The position of the transmitters is determined from *time of arrival* (TOA) mea-

Figure 3.13: *UWB hardware: a receiver (left) next to a small, mobile transmitter (right).*

surements. Reported indoor position accuracies lie in the order of decimeters (Time Domain, 2010; Ubisense, 2010), but suffer from multipath effects and *non line of sight* (NLOS) conditions.

3.3.1 Measurement model

The UWB setup consists of a network of synchronized and stationary (rigidly fixed, mounted) receivers, all taking very precise TOA measurements of signals originating from a mobile transmitter as illustrated by Figure 3.14. Although the

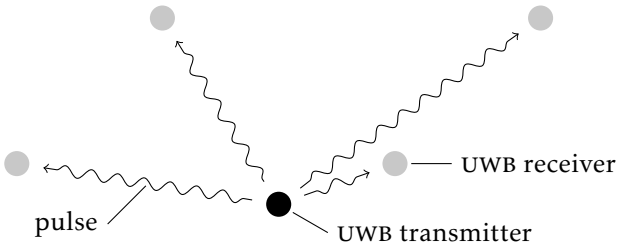


Figure 3.14: *The UWB setup consists of a number of stationary receivers making TOA measurements of signals originating from a mobile transmitter.*

receivers are synchronized to a central clock, they each have a small, constant clock offset due to for instance cable lengths. From the time at which the signal is transmitted it takes a while before it arrives at the receivers. Ideally, the signal travels directly to a receiver and encounters no medium which delays and/or attenuates it, in which case the time of flight is related to the distance traveled. Alternatively the signal can be reflected one or more times before it is received, also introducing a delay. The time at which the signal is transmitted is an unknown parameter, which makes the TOA measurements very similar to the GPS pseudorange measurements discussed in Section 3.4.

In summary, the TOA measurement $y_{\mathrm{u},m}$ of the UWB system is the time (according to its local clock) when receiver m receives a pulse from the transmitter. It can be modeled as

$$y_{\mathrm{u},m} = \tau + \|\boldsymbol{r}_m^n - \boldsymbol{t}^n\|_2 + \Delta\tau_m + \delta_{\mathrm{u},m} + e_{\mathrm{u},m}, \tag{3.18}$$

where τ is the time of transmission of the pulse, $\boldsymbol{t}$ is the position of the transmitter, $\boldsymbol{r}_m$ is the position of the m-th receiver and $\Delta\tau_m$ is the clock-offset of the m-th receiver. $\delta_{\mathrm{u},m} \geq 0$ is a possibly nonzero delay due to NLOS or multipath and $e_{\mathrm{u},m}$ is i.i.d. Gaussian noise which is virtually independent of range. Without loss of generality, all quantities in (3.18) are expressed in meters (using the speed of light when necessary) and the positions are resolved in the n-frame.

3.3.2 Calibration

The TOA measurements can be used to determine the position of the transmitter. Methods for doing so are discussed in Section 3.3.3. They require a calibrated setup; that is, they assume the positions of the receivers $\boldsymbol{r}_m$ to be known in the n-frame as well as known receiver clock offsets $\Delta\tau_m$.

Although a correct calibration of the setup is a prerequisite for accurate positioning results, it is typically taken for granted and calibration is seldom discussed. To the best of the author's knowledge, current calibration methods have been developed by the hardware manufacturers and are documented in their manuals. These methods require receiver positions and transmitter positions to be surveyed. The surveying of positions is typically a time-consuming and error-prone process which requires additional equipment. Such an elaborate procedure is only feasible for permanent setups, severely limiting the deployment of a UWB positioning system.

A major contribution of this thesis is a flexible and easy-to-use calibration algorithm capable of calibrating a UWB setup without additional measurements and hardware in a couple of minutes, thereby enabling flexible and portable UWB positioning systems. It is discussed in Section 7.2, where we apply maximum likelihood estimation to estimate the 3D receiver positions $\{\boldsymbol{r}_m^n\}_{m=1}^M$ as well as the receiver clock-offsets $\{\Delta\tau_m\}_{m=1}^M$ of all M receivers.

3.3.3 Multilateration

The process of determining the transmitter position from the TOA measurements is referred to as trilateration or, more accurately, multilateration. It is a well-studied topic for which many algorithms are reported in literature (e.g. Chan and Ho, 1994; Gezici et al., 2005; Sayed et al., 2005; Sahinoglu et al., 2008). The majority of these algorithms assume 'clean' measurements without delays, i.e. $\delta_{\mathrm{u},m} = 0$.

A common technique is to eliminate the time of transmission τ from (3.18) by constructing *time difference of arrival* (TDOA) measurements from pairs of TOA measurements. The resulting set of hyperbolic equations can then be efficiently solved for position. The drawback of this approach is that the constructed TDOA measurements are no longer independently distributed which complicates the calculations. An equivalent approach is to treat τ as an unknown and solve for both position and time. Assuming Gaussian noise, ML estimation takes the form of a nonlinear least squares problem,

$$\min_{\boldsymbol{t}^n,\tau} \sum_{m=1}^{M} \frac{1}{2} \|y_{\mathrm{u},m} - \tau - \|\boldsymbol{r}_m^n - \boldsymbol{t}^n\|_2 - \Delta\tau_m\|_{\Sigma_{\mathrm{u}}^{-1}}^2, \tag{3.19}$$

where Σ_{u} is the covariance of the TOA measurements. It can be efficiently solved using standard algorithms from nonlinear numerical optimization as discussed in Section 2.3.

The presence of non-zero delays severely affects the accuracy of the estimated position, giving rise to several ad hoc methods to detect which measurements are corrupted. Typically, multilateration is performed for a number of subsets of the measurements and the 'best' solution is returned. In Section 7.3 we present a novel multilateration method based on ℓ_1 regularization which automatically determines which measurements have a non-zero delay δ_{u} and corrects for it.

3.4 Global positioning system

The *global positioning system* (GPS) is a satellite-based positioning system suitable for outdoor applications. In its basic configuration, it consists of 24 space vehicles in 6 almost circular orbits around the earth continuously transmitting radio signals. A user can determine its position and velocity worldwide by receiving and decoding these signals from a number of satellites (Kaplan and Hegarty, 2006; Misra and Enge, 2006). This makes GPS a specific type of *global navigation satellite system* (GNNS). Other systems are for instance GALILEO and GLONAS.

GPS has been operational since 1993 and is operated by the USA. Both civil and military signals are available, where the latter are encrypted. The satellites transmit on two frequencies, called L1 (1.6 GHz) and L2 (1.2 GHz), each using a different code. Besides ranging signals, navigation data containing for instance the orbit parameters of the space vehicles is transmitted. The latter data, the so-called ephemeris data, allows the user to calculate the instantaneous position of

the satellites and to compensate for atmospheric effects. The GPS system can be augmented using base stations broadcasting local correction terms. *Differential GPS* (DGPS) and *satellite-based augmentation system* (SBAS) approaches use the information from these base stations to improve the accuracy.

Commercial receivers are available in a wide range of prices. Dual band receivers form the high-end segment and give horizontal accuracies of 1 m in autonomous mode (Misra and Enge, 2006). Single band receivers are nowadays available as single chip packages and give position with a horizontal accuracy of 10 m under good visibility (Misra and Enge, 2006). These cheap devices have opened up many consumer applications, most notable car navigation systems.

3.4.1 Measurement model

The GPS system nominally consists of 24 satellites with synchronized atomic clocks in non-geostationary orbits around the earth. They transmit time-stamped signals which are received by a user on earth, as illustrated by Figure 3.15. A

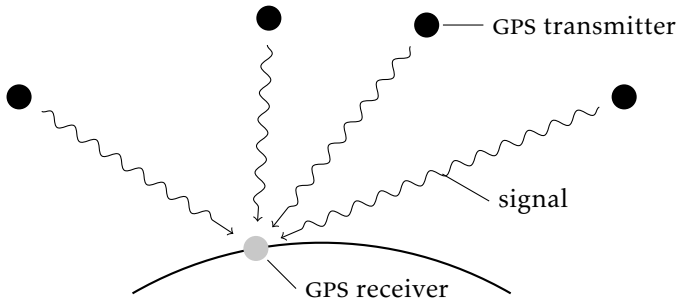


Figure 3.15: *A GPS receiver receives signals transmitted by multiple GPS satellites.*

GPS receiver can obtain three independent types of measurements from the received signals (Kaplan and Hegarty, 2006; Misra and Enge, 2006): pseudorange, Doppler and carrier phase measurements. These will be discussed below.

The code phase tracking loop in a GPS receiver compares the received signal from a satellite to a receiver generated copy. The delay between these signals is a pseudorange measurement. After a number of standard corrections this measurement is unambiguously related to the traveled distance as well as the receiver clock offset. That is, the pseudorange measurement y_ρ from the k-th satellite is given by

$$y_{\rho,k} = \|\boldsymbol{t}_k^e - \boldsymbol{r}^e\|_2 + \Delta t + \delta_{\rho,k} + e_{\rho,k}, \tag{3.20}$$

where $\boldsymbol{t}_k$ is the position of the transmitter of the k-th satellite, $\boldsymbol{r}$ is the position of the receiver and Δt is the receiver clock offset. Furthermore, $\delta_{\rho,k}$ is a residual error term resulting from e.g. uncompensated atmospheric or multipath effects and $e_{\rho,k}$ is i.i.d. Gaussian noise. Without loss of generality, all quantities in (3.20) are

expressed in meters (using the speed of light when necessary) and the positions are resolved in the e-frame.

The relative motion between a satellite and the receiver results in a Doppler shift of the carrier frequency. This shift is measured in the carrier tracking loop of the receiver. After compensation for known terms, the Doppler measurement $y_{\mathrm{d},k}$ of the k-th satellite is given by

$$y_{\mathrm{d},k} = -(\boldsymbol{u}_k^e)^T \dot{\boldsymbol{r}}_e^e + \dot{\Delta t} + e_{\mathrm{d},k}, \tag{3.21}$$

where $\boldsymbol{u}_k = \|\boldsymbol{t}_k - \boldsymbol{r}\|_2^{-1}(\boldsymbol{t}_k - \boldsymbol{r})$ is the unit vector pointing from the receiver to the satellite, $\dot{\boldsymbol{r}}_e$ is the receiver velocity w.r.t. the e-frame, $\dot{\Delta t}$ is the receiver clock drift and $e_{\mathrm{d},k}$ is i.i.d. Gaussian noise. Without loss of generality, all quantities in (3.21) are expressed in meters (using the speed of light when necessary) and the positions and the velocities are resolved in the e-frame.

Similar to the code phase tracking loop, the carrier phase tracking loop compares the received signal from a satellite to a receiver generated copy. The phase difference, which can be measured very accurately, is related to the traveled distance and an integer number of wavelengths. This makes it an ambiguous measurement, in contrast to the pseudorange measurements. Carrier phase measurements are typically used in a dual receiver configuration which allows for the elimination of atmospheric and receiver clock errors. From a pair of receivers close to each other, a and b, both observing two satellites so-called double differenced carrier phase measurements can be formed. The double differenced carrier phase measurement $y_{\phi,kl}$ from the satellites k and l is given by

$$y_{\phi,kl} = (\boldsymbol{u}_{kl}^e)^T(\boldsymbol{r}_a^e - \boldsymbol{r}_b^e) + \lambda N_{kl} + e_{\phi,kl}, \tag{3.22}$$

where $\boldsymbol{u}_{kl} = \boldsymbol{u}_k - \boldsymbol{u}_l$ is the difference of unit vectors, $\boldsymbol{r}$ is a receiver position, λ is the wavelength, $N_{kl} \in \mathbb{Z}$ is an integer number of carrier cycles and $e_{\phi,kl}$ is i.i.d. Gaussian noise. Note that correlations are introduced between measurements with common indices due to the differencing. Without loss of generality, all quantities in (3.22) are expressed in meters (using the speed of light when necessary) and the positions are resolved in the e-frame. The double difference carrier phase measurements are much more accurate than pseudorange measurements and due to the differencing common errors are canceled. The downside is the introduction of an integer ambiguity. This ambiguity stays constant as long as the tracking loop does not lose track.

3.4.2 Calibration

GPS does not require any calibration by the user. All necessary parameters are contained in the navigation data broadcast by the satellites. The GPS signals are continuously monitored by base stations and when necessary, updates are uploaded to the satellites.

3.4.3 Position, velocity and time estimation

In most cases, a GPS receiver does not only provide measurements, but also estimates its *position, velocity and time* (PVT).

The receiver position $\boldsymbol{r}$ and the receiver clock offset Δt can be solved from the pseudorange measurements. Assuming Gaussian noise and unbiased measurements, i.e., $\delta_{\rho,k} = 0$, ML estimation takes the form of a nonlinear least squares problem,

$$\min_{\boldsymbol{r}^e, \Delta t} \sum_{k=1}^{K} \frac{1}{2} \| y_{\rho,k} - \| \boldsymbol{t}_k^e - \boldsymbol{r}^e \|_2 - \Delta t \|_{\Sigma_\rho^{-1}}^2, \tag{3.23}$$

where Σ_ρ is the covariance of the pseudorange measurements. It can be efficiently solved using standard algorithms from nonlinear numerical optimization as discussed in Section 2.3. The accuracy of the estimates is given as

$$\operatorname{Cov} \begin{pmatrix} \hat{\boldsymbol{r}}^e \\ \hat{\Delta t} \end{pmatrix} = (J^T J)^{-1} \Sigma_\rho, \qquad J = \begin{bmatrix} -(\boldsymbol{u}_1^e)^T & 1 \\ \vdots & \vdots \\ -(\boldsymbol{u}_K^e)^T & 1 \end{bmatrix}, \tag{3.24}$$

where J is evaluated at $\hat{\boldsymbol{r}}^e$ and depends on the constellation of satellites. Using the estimate of the receiver clock offset, the receiver clock can be synchronized to GPS time after which the time of the measurement is known. Without loss of generality, the position estimate is typically presented as latitude, longitude and altitude.

The receiver velocity $\dot{\boldsymbol{r}}$ can be solved from the Doppler measurements. Assuming Gaussian noise, ML estimation takes the form of a least squares problem with solution

$$\begin{pmatrix} \hat{\dot{\boldsymbol{r}}}^e \\ \hat{\dot{\Delta t}} \end{pmatrix} = (J^T J)^{-1} J^T \begin{pmatrix} y_{\mathrm{d},1} \\ \vdots \\ y_{\mathrm{d},K} \end{pmatrix}, \qquad \operatorname{Cov} \begin{pmatrix} \hat{\dot{\boldsymbol{r}}}^e \\ \hat{\dot{\Delta t}} \end{pmatrix} = (J^T J)^{-1} \Sigma_{\mathrm{d}}, \tag{3.25}$$

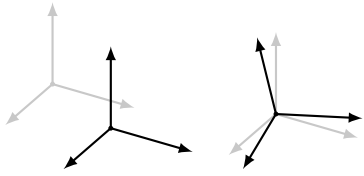
with J defined in (3.24). Without loss of generality, the velocity estimate is typically presented w.r.t. a local tangent plane coordinate frame with unit vectors pointing north, east and down, or alternatively north, west and up.

4

Kinematics

Kinematics deals with motion in absence of considerations of mass and force. It involves assigning coordinate frames to rigid bodies and describing how these move over time. It provides the connection between the sensors used in this thesis.

A general, length preserving transformation between two Cartesian coordinate frames consists of a translation and/or a rotation. These elementary transformations are illustrated in Figure 4.1. A translation is defined as a displacement of



(a) *Translation.* ***(b)*** *Rotation.*

Figure 4.1: *Elementary coordinate frame transformations.*

the origin while keeping the axes aligned, whereas a rotation is a change in the direction of the axes while keeping the origins coincident. These transformations and their properties are the topic of this chapter.

4.1 Translation

The translation of a coordinate frame corresponds to a displacement of its origin, while keeping the axes aligned. A point has new coordinate vectors in the

translated frame, see Figure 4.2. It follows that a point $\boldsymbol{x}$ can be expressed in the

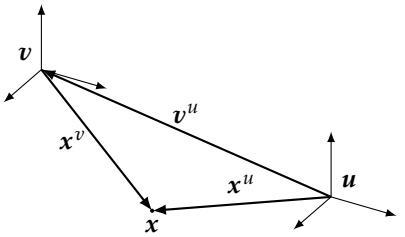


Figure 4.2: *Translation of a coordinate frame.*

translated u-frame using

$$\boldsymbol{x}^u \triangleq \boldsymbol{x}^v + \boldsymbol{v}^u. \tag{4.1}$$

Here, $\boldsymbol{x}^u$ is the position of the point $\boldsymbol{x}$ expressed in the u-frame, $\boldsymbol{x}^v$ is the position of the point $\boldsymbol{x}$ expressed in the v-frame and $\boldsymbol{v}^u$ is the position of the v-frame expressed in the u-frame. The inverse transformation is given by

$$\boldsymbol{x}^v \triangleq \boldsymbol{x}^u + \boldsymbol{u}^v, \qquad \boldsymbol{u}^v = -\boldsymbol{v}^u. \tag{4.2}$$

The latter expression can be obtained from solving (4.1) for $\boldsymbol{x}^v$ and comparing the result to the definition of the inverse transformation.

4.2 Rotation

A rotation of a coordinate frame corresponds to changing the direction of the coordinate axes, while the origin remains where it is. To introduce this concept it is convenient to consider the active form of rotation, i.e. the rotation of a vector. To arrive at the effect of rotating the coordinate frame, note that a counter clockwise rotation of the coordinate frame is equivalent to a clockwise rotation of a vector.

A geometric interpretation of vector rotation is given in Figure 4.3. The point

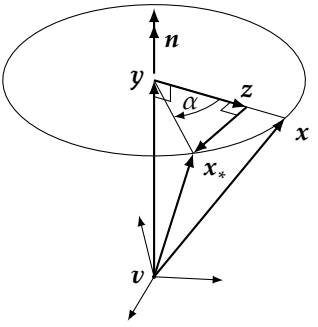


Figure 4.3: *Clockwise rotation of vector $\boldsymbol{x}$ by angle α around unit axis $\boldsymbol{n}$.*

$\boldsymbol{x}_*$ is obtained by a clockwise rotation of $\boldsymbol{x}$ around the unit axis $\boldsymbol{n}$ by an angle α. Using $\boldsymbol{y}$ and $\boldsymbol{z}$ as defined in Figure 4.3, the rotation can be decomposed as

$$\begin{aligned}\boldsymbol{x}_*^v &= \boldsymbol{y}^v + (\boldsymbol{z}^v - \boldsymbol{y}^v) + (\boldsymbol{x}_*^v - \boldsymbol{z}^v) \\ &= (\boldsymbol{x}^v \cdot \boldsymbol{n}^v)\boldsymbol{n}^v + (\boldsymbol{x}^v - (\boldsymbol{x}^v \cdot \boldsymbol{n}^v)\boldsymbol{n}^v)\cos\alpha + (\boldsymbol{x}^v \times \boldsymbol{n}^v)\sin\alpha \\ &= \boldsymbol{x}^v \cos\alpha + \boldsymbol{n}^v(\boldsymbol{x}^v \cdot \boldsymbol{n}^v)(1 - \cos\alpha) - (\boldsymbol{n}^v \times \boldsymbol{x}^v)\sin\alpha,\end{aligned}$$

where all quantities are resolved in the v-frame. This implies that a counter clockwise rotation of the coordinate frame is given by

$$\boldsymbol{x}^u = \boldsymbol{x}^v \cos\alpha + \boldsymbol{n}^v(\boldsymbol{x}^v \cdot \boldsymbol{n}^v)(1 - \cos\alpha) - (\boldsymbol{n}^v \times \boldsymbol{x}^v)\sin\alpha, \tag{4.3}$$

where the u-frame is the rotated frame. This equation is commonly referred to as the rotation formula or Euler's formula (Shuster, 1993). Note that the combination of $\boldsymbol{n}$ and α, or equivalently $\boldsymbol{\phi} = \boldsymbol{n}\alpha$ is denoted as the axis-angle parameterization or the rotation vector.

Rotations can be described using a number of different parameterizations, see e.g. Shuster (1993) for an overview. Commonly encountered parameterizations include rotation matrices, rotation vectors, Euler angles and unit quaternions. Using the rotation matrix parameterization, rotations are defined as

$$\boldsymbol{x}^u \triangleq R^{uv}\boldsymbol{x}^v. \tag{4.4}$$

The rotation matrix R^{uv} is a member of the special orthogonal group SO(3),

$$\mathrm{SO}(3) = \{R \in \mathbb{R}^{3\times 3} : RR^T = I, \det R = +1\}. \tag{4.5}$$

The inverse transformation of (4.4) is given by

$$\boldsymbol{x}^v \triangleq R^{vu}\boldsymbol{x}^u, \qquad R^{vu} = (R^{uv})^T. \tag{4.6}$$

The latter identity is straightforwardly obtained from solving (4.4) for $\boldsymbol{x}^v$. The equivalence between (4.3) and (4.4) can be made explicitly using some properties of the cross product. The cross product can written as a matrix vector product,

$$\boldsymbol{a} \times \boldsymbol{b} = [\boldsymbol{a}\times]\boldsymbol{b}, \qquad [\boldsymbol{a}\times] \triangleq \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}. \tag{4.7}$$

Furthermore, multiple cross products can be expanded as

$$\boldsymbol{u} \times (\boldsymbol{v} \times \boldsymbol{w}) = \boldsymbol{v}(\boldsymbol{w} \cdot \boldsymbol{u}) - \boldsymbol{w}(\boldsymbol{u} \cdot \boldsymbol{v}). \tag{4.8}$$

Using these relations, (4.3) can be rewritten as

$$\begin{aligned}\boldsymbol{x}^u &= \boldsymbol{x}^v \cos\alpha + \boldsymbol{n}^v(\boldsymbol{x}^v \cdot \boldsymbol{n}^v)(1 - \cos\alpha) - (\boldsymbol{n}^v \times \boldsymbol{x}^v)\sin\alpha \\ &= \boldsymbol{x}^v \cos\alpha + (\boldsymbol{n}^v \times (\boldsymbol{n}^v \times \boldsymbol{x}^v) + \boldsymbol{x}^v)(1 - \cos\alpha) - (\boldsymbol{n}^v \times \boldsymbol{x}^v)\sin\alpha \\ &= \underbrace{\left[I - (\sin\alpha)[\boldsymbol{n}^v\times] + (1 - \cos\alpha)[\boldsymbol{n}^v\times]^2\right]}_{\triangleq R^{uv}(\boldsymbol{n}^v,\alpha)}\boldsymbol{x}^v.\end{aligned} \tag{4.9}$$

This shows that the rotation formula (4.3) can be written in the form of (4.4). The

obtained matrix is identified as the rotation matrix R^{uv}.

Euler angles define a rotation using a series of three rotations around a coordinate axis. In case of the popular aerospace sequence, one rotates first an angle ψ about the z-axis, then an angle θ around the y-axis and finally an angle ϕ around the x-axis. Using (4.9), we obtain

$$\begin{aligned} R^{uv} &= R(\boldsymbol{e}_1, \phi) R(\boldsymbol{e}_2, \theta) R(\boldsymbol{e}_3, \psi) \\ &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix}. \end{aligned} \tag{4.10}$$

Here, the convention is to call the Euler angles (ϕ, θ, ψ) roll, pitch and yaw or bank, elevation and heading.

Alternatively, a rotation can be defined using unit quaternions as

$$x^u \triangleq q^{uv} \odot x^v \odot (q^{uv})^c. \tag{4.11}$$

Here, $\{x^u, x^v\} \in \mathcal{Q}_v$ are the quaternion equivalents of $\{\boldsymbol{x}^u, \boldsymbol{x}^v\}$, $q^{uv} \in \mathcal{Q}_1$ is a unit quaternion describing the rotation from the v-frame to the u-frame and $\odot$, $\cdot^c$ denote quaternion multiplication and conjugation respectively. Details on quaternions and their properties can be found in Appendix B. The inverse transformation is given by

$$x^v \triangleq q^{vu} \odot x^u \odot (q^{vu})^c, \qquad q^{vu} = (q^{uv})^c. \tag{4.12}$$

The latter identity can be obtained by solving (4.11) for x^v and comparing the result to the definition of the inverse transformation. The connection to (4.3) can be made explicit by expanding the quaternion products in (4.11) and substituting $q^{uv} = (q_0, \boldsymbol{q}) = (\cos\frac{\alpha}{2}, -\boldsymbol{n}^v \sin\frac{\alpha}{2})$. This gives

$$(0, \boldsymbol{x}^u) = \left(0, q_0^2 \boldsymbol{x}^v + (\boldsymbol{q} \cdot \boldsymbol{x}^v)\boldsymbol{q} + \boldsymbol{q} \times (\boldsymbol{q} \times \boldsymbol{x}^v) + 2q_0(\boldsymbol{q} \times \boldsymbol{x}^v)\right) \tag{4.13}$$

$$= \left(0, (q_0^2 - \boldsymbol{q} \cdot \boldsymbol{q})\boldsymbol{x}^v + 2(\boldsymbol{q} \cdot \boldsymbol{x}^v)\boldsymbol{q} + 2q_0(\boldsymbol{q} \times \boldsymbol{x}^v)\right) \tag{4.14}$$

$$= (0, \boldsymbol{x}^v \cos\alpha + \boldsymbol{n}^v(\boldsymbol{x}^v \cdot \boldsymbol{n}^v)(1 - \cos\alpha) - (\boldsymbol{n}^v \times \boldsymbol{x}^v)\sin\alpha). \tag{4.15}$$

Note that the vector component is identical to (4.3). The relation to the rotation matrix parameterization (4.4) is given by

$$x^u = q^{uv} \odot x^v \odot (q^{uv})^c = [(q^{uv})^L][(q^{uv})^R]^T x^v = \begin{bmatrix} 1 & 0_{1\times 3} \\ 0_{3\times 1} & R^{uv} \end{bmatrix} \begin{bmatrix} 0 \\ \boldsymbol{x}^v \end{bmatrix}, \tag{4.16}$$

see also Example B.2.

All rotation parameterizations describe the same quantity and can be converted to each other, see Appendix C. They can be used interchangeably. However, there are differences in the number of parameters, singularities, uniqueness and the dif-

ficulty of the corresponding differential equations describing the evolution over time. In this thesis unit quaternions are used. The rationale for doing so is that unit quaternions give a non-singular representation with only 4 parameters. Furthermore, they have a rather simple bilinear differential equation which can be integrated analytically. In contrast, Euler angles and rotation vectors have only three parameters, but have a nonlinear differential equation. Additionally, Euler angles suffer from singularities. Rotation matrices have at least 6 parameters and need to satisfy orthogonality constraints. Although quaternions are used for all the calculations, rotation matrices are occasionally used to simplify notation and give insight. Furthermore, Euler angles provide a relatively intuitive representation which is suitable for *two dimensional* (2D) visualization purposes.

4.3 Time derivatives

Position and orientation are not directly measured by an *inertial measurement unit* (IMU). Instead the time derivatives angular velocity and acceleration are measured, see Section 3.1. In this section these quantities are introduced and it is shown how position and orientation can be obtained from them.

The time derivative of a rotation is given by

$$\dot{q}^{uv} = \tfrac{1}{2}\omega^{u}_{uv} \odot q^{uv} = q^{uv} \odot \tfrac{1}{2}\omega^{v}_{uv}. \tag{4.17}$$

To derive this expression, note that a rotation can be decomposed into incremental rotations (see e.g. Shuster, 1993). Hence, it is possible to decompose a time-varying quaternion as

$$q^{uv}(t+\delta t) = \delta q \odot q^{uv}(t) = \left(\cos\tfrac{\delta\theta}{2}, \boldsymbol{n}^u \sin\tfrac{\delta\theta}{2}\right) \odot q^{uv}(t),$$

for some rotation axis $\boldsymbol{n}^u$ and angle increment $\delta\theta$. Differentiating this expression *with respect to* (w.r.t.) time results in

$$\begin{aligned}\dot{q}^{uv}(t) &\triangleq \lim_{\delta t\to 0} \frac{q^{uv}(t+\delta t) - q^{uv}(t)}{\delta t} \\ &= \lim_{\delta t\to 0} \frac{\delta q - 1}{\delta t} \odot q^{uv}(t) = \tfrac{1}{2}\omega^{u}_{uv}(t) \odot q^{uv}(t).\end{aligned}$$

Here, ω^{u}_{uv}, the instantaneous angular velocity from the v-frame to the u-frame expressed in the u-frame, is defined as

$$\omega^{u}_{uv} \triangleq 2 \lim_{\delta t\to 0} \frac{\left(\cos\frac{\delta\theta}{2} - 1, \boldsymbol{n}^u \sin\frac{\delta\theta}{2}\right)}{\delta t} = \left(0, \lim_{\delta t\to 0} \boldsymbol{n}^u \frac{\delta\theta}{\delta t}\right). \tag{4.18}$$

Note that the angular velocity $\omega^{u}_{uv} \in \mathcal{Q}_v$, i.e., it is a vector. The second equality of (4.17) is derived using a similar argument. From (4.17) we also obtain

$$\begin{aligned}\dot{q}^{uv} &\triangleq q^{uv} \odot \tfrac{1}{2}\omega^{v}_{uv} \\ &\triangleq \tfrac{1}{2}\omega^{u}_{uv} \odot q^{uv} = q^{uv} \odot q^{vu} \odot \tfrac{1}{2}\omega^{u}_{uv} \odot q^{uv}, \\ \dot{q}^{vu} &\triangleq q^{vu} \odot \tfrac{1}{2}\omega^{u}_{vu}\end{aligned}$$

$$= \tfrac{\mathrm{d}}{\mathrm{d}t}((q^{uv})^c) = (\dot{q}^{uv})^c \triangleq (\tfrac{1}{2}\omega^u_{uv} \odot q^{uv})^c = q^{vu} \odot -\tfrac{1}{2}\omega^u_{uv}.$$

This implies that the identities

$$\omega^u_{uv} = q^{uv} \odot \omega^v_{uv} \odot q^{vu}, \qquad \omega^u_{uv} = -\omega^u_{vu}, \tag{4.19}$$

can be used to change the coordinate frames of the angular velocity.

Suppose that the u-frame and the v-frame are rotated w.r.t. each other. That is, with slight abuse of notation, position vectors are transformed as

$$\boldsymbol{x}^u = R^{uv}\boldsymbol{x}^v = q^{uv} \odot x^v \odot q^{vu}. \tag{4.20}$$

Using (4.17), this expression can be differentiated to obtain

$$\begin{aligned}\dot{x}^u &= \dot{q}^{uv} \odot x^v \odot q^{vu} + q^{uv} \odot x^v \odot \dot{q}^{vu} + q^{uv} \odot \dot{x}^v \odot q^{vu} \\ &= \tfrac{1}{2}\omega^u_{uv} \odot q^{uv} \odot x^v \odot q^{vu} - q^{uv} \odot x^v \odot q^{vu} \odot \tfrac{1}{2}\omega^u_{uv} + q^{uv} \odot \dot{x}^v \odot q^{vu} \\ &= \omega^u_{uv} \times x^u + q^{uv} \odot \dot{x}^v \odot q^{vu},\end{aligned} \tag{4.21a}$$

where $\times$ is the quaternion cross product, see Appendix B.1. Since this operator extends the vector cross product, (4.21a) is equivalent to

$$\dot{\boldsymbol{x}}^u = \boldsymbol{\omega}^u_{uv} \times \boldsymbol{x}^u + R^{uv}\dot{\boldsymbol{x}}^v. \tag{4.21b}$$

To avoid confusion later on, we write (4.21b) as

$$\dot{\boldsymbol{x}}^u_u = \boldsymbol{\omega}^u_{uv} \times \boldsymbol{x}^u + \dot{\boldsymbol{x}}^u_v, \tag{4.21c}$$

where $\dot{\boldsymbol{x}}^u_u$ and $\dot{\boldsymbol{x}}^u_v$ are defined according to

$$\dot{\boldsymbol{x}}^u_u \triangleq \tfrac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}^u, \qquad \dot{\boldsymbol{x}}^u_v \triangleq R^{uv}\dot{\boldsymbol{x}}^v_v = R^{uv}\tfrac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}^v. \tag{4.22}$$

The additional subscript denotes the coordinate frame in which the differentiation is performed.

The inertial frame (i-frame), the earth frame (e-frame), the navigation frame (n-frame) and the body frame (b-frame) have been introduced in Section 3.1. The origins of the i-frame and the e-frame coincide and the transformation between these systems is a pure rotation given by

$$\boldsymbol{b}^i = R^{ie}\boldsymbol{b}^e. \tag{4.23}$$

Differentiating (4.23) twice w.r.t. time using (4.21), we obtain

$$\begin{aligned}\dot{\boldsymbol{b}}^i_i &= \boldsymbol{\omega}^i_{ie} \times \boldsymbol{b}^i + \dot{\boldsymbol{b}}^i_e, \\ \ddot{\boldsymbol{b}}^i_{ii} &= \dot{\boldsymbol{\omega}}^i_{ie} \times \boldsymbol{b}^i + \boldsymbol{\omega}^i_{ie} \times \dot{\boldsymbol{b}}^i_i + \ddot{\boldsymbol{b}}^i_{ei} \\ &= \boldsymbol{\omega}^i_{ie} \times \boldsymbol{\omega}^i_{ie} \times \boldsymbol{b}^i + \boldsymbol{\omega}^i_{ie} \times \dot{\boldsymbol{b}}^i_e + \ddot{\boldsymbol{b}}^i_{ei}.\end{aligned} \tag{4.24}$$

The last identity makes use of the fact that the angular velocity $\boldsymbol{\omega}^i_{ie}$ is constant, i.e. $\dot{\boldsymbol{\omega}}^i_{ie} = \boldsymbol{0}$. In many applications, the n-frame is defined stationary w.r.t. the e-frame. That is,

$$\boldsymbol{b}^e = R^{en}\boldsymbol{b}^n + \boldsymbol{n}^e, \tag{4.25}$$

where $\boldsymbol{n}^e$ and R^{en} are constant. Differentiation w.r.t. time gives $\dot{\boldsymbol{b}}^e_e = \dot{\boldsymbol{b}}^e_n$. Hence,

$$\dot{\boldsymbol{b}}^i_e = R^{ie}\dot{\boldsymbol{b}}^e_e = R^{ie}\dot{\boldsymbol{b}}^e_n = \dot{\boldsymbol{b}}^i_n, \tag{4.26}$$

and $\ddot{\boldsymbol{b}}^i_{ei}$ as can be expanded using (4.21) as

$$\ddot{\boldsymbol{b}}^i_{ei} = \tfrac{\mathrm{d}}{\mathrm{d}t}\dot{\boldsymbol{b}}^i_e = \tfrac{\mathrm{d}}{\mathrm{d}t}\dot{\boldsymbol{b}}^i_n = \tfrac{\mathrm{d}}{\mathrm{d}t}(R^{in}\dot{\boldsymbol{b}}^n_n) = \boldsymbol{\omega}^i_{ie} \times \dot{\boldsymbol{b}}^i_n + \ddot{\boldsymbol{b}}^i_{nn}. \tag{4.27}$$

Combining (4.24), (4.26) and (4.27) and rotating the result, we obtain

$$\ddot{\boldsymbol{b}}^n_{ii} = \boldsymbol{\omega}^n_{ie} \times \boldsymbol{\omega}^n_{ie} \times R^{ni}\boldsymbol{b}^i + 2\boldsymbol{\omega}^n_{ie} \times \dot{\boldsymbol{b}}^n_n + \ddot{\boldsymbol{b}}^n_{nn}, \tag{4.28}$$

where $\boldsymbol{\omega}^n_{ie} \times \boldsymbol{\omega}^n_{ie} \times R^{ni}\boldsymbol{b}^i$ is known as the centrifugal acceleration and $2\boldsymbol{\omega}^n_{ie} \times \dot{\boldsymbol{b}}^n_n$ as the Coriolis acceleration. This expression transforms $\ddot{\boldsymbol{b}}^n_{ii}$ to $\ddot{\boldsymbol{b}}^n_{nn}$. The latter can be integrated to obtain velocity $\dot{\boldsymbol{b}}^n_n$ and position $\boldsymbol{b}^n$:

$$\begin{cases} \tfrac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{b}^n = \dot{\boldsymbol{b}}^n_n, \\ \tfrac{\mathrm{d}}{\mathrm{d}t}\dot{\boldsymbol{b}}^n_n = \ddot{\boldsymbol{b}}^n_{nn} \end{cases} \quad \Rightarrow \quad \begin{cases} \boldsymbol{b}^n_{t+1} = \boldsymbol{b}^n_t + T\dot{\boldsymbol{b}}^n_{n,t} + \frac{T^2}{2}\ddot{\boldsymbol{b}}^n_{nn,t}, \\ \dot{\boldsymbol{b}}^n_{n,t+1} = \dot{\boldsymbol{b}}^n_{n,t} + T\ddot{\boldsymbol{b}}^n_{nn,t} \end{cases} \tag{4.29}$$

The integration has been performed under the assumption that the acceleration $\ddot{\boldsymbol{b}}^n_{nn,\tau}$ is constant in the interval $\tau \in (t, t+T)$.

In the context of navigation over large distances it is customary to move and rotate the n-frame along the surface of the earth. In that case, (4.25) is not satisfied with constant $\boldsymbol{n}^e$ and R^{en}, so (4.26)–(4.29) are no longer valid. Instead, $\ddot{\boldsymbol{b}}^i_{ei}$ can be expanded using (4.21) as

$$\ddot{\boldsymbol{b}}^i_{ei} = \tfrac{\mathrm{d}}{\mathrm{d}t}\dot{\boldsymbol{b}}^i_e = \tfrac{\mathrm{d}}{\mathrm{d}t}(R^{in}\dot{\boldsymbol{b}}^n_e) = \boldsymbol{\omega}^i_{in} \times \dot{\boldsymbol{b}}^i_e + \ddot{\boldsymbol{b}}^i_{en}. \tag{4.30}$$

Substituting this expression in (4.24) and rotating the result, we obtain

$$\ddot{\boldsymbol{b}}^n_{ii} = \boldsymbol{\omega}^n_{ie} \times \boldsymbol{\omega}^n_{ie} \times R^{ni}\boldsymbol{b}^i + (2\boldsymbol{\omega}^n_{ie} + \boldsymbol{\omega}^n_{en}) \times \dot{\boldsymbol{b}}^n_e + \ddot{\boldsymbol{b}}^n_{en}, \tag{4.31}$$

since $\boldsymbol{\omega}^n_{in} = \boldsymbol{\omega}^n_{ie} + \boldsymbol{\omega}^n_{en}$. This expression transforms $\ddot{\boldsymbol{b}}^n_{ii}$ to $\ddot{\boldsymbol{b}}^n_{en}$. The latter can be integrated to obtain the ground speed $\dot{\boldsymbol{b}}^n_e$.

The orientation between the b-frame and the i-frame can be decomposed as

$$q^{ib} = q^{ie} \odot q^{en} \odot q^{nb}. \tag{4.32}$$

Differentiating this expression w.r.t. time using (4.17) and (4.19) gives

$$\begin{aligned} q^{ib} \odot \tfrac{1}{2}\omega^b_{ib} &= q^{ie} \odot \tfrac{1}{2}\omega^e_{ie} \odot q^{en} \odot q^{nb} + q^{ie} \odot q^{en} \odot \tfrac{1}{2}\omega^n_{en} \odot q^{nb} \\ &\quad + q^{ie} \odot q^{en} \odot q^{nb} \odot \tfrac{1}{2}\omega^n_{nb} \\ &= q^{ib} \odot q^{be} \odot \tfrac{1}{2}\omega^e_{ie} \odot q^{eb} + q^{ib} \odot q^{bn} \odot \tfrac{1}{2}\omega^b_{en} \odot q^{nb} \\ &\quad + q^{ib} \odot \tfrac{1}{2}\omega^n_{nb} \\ &= q^{ib} \odot \tfrac{1}{2}(\omega^b_{ie} + \omega^b_{en} + \omega^b_{nb}) \end{aligned} \tag{4.33}$$

Comparing terms gives

$$\boldsymbol{\omega}^b_{ib} = \boldsymbol{\omega}^b_{ie} + \boldsymbol{\omega}^b_{en} + \boldsymbol{\omega}^b_{nb}. \tag{4.34}$$

This expression transforms ω_{ib}^{b} to ω_{nb}^{b}. The latter can be integrated to obtain the orientation:

$$\frac{\mathrm{d}}{\mathrm{d}t} q^{nb} = q^{nb} \odot \tfrac{1}{2}\omega_{nb}^{b} \quad \Rightarrow \quad q_{t+1}^{nb} = q_{t}^{nb} \odot \exp(\tfrac{T}{2}\omega_{nb,t}^{b}) \tag{4.35}$$

The integration has been performed under the assumption that the angular velocity $\omega_{nb,\tau}^{b}$ is constant in the interval $\tau \in (t, t+T)$. Details on the integration of quaternions are provided in Appendix B.3.

4.4 Coordinate frame alignment

A further topic of kinematics is the alignment of coordinate frames, that is, finding the relative position and orientation between two coordinate frames. This problem has received a lot of attention in the field of robotics under the name of hand-eye calibration (Tsai and Lenz, 1989; Daniilidis, 1999; Strobl and Hirzinger, 2006). In this thesis, the problem of aligning coordinate frames occurs for instance in the context of comparing filter performance with an external optical reference system such as Vicon, see Figure 4.4. In this section we discuss two cases which will be used in subsequent chapters.

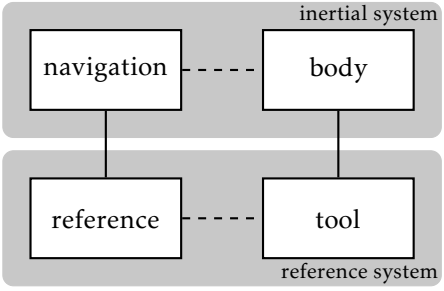


Figure 4.4: *A reference system introduces additional coordinate frames. Solid lines indicate rigid connections which have to be calibrated, dotted lines are non-rigid, time varying relations.*

First, consider the case of determining the relative orientation from measuring a vector quantity in two coordinate frames, here denoted the u-frame and the v-frame. This gives rise to the following *maximum likelihood* (ML) problem, see Section 2.1,

$$\begin{aligned} \max_{\theta} \quad & \sum_{n=1}^{N} p(\boldsymbol{e}_n^u) + p(\boldsymbol{e}_n^v) \\ \text{s.t.} \quad & \boldsymbol{y}_n^u = \boldsymbol{v}_n^u + \boldsymbol{e}_n^u, \qquad n = 1, \ldots, N \\ & \boldsymbol{y}_n^v = R^{vu}\boldsymbol{v}_n^u + \boldsymbol{e}_n^v, \quad n = 1, \ldots, N \end{aligned} \tag{4.36}$$

with parameter vector $\theta = \{R^{vu}, \{\boldsymbol{v}_n^u, \boldsymbol{e}_n^u, \boldsymbol{e}_n^v\}_{n=1}^{N}\}$. Dependent on the measurement noise, solving the general problem (4.36) can be very hard. In case of

$\boldsymbol{e}_n^u \sim \mathcal{N}(0, \Sigma_u)$ and $\boldsymbol{e}_n^v \sim \mathcal{N}(0, \Sigma_v)$ with $\Sigma_u = \Sigma_v = \sigma^2 I_3$, we can introduce a combined noise $\boldsymbol{e}_n = \boldsymbol{y}_n^u - R^{uv}\boldsymbol{y}_n^v = \boldsymbol{e}_n^u - R^{uv}\boldsymbol{e}_n^v \sim \mathcal{N}(0, 2\sigma^2 I_3)$. Under this assumption solving (4.36) is equivalent to solving

$$\begin{aligned} \min_{R^{uv}} \quad & \sum_{n=1}^{N} \|\boldsymbol{e}_n\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{e}_n = \boldsymbol{y}_n^u - R^{uv}\boldsymbol{y}_n^v, \quad n = 1, \ldots, N \end{aligned} \tag{4.37}$$

Parameterizing the orientation R^{uv} using the unit quaternion q^{uv}, the closed form solution to this problem is given by Theorem 4.1.

4.1 Theorem (Relative orientation from vector measurements). *Let $\{\boldsymbol{y}_n^u\}_{n=1}^N$ and $\{\boldsymbol{y}_n^v\}_{n=1}^N$ measurements satisfying $\boldsymbol{y}_n^u \approx q^{uv} \odot \boldsymbol{y}_n^v \odot q^{vu}$. Then the sum of the squared residuals,*

$$V(q^{uv}) = \sum_{n=1}^{N} \|\boldsymbol{e}_n\|_2^2 = \sum_{n=1}^{N} \|\boldsymbol{y}_n^u - q^{uv} \odot \boldsymbol{y}_n^v \odot q^{vu}\|_2^2, \tag{4.38}$$

is minimized by $\hat{q}^{uv} = \boldsymbol{v}_1$, where $\boldsymbol{v}_1$ is the eigenvector corresponding to the largest eigenvalue λ_1 of the system $A\boldsymbol{v} = \lambda\boldsymbol{v}$ with

$$A = -\sum_{n=1}^{N} (\boldsymbol{y}_n^u)^L (\boldsymbol{y}_n^v)^R. \tag{4.39}$$

Proof: Analogous to Horn (1987), the squared residuals can be written as

$$\|\boldsymbol{e}_n\|_2^2 = \|\boldsymbol{y}_n^u\|_2^2 - 2\boldsymbol{y}_n^u \cdot (q^{uv} \odot \boldsymbol{y}_n^v \odot q^{vu}) + \|\boldsymbol{y}_n^v\|_2^2.$$

Minimization only affects the middle term, which can be simplified to

$$\begin{aligned} \boldsymbol{y}_n^u \cdot (q^{uv} \odot \boldsymbol{y}_n^v \odot q^{vu}) &= -(\boldsymbol{y}_n^u \odot (q^{uv} \odot \boldsymbol{y}_n^v \odot q^{vu}))_0 \\ &= -(\boldsymbol{y}_n^u \odot q^{uv})^T (\boldsymbol{y}_n^v \odot q^{vu})^c \\ &= -(q^{uv})^T (\boldsymbol{y}_n^u)^L (\boldsymbol{y}_n^v)^R q^{uv}, \end{aligned}$$

using the relation $(a \odot b)_0 = a^T b^c$ for the scalar part of quaternion multiplication. The minimization problem can now be restated as

$$\underset{\|q^{uv}\|_2=1}{\arg\min} \sum_{n=1}^{N} \|\boldsymbol{e}_n\|_2^2 = \underset{\|q^{uv}\|_2=1}{\arg\max} \; (q^{uv})^T A q^{uv},$$

where A is defined in (4.39). Using (B.3),(B.19) and (B.20) we have

$$(\boldsymbol{y}_n^u)^L (\boldsymbol{y}_n^v)^R = [-(\boldsymbol{y}_n^u)^L]^T [-(\boldsymbol{y}_n^v)^R]^T = [(\boldsymbol{y}_n^v)^R (\boldsymbol{y}_n^u)^L]^T = [(\boldsymbol{y}_n^u)^L (\boldsymbol{y}_n^v)^R]^T,$$

from which can be concluded that A is a real symmetric matrix.

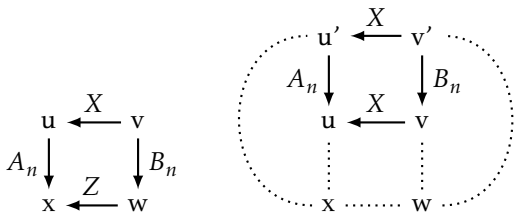
Let $q^{uv} = V\boldsymbol{\alpha}$ with $\|\boldsymbol{\alpha}\|_2 = 1$, where V is an orthonormal basis obtained from the

symmetric eigenvalue decomposition of $A = V\Lambda V^T$. Then,

$$(q^{uv})^T A q^{uv} = \boldsymbol{\alpha}^T V^T V \Lambda V^T V \boldsymbol{\alpha} = \sum_{i=1}^{4} \alpha_i^2 \lambda_i \leq \lambda_1,$$

where λ_1 is the largest eigenvalue of A. Equality is obtained for $\boldsymbol{\alpha} = (1, 0, 0, 0)^T$, that is, $\hat{q}^{uv} = \boldsymbol{v}_1$. □

A second alignment problem is to determine the relative pose from pairs of relative pose measurements. Two closely related scenarios, both known as hand-eye calibration, are described in robotics literature, see e.g. Tsai and Lenz (1989); Strobl and Hirzinger (2006). The classical hand-eye calibration scenario is to



(a) $AX = ZB$ ***(b)*** $AX = XB$

Figure 4.5: *Two versions of the hand-eye calibration problem.*

move a manipulator (hand) from u to u' and to observe this change with a sensor (eye) which moves from v to v', as illustrated by Figure 4.5b. Evaluating the two routes from v' to u yields the relation $AX = XB$, from which the unknown transformation X can be solved given a number of relative poses $\{A_n, B_n\}_{n=1}^N$. Instead of working with changes in manipulator pose and sensor pose, it is possible to use these poses directly, see Figure 4.5a. Evaluating the two routes from v to x in this slightly more general scenario yields the relation $AX = ZB$, from which the unknown transformations X and Z can be jointly solved given a number of poses $\{A_n, B_n\}_{n=1}^N$.

Expanding the $AX = ZB$ problem, we have that the measurements $\{q_n^{uv}, \boldsymbol{v}_n^u\}_{n=1}^N$ and $\{q_n^{xw}, \boldsymbol{w}_n^x\}_{n=1}^N$ are, in absence of noise, related by

$$\begin{cases} q_n^{uv} \odot q^{vw} = q^{ux} \odot q_n^{xw}, \\ \boldsymbol{v}_n^u + R_n^{uv} \boldsymbol{w}^v = \boldsymbol{x}^u + R^{ux} \boldsymbol{w}_n^x. \end{cases} \tag{4.40}$$

This is an inherently coupled and nonlinear problem in the unknowns $\{q^{ux}, \boldsymbol{x}^u\}$ and $\{q^{vw}, \boldsymbol{w}^v\}$. Many nonlinear optimization approaches have been formulated (Strobl and Hirzinger, 2006). However, assuming known orientations, the trans-

lational part is linear in the unknowns $\boldsymbol{x}^u$ and $\boldsymbol{w}^v$,

$$\boldsymbol{v}_n^u - R^{ux}\boldsymbol{w}_n^x = \begin{bmatrix} I_3 & -R_n^{uv} \end{bmatrix} \begin{pmatrix} \boldsymbol{x}^u \\ \boldsymbol{w}^v \end{pmatrix}. \tag{4.41}$$

Decoupling approaches provide a starting point for the optimization approaches and are frequently employed directly with satisfactory results. An explicit solution to the rotational part is given by Theorem 4.2.

4.2 Theorem (Relative orientation from orientation measurements). *Suppose* $\{q_n^{uv}\}_{n=1}^N$ *and* $\{q_n^{xw}\}_{n=1}^N$ *are measurements satisfying* $q_n^{uv} \odot q^{vw} \approx q^{ux} \odot q_n^{xw}$. *Then the residual rotation error,*

$$V(q^{ux}, q^{vw}) = \sum_{n=1}^{N} \|e_n\|_2^2 = \sum_{n=1}^{N} \|q_n^{uv} \odot q^{vw} \odot q_n^{wx} \odot q^{xu} - 1\|_2^2, \tag{4.42}$$

is minimized by $\hat{q}^{ux} = \boldsymbol{v}_1$ *and* $\hat{q}^{vw} = \boldsymbol{u}_1$, *where* $\boldsymbol{u}_1$ *and* $\boldsymbol{v}_1$ *are the first left and right singular vectors of the matrix* $A = U\Sigma V^T$, *with*

$$A = \sum_{n=1}^{N} [(q_n^{uv})^L]^T [(q_n^{xw})^R]. \tag{4.43}$$

Proof: The residual orientation error can be rewritten as

$$\begin{aligned} \|e_n\|_2^2 &= \|q_n^{uv} \odot q^{vw} \odot q_n^{wx} \odot q^{xu} - 1\|_2^2 \\ &= (q_n^{uv} \odot q^{vw} \odot q_n^{wx} \odot q^{xu} - 1)(q_n^{uv} \odot q^{vw} \odot q_n^{wx} \odot q^{xu} - 1)^c \\ &= 2 - (q_n^{uv} \odot q^{vw} \odot q_n^{wx} \odot q^{xu}) - (q_n^{uv} \odot q^{vw} \odot q_n^{wx} \odot q^{xu})^c \end{aligned}$$

Using the quaternion properties, $q + q^c = 2q_0$ and $(a \odot b)_0 = a^T b^c$, the above expression can be simplified to

$$\|e_n\|_2^2 = 2 - 2(q_n^{uv} \odot q^{vw})^T (q_n^{wx} \odot q^{xu})^c = 2 - 2(q^{vw})^T [(q_n^{uv})^L]^T [(q_n^{xw})^R] q^{ux}.$$

The minimization problem can now be restated as

$$\arg\min \sum_{n=1}^{N} \|e_n\|_2^2 = \arg\max (q^{vw})^T A q^{ux},$$

subject to $\|q^{ux}\|_2 = 1$, $\|q^{vw}\|_2 = 1$ and where A is defined in (4.43).

Let $q^{vw} = U\boldsymbol{\alpha}$ and $q^{ux} = V\boldsymbol{\beta}$ with $\|\boldsymbol{\alpha}\|_2 = 1$ and $\|\boldsymbol{\beta}\|_2 = 1$, where U and V are orthonormal bases obtained from the singular value decomposition of $A = U\Sigma V^T$. Then,

$$(q^{vw})^T A q^{ux} = \boldsymbol{\alpha}^T U^T U \Sigma V^T V \boldsymbol{\beta} = \sum_{i=1}^{4} \alpha_i \sigma_i \beta_i \leq \sigma_1,$$

where σ_1 is the largest singular value. To obtain equality, choose both $\boldsymbol{\alpha} = \boldsymbol{\beta} = (1, 0, 0, 0)^T$, that is, $\hat{q}^{ux} = \boldsymbol{v}_1$ and $\hat{\boldsymbol{q}}^{vw} = \boldsymbol{u}_1$. □

Algorithm 4.1 Reference system calibration (pose)

1. Acquire pose measurements $\{q_k^{uv}, \boldsymbol{v}_k^u\}_{k=1}^K$ and $\{q_k^{xw}, \boldsymbol{w}_k^x\}_{k=1}^K$ from the sensor unit and the reference system respectively.
2. Compute an estimate $\hat{q}^{ux}, \hat{q}^{vw}$ from $\{q_k^{uv}\}$ and $\{q_k^{xw}\}$ using Theorem 4.2.
3. Obtain a least squares estimate $\hat{\boldsymbol{x}}^u, \hat{\boldsymbol{w}}^v$ from $\{\boldsymbol{v}_k^u\}$ and $\{\boldsymbol{w}_k^x\}$ using (4.41).

Algorithm 4.2 Reference system calibration (inertial)

1. Capture inertial measurements $\{\boldsymbol{y}_{\omega,k}\}_{k=1}^K, \{\boldsymbol{y}_{\mathrm{a},k}\}_{k=1}^K$ and $\{\boldsymbol{z}_{\omega,k}\}_{k=1}^K, \{\boldsymbol{z}_{\mathrm{a},k}\}_{k=1}^K$ from the sensor unit and the reference system respectively. Rotate around all 3 axes, with sufficiently exiting angular velocities.
2. Compute an estimate $\hat{q}^{bt}$ from $\{\boldsymbol{y}_{\omega,k} = \boldsymbol{\omega}_k^b\}$ and $\{\boldsymbol{z}_{\omega,k} = \boldsymbol{\omega}_k^t\}$ using Theorem 4.1.
3. Compute a least squares estimate $\boldsymbol{b}^t$ from $\{\boldsymbol{y}_{\mathrm{a},k}\}$ and $\{\boldsymbol{z}_{\mathrm{a},k}\}$ using (4.45).

Notice that $q_n^{uv} \odot q^{vw} \odot q_n^{wx} \odot q^{xu} = \delta q_n$, where $\delta q_n = (\cos\frac{\alpha_n}{2}, \boldsymbol{n}_n^u \sin\frac{\alpha_n}{2}) = (1, \boldsymbol{0})$ in absence of errors. With this notation, the cost function (4.42) can be interpreted as

$$\begin{aligned} V(q^{ux}, q^{vw}) &= \sum_{n=1}^N \|\delta q_n - 1\|_2^2, = \sum_{n=1}^N \|(\cos\tfrac{\alpha_n}{2} - 1, \boldsymbol{n}_n^u \sin\tfrac{\alpha_n}{2})\|_2^2 \\ &= \sum_{n=1}^N (\cos\tfrac{\alpha_n}{2} - 1)^2 + (\sin\tfrac{\alpha_n}{2})^2 = 2\sum_{n=1}^N (1 - \cos\tfrac{\alpha_n}{2}) \approx \frac{1}{4}\sum_{n=1}^N \alpha_n^2. \end{aligned}$$

That is, Theorem 4.2 minimizes the error angles.

Part II contains several inertial tracking algorithms whose performance is evaluated by a comparison with a reference system. This implies that the associated coordinate frames have to be aligned. Dependent of the type of reference system, different alignment methods have to be used. In case of a reference system providing pose measurement, e.g., an industrial robot or a Vicon system, Algorithm 4.1 can be used to align the systems.

A second type of reference system is a high grade inertial navigation system. In this case, Algorithm 4.2 can be used to align the systems. It uses the gyroscope and accelerometer measurements of both systems. Since they are rigidly connect, that is, $\boldsymbol{b}_k^n = \boldsymbol{t}_k^n + R_k^{nt}\boldsymbol{b}^t$, we have

$$\begin{aligned} \ddot{\boldsymbol{b}}_{ii,k}^n &= \ddot{\boldsymbol{t}}_{ii,k}^n + \dot{\boldsymbol{\omega}}_{nt,k}^n \times R_k^{nt}\boldsymbol{b}^t + \boldsymbol{\omega}_{nt,k}^n \times \boldsymbol{\omega}_{nt,k}^n \times R_k^{nt}\boldsymbol{b}^t \\ &= \ddot{\boldsymbol{t}}_{ii,k}^n + R_k^{nt}\left([\dot{\boldsymbol{\omega}}_{nt,k}^t\times] + [\boldsymbol{\omega}_{nt,k}^t\times]^2\right)\boldsymbol{b}^t. \end{aligned} \tag{4.44}$$

Hence, a pair of accelerometer measurements $\boldsymbol{y}_{\mathrm{a},k}$ and $\boldsymbol{z}_{\mathrm{a},k}$ can be combined as

$$R_k^{tb}\boldsymbol{y}_{\mathrm{a},k} - \boldsymbol{z}_{\mathrm{a},k} = R_k^{tn}\left(\ddot{\boldsymbol{b}}_{ii,k}^n - \ddot{\boldsymbol{t}}_{ii,k}^n\right) = \left([\dot{\boldsymbol{\omega}}_{nt,k}^t\times] + [\boldsymbol{\omega}_{nt,k}^t\times]^2\right)\boldsymbol{b}^t. \tag{4.45}$$

Part II

Sensor fusion applications

5

Inertial and magnetic sensors

Standalone *inertial measurement units* (IMUs) are an excellent source of orientation information. This chapter discusses the sensor fusion behind this application together with the associated problem of magnetic field calibration.

5.1 Problem formulation

Traditionally standalone inertial sensors have been used in aviation, aerospace and shipping applications. The employed strapdown navigation solutions (see Section 3.1.3) require accurate and therefore bulky and expensive sensors. With the relatively recent development of *micro-machined electromechanical system* (MEMS) technology, inertial sensing components have become very small and inexpensive, at the cost of reduced accuracy. These (ongoing) developments have enabled many new application areas for inertial sensors. In this chapter we focus on orientation estimation and related subjects such as multi-segment systems and magnetic field calibration.

IMUs are nowadays available as chipsets and as result they have been integrated in many devices, including mobile phones. They are also available as very small, unobtrusive and wireless devices, see Figure 5.1. This allows them to be used in a wide range of applications, including putting them on small vehicles and wearing them on the body. The latter allows measuring the movement of a person or animal. Motion capture is of interest in for instance clinical applications where the clinical specialist or physiotherapist is able to analyze and monitor the movement of a patient. Another application area is in the movie and gaming industry where the movements of an actor form the basis for special effects or game characters.

Figure 5.1: *An Xsens MTw, a wireless IMU, is placed in an easy-to-wear strap.*

In above applications, an IMU is used for orientation estimation. In case position estimates are required an additional source of information has to be used, see Chapter 6–8. The orientation estimates are obtained by sensor fusion of measurements from the integrated gyroscopes, accelerometers and magnetometers. In the context of body worn sensors, the joints between body segments provide an additional source of information which can be exploited.

When using the magnetometers for orientation estimation, care should be exerted that these measure the undistorted earth magnetic field. In case the IMU is mounted on a rigid magnetic object, such as a car or a prosthesis this introduces an orientation-dependent magnetic distortion which has to be calibrated for.

The remainder of this chapter is organized as follows: sensor fusion for orientation estimation using standalone inertial sensors is discussed in Section 5.2, Section 5.3 discusses sensor fusion for multiple segments connected with joints and Section 5.4 discusses how to calibrate local magnetic distortions.

5.2 Orientation estimation

The basic idea of orientation estimation of an IMU is sensor fusion of its gyroscopes, accelerometers and magnetometers. All these sensors contained in an IMU provide information about its orientation. The angular velocity, measured by the gyroscopes, can be integrated to obtain orientation. The integration drift caused by noise and slow time-varying biases can be compensated using the magnetometers and accelerometers.

The recent survey by Crassidis et al. (2007) gives a thorough overview of orientation estimation methods, where the *extended Kalman filter* (EKF) turns out to be the sensor fusion method of choice for many applications. The models are

derived using strapdown navigation theory (Titterton and Weston, 1997; Savage, 1998). A closely related topic is that of pedestrian navigation (Skog et al., 2010; Skog, 2009; Foxlin, 2005), where zero velocity updates are used to stabilize the integration drift.

As discussed in Chapter 2, state-space models are one of the key ingredients of sensor fusion. For the sensor unit introduced earlier in this section, such a model can be constructed from the models in Chapter 3 and Chapter 4. They are repeated below for the reader's convenience.

Assuming zero acceleration of the sensor, the accelerometer measurement model (3.3) becomes

$$\boldsymbol{y}_{\mathrm{a}} = -R^{bn}\boldsymbol{g}^{n} + \boldsymbol{\delta}_{\mathrm{a}}^{b} + \boldsymbol{e}_{\mathrm{a}}^{b}, \tag{5.1}$$

where $\boldsymbol{g}$ is the known gravity vector, $\boldsymbol{\delta}_{\mathrm{a}}$ is a slowly time-varying bias and $\boldsymbol{e}_{\mathrm{a}}$ is *independently and identically distributed* (i.i.d.) Gaussian noise. The zero acceleration assumption is likely to be violated occasionally, since the IMU will be accelerating during motion. However, without additional sensors such as vision, *ultra-wideband* (UWB) or *global positioning system* (GPS), the acceleration is not known and cannot be compensated for. For the purpose of orientation estimation on humans, the accelerometer bias is negligible in comparison to the gravity vector.

Assuming a magnetically undistorted environment, the magnetometer measurement model (3.4) becomes

$$\boldsymbol{y}_{\mathrm{m},t} = R_{t}^{bn}\boldsymbol{m}_{\mathrm{e}}^{n} + \boldsymbol{e}_{\mathrm{m},t}^{b}, \tag{5.2}$$

where $\boldsymbol{m}_{\mathrm{e}}$ is the earth magnetic field vector which gives the direction of the magnetic north and $\boldsymbol{e}_{\mathrm{m}}$ is i.i.d. Gaussian noise. Ferromagnetic objects are commonly present, especially in indoor environments. Examples include furniture such desks, chairs and cabinets, but also structural elements like pillars and radiators. Therefore, care has to be exerted to ensure that the magnetic field is not distorted.

Following Section 4.3, we model orientation of the IMU using the gyroscope measurements,

$$q_{t+1}^{nb} = q_{t}^{nb} \odot \exp(\tfrac{T}{2}\boldsymbol{\omega}_{nb,t}^{b}), \tag{5.3}$$

where q^{nb} is a unit quaternion describing the orientation of the navigation frame (n-frame) relative to the body frame (b-frame) and T denotes the sampling interval. The angular velocity $\boldsymbol{\omega}_{nb}^{b}$ is calculated from the gyroscope measurements $\boldsymbol{y}_{\omega}$ using (3.2)

$$\boldsymbol{\omega}_{nb,t}^{b} = \boldsymbol{y}_{\omega,t} - R^{bn}\boldsymbol{\omega}_{ie}^{n} - \boldsymbol{\delta}_{\omega,t}^{b} - \boldsymbol{e}_{\omega,t}^{b}. \tag{5.4}$$

The gyroscope bias $\boldsymbol{\delta}_{\omega}^{b}$ is slowly time-varying. Hence, it is modeled as a random

walk,

$$\delta^b_{\omega,t+1} = \delta^b_{\omega,t} + e^b_{\delta_\omega,t}, \tag{5.5}$$

where $e^b_{\delta_\omega}$ is i.i.d. Gaussian noise.

Combining (5.1)–(5.5) we obtain a discrete-time nonlinear state-space model. Its state vector is, with slight abuse of notation, given by

$$x = \left(q^{nb}, \delta^b_\omega, \right)^T. \tag{5.6}$$

It can be used in an EKF to fuse the inertial measurements, see Section 2.2. The difficulty with this approach is that the assumptions in (5.1) and (5.2) are frequently violated: the IMU will be often accelerating and magnetic distortions are very common, especially in indoor environments. Example 5.1 shows what happens when relying on invalid assumptions.

5.1 Example: Orientation errors due to magnetic distortions

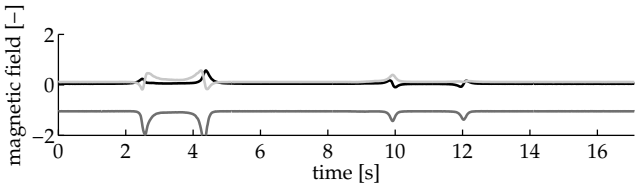
Consider a stationary IMU in a magnetically disturbed environment, in this example due to a moving pair of scissors. The changing magnetic field obviously does not match with the assumption of (5.2). When this situation is not detected and the measurements are used for orientation estimation, estimation errors are introduced, see Figure 5.2. Besides the orientation, the gyroscope biases are also affected and it takes some time before these are recovered. When accounting for the distorted measurements, the orientation estimates remain accurate.

The key solution to a robust and accurate orientation estimate is to rely mainly on the gyroscopes and selectively apply measurement updates using the accelerometers and magnetometers. Detecting when to apply these updates is a topic by itself and falls outside the scope of this thesis. One of the main ideas is to use hypothesis testing on the sensor measurements, see for instance Skog et al. (2010).

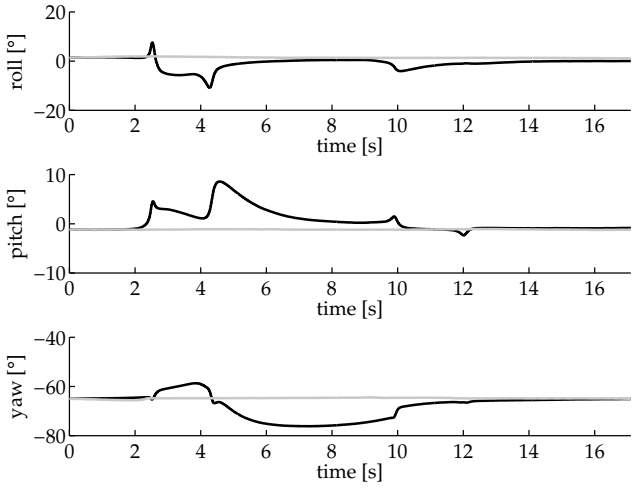
5.3 Multi-segment system

In clinical applications, motion capture is used to asses and monitor the movement of patients. It provides a valuable tool for medical specialist with applications in for instance diagnosis of certain conditions, pre- and post-surgery comparisons and monitoring of revalidation. Inertial motion capture technology can bring these analyzes out of specialized labs and into the practice of a specialist or the natural environment of the patient.

The basic idea of inertial motion capture is to attach IMUs to several body segments. For instance, consider an IMU on a foot, one on the lower leg and one on the upper leg. Since these body segments are connected to each other with a joint, in this case the ankle and the knee, they form a multi-segment chain or so-called articulated body. If the orientation of each segment is known over time, this fully determines the motion of the leg. That is, one could use the approach



(a) *Measured signal*



(b) *Orientation estimate*

Figure 5.2: *Orientation estimates under magnetic disturbances. Shown are estimates with (–) and without (–) accounting for the magnetic disturbances.*

of Section 5.2 to estimate the orientation of every segment individually. However, because the segments are linked to one another, their motion is constrained, see Figure 5.3. This additional piece of information allows replacing the tricky

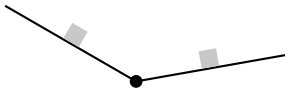


Figure 5.3: *A simple multi-segment model. Shown are two segments (–) connected with a joint (•). Each segment has an* IMU *(■) attached to it.*

assumptions of zero acceleration and an undistorted magnetic field and gives a very robust and self-contained system.

Multi-segment models are typically used in the field of bio-mechanical engineering (Schepers et al., 2007; Sabatini et al., 2005). Roetenberg and Luinge (2011) describe experimental results for a filter using these models. In this section, we use a similar approach to obtain a smoothing framework.

Although real-time visualizations are desirable, both medical specialists and researchers typically complete a measurement with a patient and analyze the data afterward in the form of a report with plots of relevant quantities. That implies that for clinical application, smoothing, see Section 2.2.2, is a logical approach. As discussed in Chapter 2, state-space models are one of the key ingredients of sensor fusion. For the multi-segment model introduced earlier in this section, such a model can be constructed from the models in Chapter 3 and Chapter 4. They are repeated below for the reader's convenience. Without loss of generality, we consider a system consisting of two segments. Larger systems are treated in exactly the same way.

Following Section 4.3, we model the position and orientation of each IMU using the inertial measurements,

$$\boldsymbol{b}^n_{k,t+1} = \boldsymbol{b}^n_{k,t} + T\dot{\boldsymbol{b}}^n_{k,t} + \tfrac{T^2}{2}\ddot{\boldsymbol{b}}^n_{k,t}, \tag{5.7a}$$

$$\dot{\boldsymbol{b}}^n_{k,t+1} = \dot{\boldsymbol{b}}^n_{k,t} + T\ddot{\boldsymbol{b}}^n_{k,t}, \tag{5.7b}$$

$$q^{nb}_{k,t+1} = q^{nb}_{k,t} \odot \exp(\tfrac{T}{2}\boldsymbol{\omega}^b_{nb,k,t}), \tag{5.7c}$$

where $\boldsymbol{b}^n_k$ and $\dot{\boldsymbol{b}}^n_k$ denote the position and velocity of the b-frame of sensor k resolved in the n-frame, q^{nb}_k is a unit quaternion describing the orientation of the n-frame relative to the b-frame of sensor k and T denotes the sampling interval. The acceleration $\ddot{\boldsymbol{b}}^n_k$ and angular velocity $\boldsymbol{\omega}^b_{nb,k}$ are calculated for each sensor from the corresponding accelerometer measurements $\boldsymbol{y}_{\mathrm{a},k}$ and gyroscope measurements $\boldsymbol{y}_{\omega,k}$ using (3.2) and (3.3),

$$\ddot{\boldsymbol{b}}^n_{k,t} = R^{nb}_{k,t}\left(\boldsymbol{y}_{\mathrm{a},k,t} - \boldsymbol{\delta}^b_{\mathrm{a},k,t} - \boldsymbol{e}^b_{\mathrm{a},k,t}\right) - 2\boldsymbol{\omega}^n_{ie} \times \dot{\boldsymbol{b}}^n_{k,t} + \boldsymbol{g}^n, \tag{5.8a}$$

$$\boldsymbol{\omega}^b_{nb,k,t} = \boldsymbol{y}_{\omega,k,t} - R^{bn}_{k,t}\boldsymbol{\omega}^n_{ie} - \boldsymbol{\delta}^b_{\omega,k,t} - \boldsymbol{e}^b_{\omega,k,t}. \tag{5.8b}$$

The inertial bias terms $\boldsymbol{\delta}^b_{\mathrm{a},k}$ and $\boldsymbol{\delta}^b_{\omega,k}$ of sensor k are slowly time-varying. Hence,

they are modeled as random walks,

$$\boldsymbol{\delta}^b_{\mathrm{a},k,t+1} = \boldsymbol{\delta}^b_{\mathrm{a},k,t} + \boldsymbol{e}^b_{\delta_\mathrm{a},k,t}, \tag{5.9a}$$

$$\boldsymbol{\delta}^b_{\omega,k,t+1} = \boldsymbol{\delta}^b_{\omega,k,t} + \boldsymbol{e}^b_{\delta_\omega,k,t}, \tag{5.9b}$$

where $\boldsymbol{e}^b_{\delta_\mathrm{a},k}$ and $\boldsymbol{e}^b_{\delta_\omega,k}$ are i.i.d. Gaussian noises.

The two segments are connected by a joint. This implies that the endpoints of the segments should coincide. This is modeled by using a joint measurement model,

$$\boldsymbol{y}_{\mathrm{j},t} \triangleq \boldsymbol{0} = \boldsymbol{j}^n - \boldsymbol{j}^n = \boldsymbol{b}^n_{1,t} + R^{nb}_{1,t}\boldsymbol{j}^b_1 - \boldsymbol{b}^n_{2,t} - R^{nb}_{2,t}\boldsymbol{j}^b_2 + \boldsymbol{e}_{\mathrm{j},t}, \tag{5.10}$$

where $\boldsymbol{j}$ it the position of the joint and $\boldsymbol{e}_\mathrm{j}$ is i.i.d. Gaussian noise modeling the flexibility of the joint.

The joint measurement model makes the relative orientation and position of the IMUs observable under very mild acceleration of the joint. In practice, any kind of human motion other than standing completely still, provides enough acceleration. However, for absolution position and orientation, additional sensors or assumptions, including those of Section 5.2, have to be included for at least one IMU. Alternatively, the unobservable modes of the state can be eliminated by a state transformation to relative parameters.

Combining (5.7)–(5.10) we obtain a discrete-time nonlinear state-space model. Its state vector is, with slight abuse of notation, given by

$$\boldsymbol{x} = \left(\{\boldsymbol{b}^n_k, \dot{\boldsymbol{b}}^n_k, q^{nb}_k, \boldsymbol{\delta}^b_{\mathrm{a},k}, \boldsymbol{\delta}^b_{\omega,k}\}_{k=1}^2\right)^T. \tag{5.11}$$

The model is used to solve the smoothing problem (2.30) to estimate $\boldsymbol{x}_{0:t}$, see Section 2.2.2. Using the state vector (5.11), the resulting optimization problem contains a large number of nonlinear equality constraints, $\|q^{nb}_k\|_2 = 1$, since the quaternions have to be unit-length. These constraints pose are very difficult to solve. A simple reparameterization of the problem eliminates these constraints completely. The idea is to parameterize the orientation using local Euclidean parameterization of the manifold (Grisetti et al., 2010; Crassidis et al., 2007). That is, define

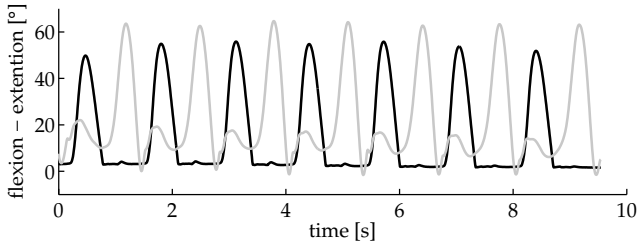
$$q^{nb}_{k,t} \triangleq \exp(\tfrac{1}{2}\boldsymbol{\phi}^n_{k,t}) \odot \bar{q}^{nb}_{k,t}, \tag{5.12}$$

where exp is the quaternion exponential, see Appendix B.3 and Section C.1. That is, the orientation q^{nb} is split in a constant nominal orientation $\bar{q}^{nb}$ and an orientation error $\boldsymbol{\phi}^n$. Using this parameterization the new state vector becomes

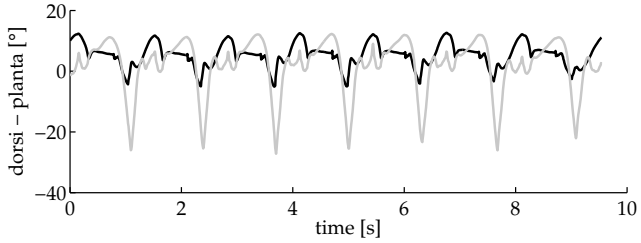
$$\tilde{\boldsymbol{x}} = \left(\{\boldsymbol{b}^n_k, \dot{\boldsymbol{b}}^n_k, \boldsymbol{\phi}^n_k, \boldsymbol{\delta}^b_{\mathrm{a},k}, \boldsymbol{\delta}^b_{\omega,k}\}_{k=1}^2\right)^T \tag{5.13}$$

and the smoothing problem takes the form of a nonlinear least squares problem, see Example 2.4, which can be solved using the methods described in Section 2.3. However, it is advantageous to update the nominal orientation trajectory after every iteration of the solver. This makes sure that the parameterization remains almost Euclidean and improves the convergence speed.

Tests on a filter using the multi-segment model show already very useful results. An example is given in Figure 5.4, which shows joint angles for a patient with a knee and ankle prosthesis during normal walking. The prosthesis allows only



(a) *Knee*



(b) *Ankle*

Figure 5.4: *Knee and angle joint angles of a patient during normal walking. Shown are the left, prosthetic side (–) and the right side (–).*

motion in the sagittal plane. Note the different walking pattern of the left, prosthetic side and the normal right side. A passive knee prosthesis is characterized by a stance phase with a constant joint angle, whereas a passive ankle prosthesis is characterized by a swing phase with an almost constant angle. Both effects are clearly visible in Figure 5.4a. More details, including a comparison to an optical reference system can be found in Garofalo (2010).

It remains future work to implement a solver for the smoothing approach and validate its performance. A very interesting extension would be the addition of force and torque sensors to the setup. This would enable complete dynamic evaluation including the calculation of joint loads.

5.4 Magnetic field calibration

In this section we describe a novel calibration algorithm to calibrate the magnetic field when the IMU is mounted close to a ferro-magnetic object.

5.4.1 Calibration algorithm

The key assumption when using magnetometers for orientation estimation is the presence of a homogeneous external magnetic field. The earth magnetic field is the most common external field, but inside buildings other conditions can occur. Ferromagnetic objects introduce local distortions to the magnetic field. These distortions violate the assumption of a homogeneous field and result in inaccurate orientation estimates when not properly accounted for.

When a calibrated IMU is mounted rigidly on or close to a rigid ferromagnetic object, the magnetometers no longer measure the external field directly (Vasconcelos et al., 2008; Renaudin et al., 2010). The object induces an orientation-dependent distortion to the field. This distortion can be decomposed in so-called soft and hard iron effects. The former effect is the result of the interaction of the object with the external magnetic field which changes the intensity as well as the direction of the field, whereas the latter stems from permanent magnetism and introduces an additive field. This gives rise to the following model for the magnetometer measurements, see (3.4),

$$\boldsymbol{y}_{\mathrm{m}} = \boldsymbol{m}^b + \boldsymbol{e}^b_{\mathrm{m}} = D\boldsymbol{m}^b_{\mathrm{e}} + \boldsymbol{d}^b + \boldsymbol{e}^b_{\mathrm{m}}, \tag{5.14}$$

where $\boldsymbol{m}$ is the local magnetic field vector, $\boldsymbol{m}_{\mathrm{e}}$ is the external magnetic field vector, D models the soft iron effect, $\boldsymbol{d}$ models the hard iron effect and $\boldsymbol{e}^b_{\mathrm{m}}$ is i.i.d. Gaussian noise. All vectors in (5.14) are resolved in the b-frame. On physical grounds mirroring effects can be excluded, i.e., $\det D > 0$. Furthermore, the external magnetic field is assumed to be homogeneous. That is,

$$\boldsymbol{m}^b_{\mathrm{e}} = R^{bn}\boldsymbol{m}^n_{\mathrm{e}}, \tag{5.15}$$

with $\boldsymbol{m}_{\mathrm{e}}$ constant in the n-frame. Without loss of generality we can define the positive x-direction of the n-frame to be aligned with the external magnetic field and scale the field to have unit-norm. That is, $\boldsymbol{m}^n_{\mathrm{e}} \in \mathcal{M}$, with $\mathcal{M}$ defined as

$$\mathcal{M} = \{\boldsymbol{x} \in \mathrm{S}(2) : \boldsymbol{x}_y = 0\}, \qquad \mathrm{S}(2) = \{\boldsymbol{x} \in \mathbb{R}^3 : \|\boldsymbol{x}\|_2 = 1\}, \tag{5.16}$$

where S(2) is the unit sphere.

The objective of magnetic field calibration is to estimate the distortion parameters D and $\boldsymbol{d}$. Several methods can be found in literature (e.g. Vasconcelos et al., 2008; Renaudin et al., 2010). However, these methods express their results in an arbitrary coordinate frame instead of the b-frame and an additional alignment problem needs to be solved before the magnetometers can be used. In this section we propose a novel calibration method which includes the alignment problem.

Our approach includes additional measurements of the vertical, which can be for instance obtained from the accelerometers when the IMU is stationary. The vertical measurements are modeled as

$$\boldsymbol{y}_{\mathrm{v}} = \boldsymbol{v}^b + \boldsymbol{e}_{\mathrm{v}} = R^{bn}\boldsymbol{v}^n + \boldsymbol{e}^b_{\mathrm{v}}, \tag{5.17}$$

where the vertical $\boldsymbol{v}^n = (0, 0, 1)^T$ and $\boldsymbol{e}_{\mathrm{v}}$ is i.i.d. Gaussian noise. Considering a set of K measurements $\{\boldsymbol{y}_{\mathrm{m},k}, \boldsymbol{y}_{\mathrm{v},k}\}_{k=1}^K$ of magnetometer and vertical measurements

taken while holding the IMU in different orientations, we formulate the calibration as an *maximum likelihood* (ML) problem, see Section 2.1, according to

$$\min_{\theta} \quad \frac{1}{2}\sum_{k=1}^{K} \|e^b_{\mathrm{m},k}\|^2_{\Sigma_{\mathrm{m}}^{-1}} + \|e^b_{\mathrm{v},k}\|^2_{\Sigma_{\mathrm{v}}^{-1}} \tag{5.18a}$$

$$\text{s.t.} \quad e^b_{\mathrm{m},k} = y_{\mathrm{m},k} - DR^{bn}_k m^n_{\mathrm{e}} - d^b, \quad k = 1,\dots,K \tag{5.18b}$$

$$e^b_{\mathrm{v},k} = y_{\mathrm{v},k} - R^{bn}_k v^n, \qquad k = 1,\dots,K \tag{5.18c}$$

$$R^{bn}_k \in \mathrm{SO}(3), \qquad k = 1,\dots,K \tag{5.18d}$$

$$m^n_{\mathrm{e}} \in \mathcal{M}, \tag{5.18e}$$

with variables $\theta = \{D, d^b, m^n_{\mathrm{e}}, \{R^{bn}_k\}_{k=1}^K\}$. The solvers described in Section 2.3 can efficiently solve (5.18), but require a reasonable initial estimate to converge to the correct optimum since the problem is non-convex.

To obtain an initial guess for θ, note that the external magnetic field m^n_{e} has unit length. Expanding this using (5.14) and (5.15) we obtain

$$\begin{aligned} 0 = \|m^n_{\mathrm{e}}\|_2^2 - 1 &= \|R^{nb}_k m^b_{\mathrm{e},k}\|_2^2 - 1 = \|m^b_{\mathrm{e},k}\|_2^2 - 1 \\ &\approx \|D^{-1}(y_{\mathrm{m},k} - d^b)\|_2^2 - 1 \\ &= y^T_{\mathrm{m},k} D^{-T} D^{-1} y_{\mathrm{m},k} - 2y^T_{\mathrm{m},k} D^{-T} D^{-1} d^b + (d^b)^T D^{-T} D^{-1} d^b - 1, \end{aligned} \tag{5.19}$$

where the magnetometer noise has been ignored. Introducing the notation

$$A = D^{-T} D^{-1}, \qquad b = -2D^{-T} D^{-1} d^b, \qquad c = (d^b)^T D^{-T} D^{-1} d^b - 1, \tag{5.20}$$

we can write (5.19) as

$$y^T_{\mathrm{m},k} A y_{\mathrm{m},k} + y^T_{\mathrm{m},k} b + c = 0, \tag{5.21}$$

which is the algebraic equation of an ellipsoid in y_{m}. It is a linear equation in the parameters A, b and c. Following Gander et al. (1994), we rearrange (5.21) as

$$Y_k \theta = 0, \tag{5.22a}$$

with Y_k and θ defined as

$$Y_k \triangleq \begin{bmatrix} y^T_{\mathrm{m},k} \otimes y^T_{\mathrm{m},k} & y^T_{\mathrm{m},k} & 1 \end{bmatrix}, \qquad \theta \triangleq \begin{bmatrix} \operatorname{vec} A \\ b \\ c \end{bmatrix} = 0, \tag{5.22b}$$

where $\otimes$ denotes the Kronecker product and vec denotes the vectorize operator (Magnus and Neudecker, 1999). Stacking (5.22) for a set of magnetometer measurements, a system $Y\theta = \mathbf{0}$ is obtained from which A, b and c can be obtained. Since A is symmetric, it has only six unknowns and 3 columns from Y can be eliminated. Using the singular value decomposition $Y = U\Sigma V^T = \sum_{i=1}^n u_i \sigma_i v_i^T$, a nontrivial solution is given by $\hat{\theta}_{\mathrm{s}} = v_n$ (Golub and Van Loan, 1996). This solution is determined up to scale, since $\hat{\theta} = \alpha\hat{\theta}_{\mathrm{s}}$, $\alpha \in \mathbb{R}$ also solves $Y\hat{\theta} = 0$. The

estimated ellipsoid parameters $\hat{A}$, $\hat{\boldsymbol{b}}$ and $\hat{c}$ are now calculated as

$$\begin{bmatrix} \operatorname{vec} \hat{A} \\ \hat{\boldsymbol{b}} \\ \hat{c} \end{bmatrix} = \alpha \begin{bmatrix} \operatorname{vec} \hat{A}_\mathrm{s} \\ \hat{\boldsymbol{b}}_\mathrm{s} \\ \hat{c}_\mathrm{s} \end{bmatrix} = \alpha \hat{\boldsymbol{\theta}}. \tag{5.23a}$$

Rearranging (5.20) as

$$1 = (\boldsymbol{d}^b)^T D^{-T} D^{-1} \boldsymbol{d}^b - c = \tfrac{1}{4} \boldsymbol{b}^T A^{-1} \boldsymbol{b} - c = \alpha \left(\tfrac{1}{4} \boldsymbol{b}_\mathrm{s}^T A_\mathrm{s}^{-1} \boldsymbol{b}_\mathrm{s} - c_\mathrm{s} \right),$$

the scale factor α is recovered as

$$\alpha = \left(\tfrac{1}{4} \hat{\boldsymbol{b}}_\mathrm{s}^T \hat{A}_\mathrm{s}^{-1} \hat{\boldsymbol{b}}_\mathrm{s} - \hat{c}_\mathrm{s} \right)^{-1}. \tag{5.23b}$$

Given $\hat{A}$ and $\hat{\boldsymbol{b}}$, initial estimates of the distortion parameters $\hat{D}_0$ and $\hat{\boldsymbol{d}}_0^b$ are obtained using (5.20) as

$$\hat{D}_0 \hat{D}_0^T = \hat{A}^{-1}, \qquad \hat{\boldsymbol{d}}_0^b = -\tfrac{1}{2} \hat{A}^{-1} \hat{\boldsymbol{b}}. \tag{5.24}$$

The factorization $\hat{D}_0 \hat{D}_0^T = \hat{A}^{-1}$ is not unique; if $\tilde{D}_0 \tilde{D}_0^T = \hat{A}^{-1}$ then $\hat{D}_0 = \tilde{D}_0 U$ with $UU^T = I_3$ also satisfies $\hat{D}_0 \hat{D}_0^T = \hat{A}^{-1}$. We determine $\tilde{D}_0$ using a Cholesky factorization, see Appendix A.3. The requirement $\det D > 0$ limits this unitary freedom to a rotational freedom; we have $\hat{D}_0 = \tilde{D}_0 R$ with $RR^T = I_3, \det R = 1$. That is, an initial magnetometer measurement model is given by

$$\boldsymbol{y}_\mathrm{m} \approx \tilde{D}_0 R \boldsymbol{m}_\mathrm{e}^b + \hat{\boldsymbol{d}}_0^b, \tag{5.25}$$

where $R \in \mathrm{SO}(3)$ is yet to be determined.

The inner product between the vertical $\boldsymbol{v}^n$ and the unknown external magnetic field $\boldsymbol{m}_\mathrm{e}^n$ can be expanded using (5.15), (5.17) and (5.25) as

$$\delta = \boldsymbol{v}^n \cdot \boldsymbol{m}_\mathrm{e}^n = R_k^{nb} \boldsymbol{v}_k^b \cdot R_k^{nb} \boldsymbol{m}_{\mathrm{e},k}^b = \boldsymbol{v}_k^b \cdot \boldsymbol{m}_{\mathrm{e},k}^b \approx \boldsymbol{y}_{\mathrm{v},k}^T R^T \tilde{D}_0^{-1} (\boldsymbol{y}_{\mathrm{m},k} - \hat{\boldsymbol{d}}_0^b). \tag{5.26}$$

The resulting system of equations is used in a ML fashion to formulate the problem

$$\min_{\boldsymbol{\theta}} \quad \frac{1}{2} \sum_{k=1}^{K} \|e_k\|_2^2 \tag{5.27a}$$

$$\text{s.t.} \quad e_k = \delta - \boldsymbol{y}_{\mathrm{v},k}^T R^T \tilde{D}_0^{-1} (\boldsymbol{y}_{\mathrm{m},k} - \hat{\boldsymbol{d}}_0^b), \quad k = 1, \ldots, K \tag{5.27b}$$

$$R \in \mathrm{SO}(3) \tag{5.27c}$$

with variables $\boldsymbol{\theta} = \{R, \delta\}$. Solving (5.27), starting in $\boldsymbol{\theta}_0 = \{I_3, -\sin \vartheta\}$, where ϑ is the inclination angle of the local earth magnetic field, gives initial estimates $\hat{R}_0$ and $\hat{\delta}_0$. With these, the initial measurement model (5.25) is fully specified. Furthermore, the first identity of (5.26) in combination with $\boldsymbol{m}_\mathrm{e}^n \in \mathcal{M}$ implies that an initial estimate of the external magnetic field is given by

$$\hat{\boldsymbol{m}}_{\mathrm{e},0}^n = \begin{pmatrix} \sqrt{1 - \hat{\delta}_0^2} & 0 & \hat{\delta}_0 \end{pmatrix}^T \tag{5.28}$$

Finally, using (5.15), (5.17) and (5.25), we have

$$\begin{bmatrix} \boldsymbol{v}^n \\ \hat{\boldsymbol{m}}^n_{\mathrm{e},0} \end{bmatrix} \approx R^{nb}_k \begin{bmatrix} \boldsymbol{y}_{\mathrm{v},k} \\ \hat{R}_0^T \tilde{D}_0^{-1}(\boldsymbol{y}_{\mathrm{m},k} - \hat{\boldsymbol{d}}^b_0) \end{bmatrix}, \quad k = 1, \ldots, K, \tag{5.29}$$

and we can obtain initial estimates $\{\hat{R}^{nb}_{k,0}\}_{k=1}^K$ by repeated application of Theorem 4.1.

Combining the steps discussed above, a starting point for (5.18) is given by

$$\boldsymbol{\theta}_0 = \left(\tilde{D}_0 \hat{R}_0, \hat{\boldsymbol{d}}^b_0, \hat{\boldsymbol{m}}^n_{\mathrm{e},0}, \{\hat{R}^{bn}_{k,0}\}_{k=1}^K\right)^T, \tag{5.30}$$

and the complete algorithm for calibrating the magnetic field is summarized in Algorithm 5.1.

Algorithm 5.1 Magnetic field calibration

1. Mount the IMU on the ferromagnetic device or object.
2. Position the assembly in a homogeneous magnetic environment.
3. Collect a dataset $\{\boldsymbol{y}_{\mathrm{m},k}, \boldsymbol{y}_{\mathrm{v},k}\}_{k=1}^N$ while rotating the assembly in all possible directions.
4. Perform an ellipse fit and obtain $\tilde{D}_0$ and $\hat{\boldsymbol{d}}^b_0$ using (5.22), (5.23) and (5.24).
5. Align the ellipse to the b-frame and obtain $\hat{R}_0$ and $\hat{\delta}_0$ by solving (5.27). The optimization is started in $\boldsymbol{\theta}_0 = \{I_3, -\sin\vartheta, \{0\}_{k=1}^K\}$, where ϑ is the inclination angle of the local earth magnetic field. Obtain $\hat{\boldsymbol{m}}^n_{\mathrm{e},0}$ using (5.28).
6. Determine the orientations of the IMU by repeated application of Theorem 4.1 on (5.29) and obtain $\{\hat{R}^{nb}_{k,0}\}_{k=1}^K$.
7. Solve the ML problem (5.18). The optimization is started in
$$\boldsymbol{\theta}_0 = \left(\tilde{D}_0 \hat{R}_0, \hat{\boldsymbol{d}}^b_0, \hat{\boldsymbol{m}}^n_{\mathrm{e},0}, \{\hat{R}^{bn}_{k,0}\}_{k=1}^K\right)^T,$$
using the results from Step 4–6.
8. Calculate the calibration accuracy using (2.39).

5.4.2 Experiments

Algorithm 5.1 has been used to calibrate the magnetic field of an IMU mounted in an Aero L-29 Delphin jet aircraft, see Figure 5.5. As the airplane is constructed from metal, the magnetometers of the IMU will suffer from magnetic distortion. Because of its sheer size and weight, the airplane cannot be easily rotated in all directions. Taxiing on the ground, only the heading can be changed, which does not provides enough excitation to determine a good magnetic field calibration. In flight, however, it is possible to tilt and bank the airplane to some extent. In this section we present results using a dataset recorded during a flight of approximately 35 min in which the airplane performed several maneuvers, including banked curves and barrel rolls in order to reach as many orientations as possible.

Figure 5.6 shows the magnetometer measurements before and after applying the distortion correction. The original magnetometer measurements are distorted and therefore lie on the surface of an ellipsoid, instead of on a sphere. After the

Figure 5.5: *The Aero L-29 Delphin jet used to test the magnetic field calibration.*

magnetic field calibration, the corrected magnetometer measurements do form a unit sphere so they can be used for sensor fusion. Note that the measurements are not evenly distributed on the sphere, due to the limited maneuverability of an airplane. Nevertheless it is possible to successfully perform a magnetic field calibration. The estimated distortion parameters with their 99% confidence intervals are shown below,

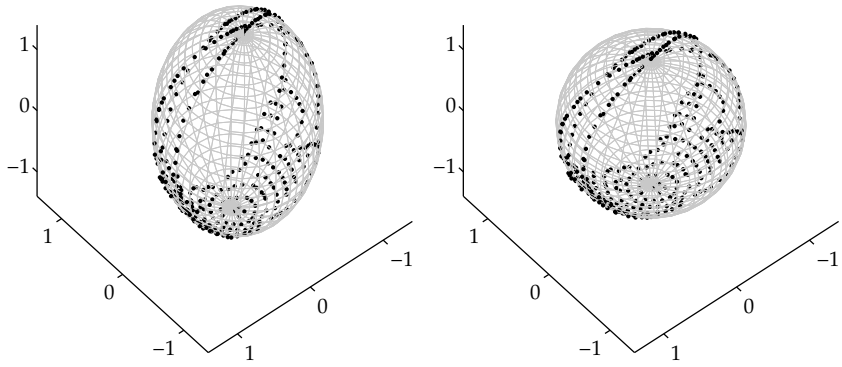
$$\hat{D} = \begin{bmatrix} 1.094 & 0.021 & -0.066 \\ 0.033 & 1.063 & 0.126 \\ -0.136 & -0.013 & 0.821 \end{bmatrix} \pm \begin{bmatrix} 0.018 & 0.014 & 0.009 \\ 0.016 & 0.003 & 0.006 \\ 0.012 & 0.007 & 0.003 \end{bmatrix}, \tag{5.31a}$$

$$\hat{\boldsymbol{d}} = \begin{pmatrix} 0.160 \\ 0.057 \\ -0.110 \end{pmatrix} \pm \begin{pmatrix} 0.006 \\ 0.003 \\ 0.003 \end{pmatrix}. \tag{5.31b}$$

These parameters clearly deviate from I_3 and $\mathbf{0}_{3\times1}$. Without compensation, the magnetometers will introduce large errors, which basically render them useless.

Figure 5.7 shows the normalized residuals of the magnetometer measurements in (5.18). The observed residuals are small and centered around zero, implying that a correct ML estimate has been obtained. The histogram has a smaller peak than expected from the theoretical distribution. This is most likely caused by the freedom the algorithm has in choosing the orientations $\{\hat{R}_k^{nb}\}_{k=1}^K$. In particular, the heading is only determined by a single magnetometer measurement, which implies that the magnetometer residual can be made very small in that direction.

The performance increase when using magnetometers in combination with magnetic field calibration of the sensor fusion accuracy is demonstrated in Chapter 8 where we will return to this experiment.



(a) *Original* ***(b)*** *Calibrated*

Figure 5.6: *Magnetometer measurements. Shown are the magnetometer measurements (·) together with the ellipse or sphere on which they lie (—).*

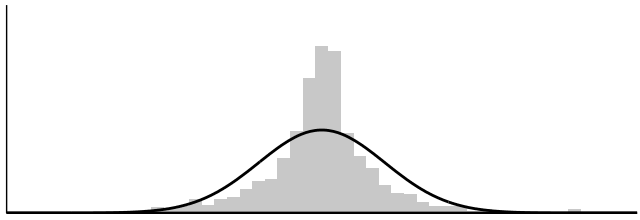


Figure 5.7: *Normalized residuals of the magnetometer measurements after calibration. Both the empirical distribution (bar) and the theoretical distribution (line) are shown.*

6

Inertial and vision

The combination of *inertial measurement units* (IMUs) and vision is very suitable for pose estimation, especially for applications in robotics and *augmented reality* (AR). This chapter discusses sensor fusion using this combination of sensors, together with the associated problem of relative pose calibration.

6.1 Problem formulation

The combination of vision and inertial sensors is very suitable for a wide range of robotics applications and a solid introduction to the technology is provided by Corke et al. (2007). The high-dynamic motion measurements of the IMU are used to support the vision algorithms by providing accurate predictions where features can be expected in the upcoming frame. This facilitates development of robust real-time pose estimation and feature detection / association algorithms, which are the cornerstones in many applications, including *simultaneous localization and mapping* (SLAM) (Bailey and Durrant-Whyte, 2006; Durrant-Whyte and Bailey, 2006) and AR (Bleser and Stricker, 2008; Chandaria et al., 2007b).

One of the main ideas of AR is to overlay a real scene with computer generated graphics in real-time. This can be accomplished by showing the virtual objects on see-through head-mounted displays or superimposing them on live video imagery. Figure 6.1 illustrates the concept of AR with some examples. In order to have realistic augmentation it is essential to know the position and orientation of the camera with high accuracy and low latency. This knowledge is required to position and align the virtual objects correctly on top of the real world and to ensure that they appear to stay in the same location regardless of the camera movement.

(a) Sport coverage using virtual annotations.

(b) Maintenance assistance.

(c) Visualization of virtual objects in TV shows.

(d) Virtual reconstruction.

Figure 6.1: *Examples of* AR *applications. By courtesy of BBC R&D and Fraunhofer IGD.*

Besides a good sensor fusion method to combine inertial and vision measurements, the problem of estimating the relative translation and orientation between a camera and an IMU needs to be addressed to successfully apply the proposed tracking technology in real applications. Accurate knowledge of the quantities is an important enabler for high-quality and robust sensor fusion.

The remainder of this chapter is organized as follows: the sensor fusion method for pose estimation using inertial and vision measurements is discussed in Section 6.2 and the relative pose calibration is discussed in Section 6.3. Earlier versions of the work in this chapter have been published in Hol et al. (2007, 2010b).

6.2 Pose estimation

In this section we discuss a 6 *degrees of freedom* (DOF) tracking system, estimating both position and orientation using sensor fusion of vision and inertial sensors. The material has been derived within the context of AR. However, the approaches and results are also applicable to other applications areas including SLAM.

Figure 6.2: *An Xsens prototype sensor unit, integrating an IMU and a camera into a single housing.*

6.2.1 Sensor fusion

Our setup is based on a prototype sensor unit, shown in Figure 6.2. It integrates an IMU and a camera in a single unit. The devices are synchronized at hardware level and have a common clock, which significantly simplifies the signal processing.

The combination of inertial and vision sensors has previously been used in literature, see e.g. Corke et al. (2007) for an overview. Reported systems apply various methods: inertial measurements are used as backup (Aron et al., 2007), for short-time pose prediction (Klein and Drummond, 2004), or depth map alignment (Lobo and Dias, 2004). Alternatively, vision and inertial subsystems are loosely coupled, using visual pose measurements (Ribo et al., 2004; Chroust and Vincze, 2004; Armesto et al., 2007). More recently, the combination of vision and inertial sensors has been used in the context of SLAM (Gemeiner et al., 2007; Sjanic et al., 2011) as well as calibration (Mirzaei and Roumeliotis, 2008; Kelly and Sukhatme, 2011; Jones and Soatto, 2011).

Given the setting of AR, the basic assumption of our approach is that a textured *three dimensional* (3D) model of the tracking environment is available. Such a model can be obtained from for instance technical drawings or structure from motion algorithms. By using a 3D scene model containing natural landmarks, there is no need for a prepared environment with artificial markers. This allows for AR applications outside dedicated studios, for instance in outdoor environments.

Given a reasonably accurate prediction of the camera pose, e.g. from inertial navigation, an artificial image can be obtained by projecting the 3D model. This artificial image will resemble the camera image and is used to construct *two dimensional* (2D) search templates which are matched against the camera image,

see Figure 6.3. For a successful match the association problem is already solved

Figure 6.3: *Correspondences are generated by comparing the 3D scene model viewed from the predicted camera pose (left) to the actually observed camera image (right).*

and a correspondence is obtained directly. A schematic overview of the sensor fusion approach is shown in Figure 6.4. The IMU provides rapid measurements

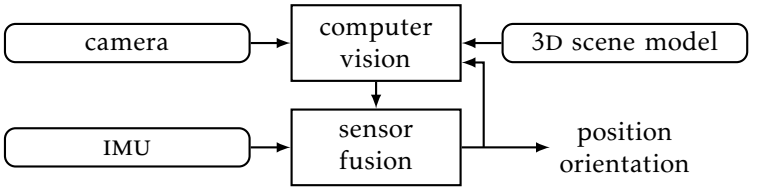


Figure 6.4: *Estimating camera pose by fusing measurements from an inertial measurement unit and a computer vision system.*

of acceleration and angular velocity. The image processing system generates correspondences between the camera image and the scene model. This 3D scene model contains positions of various natural markers and is generated offline using images and/or drawings of the scene. The inertial and vision measurements are combined in the sensor fusion model to obtain the camera pose.

When combining vision with inertial sensors, it is very important that the 3D scene model both has a metric scale as well as a known gravity vector. These properties are implicitly present in Section 3.2 where the scene model is defined in the navigation frame (n-frame). However, coming from a pure vision perspective, they are often overlooked. Since a camera is a projective device, it suffers from scale ambiguity. Changing the scale of a scene model gives scaled, but indistinguishable tracking results. Similarly, cameras have no sense of vertical, so changing the orientation of the scene model gives rotated but indistinguishable tracking results. Hence, for vision-only applications scene models typically have an arbitrary scale and an arbitrary rotation; a standard choice is to define the unit length and x-direction using the first two images.

For the inertial-vision combination, the scale and orientation are relevant. Sensor fusion utilizes position information both from the camera and the IMU, which implies that these quantities must have identical units. Additionally, the accelerometers measure gravity which implies that the scene model should be vertically aligned, or equivalently the gravity vector has to be known in the scene model. Scale is also important when assumptions are made about the motions of the camera, for instance the type and parameters of a motion model (Davison et al., 2007). To determine the scale of a scene model, one can compare it to a *computer aided design* (CAD) model or an object with known dimensions. The gravity vector can be determined by averaging the accelerometer measurements over some time while the sensor unit is stationary in a known pose. An interesting solution might be to include metric and direction information, for instance using accelerometers, in the algorithms for building the scene models, for instance using SLAM.

As discussed in Chapter 2, state-space models are one of the key ingredients of sensor fusion. For the sensor unit introduced earlier in this section, such a model can be constructed from the models in Chapter 3 and Chapter 4. They are repeated below for the reader's convenience.

According to Section 3.2.1, the correspondences are modeled as

$$\boldsymbol{p}^{i}_{t,k} = \mathcal{P}(\boldsymbol{p}^{c}_{t,k}) + \boldsymbol{e}^{i}_{t,k}, \tag{6.1}$$

where $\boldsymbol{p}^{i}_{t,k}$ is a 2D feature observed in the image, $\boldsymbol{p}^{c}_{t,k}$ its corresponding 3D position in the camera frame (c-frame) and $\boldsymbol{e}^{i}_{t,k}$ is *independently and identically distributed* (i.i.d.) Gaussian noise. The projection function $\mathcal{P}$ consists of (3.14) in combination with (3.10) or (3.12), dependent on whether a perspective or wide-angle objective is used.

Following Section 4.3, we model the position and orientation of the IMU using the inertial measurements,

$$\boldsymbol{b}^{n}_{t+1} = \boldsymbol{b}^{n}_{t} + T\dot{\boldsymbol{b}}^{n}_{t} + \tfrac{T^2}{2}\ddot{\boldsymbol{b}}^{n}_{t}, \tag{6.2a}$$

$$\dot{\boldsymbol{b}}^{n}_{t+1} = \dot{\boldsymbol{b}}^{n}_{t} + T\ddot{\boldsymbol{b}}^{n}_{t}, \tag{6.2b}$$

$$q^{nb}_{t+1} = q^{nb}_{t} \odot \exp(\tfrac{T}{2}\boldsymbol{\omega}^{b}_{nb,t}), \tag{6.2c}$$

where $\boldsymbol{b}^n$ and $\dot{\boldsymbol{b}}^n$ denote the position and velocity of the body frame (b-frame) resolved in the n-frame, q^{nb} is a unit quaternion describing the orientation of the n-frame relative to the b-frame and T denotes the sampling interval. The acceleration $\ddot{\boldsymbol{b}}^n$ and angular velocity $\boldsymbol{\omega}^{b}_{nb}$ are calculated from the accelerometer measurements $\boldsymbol{y}_{\mathrm{a}}$ and the gyroscope measurements $\boldsymbol{y}_{\omega}$ using (3.2) and (3.3),

$$\ddot{\boldsymbol{b}}^{n}_{t} = R^{nb}_{t}\left(\boldsymbol{y}_{\mathrm{a},t} - \boldsymbol{\delta}^{b}_{\mathrm{a},t} - \boldsymbol{e}^{b}_{\mathrm{a},t}\right) - 2\boldsymbol{\omega}^{n}_{ie} \times \dot{\boldsymbol{b}}^{n}_{t} + \boldsymbol{g}^{n}, \tag{6.3a}$$

$$\boldsymbol{\omega}^{b}_{nb,t} = \boldsymbol{y}_{\omega,t} - R^{bn}_{t}\boldsymbol{\omega}^{n}_{ie} - \boldsymbol{\delta}^{b}_{\omega,t} - \boldsymbol{e}^{b}_{\omega,t}. \tag{6.3b}$$

The inertial bias terms $\boldsymbol{\delta}^{b}_{\mathrm{a}}$ and $\boldsymbol{\delta}^{b}_{\omega}$ are slowly time-varying. Hence, they are mod-

eled as random walks,

$$\boldsymbol{\delta}^b_{\mathrm{a},t+1} = \boldsymbol{\delta}^b_{\mathrm{a},t} + \boldsymbol{e}^b_{\delta_\mathrm{a},t}, \tag{6.4a}$$

$$\boldsymbol{\delta}^b_{\omega,t+1} = \boldsymbol{\delta}^b_{\omega,t} + \boldsymbol{e}^b_{\delta_\omega,t}, \tag{6.4b}$$

where $\boldsymbol{e}^b_{\delta_\mathrm{a}}$ and $\boldsymbol{e}^b_{\delta_\omega}$ are i.i.d. Gaussian noises.

The correspondences provided by the computer vision algorithm contain $\boldsymbol{p}^n_{t,k}$, the position of the feature in the n-frame. Since the camera is rigidly attached to the IMU, these quantities can be expressed in the c-frame as required by (6.1) using

$$\boldsymbol{p}^c_{t,k} = R^{cb}\left(R^{bn}_t(\boldsymbol{p}^n_{t,k} - \boldsymbol{b}^n_t) - \boldsymbol{c}^b\right). \tag{6.5}$$

Here, R^{cb} is the rotation between the c-frame and the b-frame and $\boldsymbol{c}^b$ is the position of the c-frame expressed in the b-frame. An accurate value for this relative pose is of utmost importance to obtain a system that works in practice. Section 6.3 introduces a calibration method to obtain this. For now, we assume that the relative pose is known.

Combining (6.1)–(6.5) we obtain a discrete-time nonlinear state-space model. Its state vector is, with slight abuse of notation, given by

$$\boldsymbol{x} = \left(\boldsymbol{b}^n, \dot{\boldsymbol{b}}^n, q^{nb}, \boldsymbol{\delta}^b_\mathrm{a}, \boldsymbol{\delta}^b_\omega,\right)^T. \tag{6.6}$$

This model is used in an *extended Kalman filter* (EKF) to fuse the vision measurements and the inertial measurements, see Section 2.2. The EKF handles the different sample rates and a varying number of correspondences straightforwardly. It runs at the high data rate of the IMU and the vision updates are only performed when correspondences are available. The IMU gives very accurate short-term predictions of the camera pose which can be used to simplify the computer vision significantly.

The above discussion is summarized in Algorithm 6.1.

Algorithm 6.1 Sensor fusion using inertial and vision measurements

1. Perform an initialization and set $p(\boldsymbol{x}_0)$.
2. Do a time update. Propagate $p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1})$ to $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})$ using the process model (6.2), (6.3) and (6.4) with the inertial measurements as input signals.
3. If a new image is available
   (a) Predict feature positions. Obtain $p(\boldsymbol{y}_t|\boldsymbol{y}_{1:t-1})$ from $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})$ and the measurement model (6.1) and (6.5).
   (b) Detect features in the image by searching around their predicted positions.
   (c) Do a measurement update. Use the measured features and the measurement model to correct $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})$ and obtain $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$.
4. Set $t := t + 1$ and iterate from Step 2.

6.2.2 Experiments

The pose estimation system has been tested in a number of scenarios. Its accuracy has been evaluated using an industrial robot as ground truth (Hol et al., 2007). Furthermore, the system has been tested as an augmented reality application (Chandaria et al., 2007b). Both experiments are discussed in this section.

Performance validation

The sensor unit is mounted onto a high precision ABB IRB1440 industrial robot, see Figure 6.5. The reason for this is that the robot will allow us to make repeat-

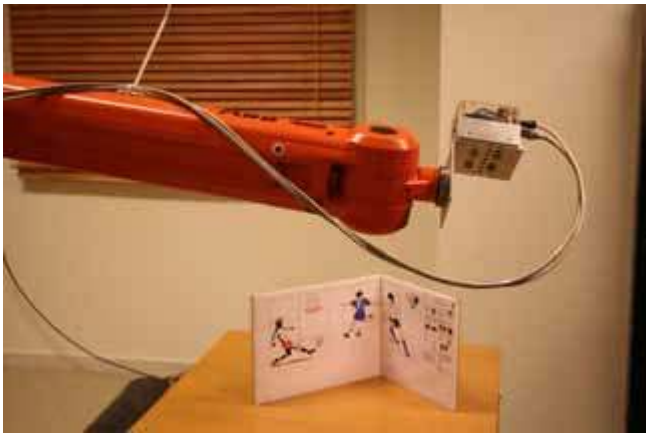


Figure 6.5: *A prototype of the sensor unit is mounted onto an industrial robot. The background shows the scene that has been used in the experiments.*

able 6 DOF motions and it will provide the true position and orientation. The robot has an absolute accuracy of 2 mm and a repeatability of 0.2 mm. This enables a systematic and objective performance evaluation based on absolute pose errors instead of the commonly used feature reprojection errors. The sensor unit contains 1200°/s gyroscopes, 17 m/s^2 accelerometers as well as a 320 × 240 pixel color camera equipped with a 3.2 mm lens. It provides 100 Hz inertial measurements synchronized with 12.5 Hz images.

The scene used for the experiments consists of two orthogonal planar surfaces as shown in Figure 6.5. Because of the simple geometry, the scene model could be constructed from a textured CAD model. Its coordinate system is such that the x-axis points upward and that the y and z-axis span the horizontal plane. Although the scene was carefully positioned, it had to be calibrated *with respect to* (w.r.t.) gravity. It should be emphasized that the scene has been kept simple for experimentation purposes only. The system itself can handle very general scenes as long as a textured scene model is available, for instance generated using the methods described in Koeser et al. (2007).

With the setup several trajectories have been tested. In this thesis, an eight-shaped trajectory, shown in Figure 6.6, will be discussed in detail. The sensor

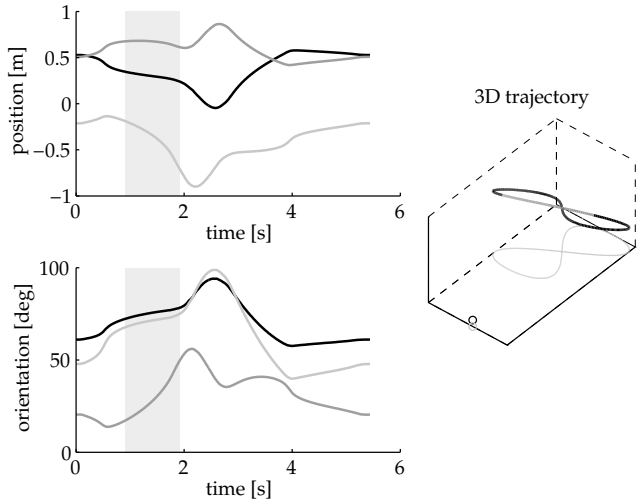


Figure 6.6: *The eight-shaped trajectory undertaken by the sensor unit. The gray shaded parts mark the interval where vision is deactivated. The circle indicates the origin of the scene model.*

unit traverses this 2.6 m eight-shaped trajectory in 5.4 s, keeping the scene in view at all times. The motion contains accelerations up to 4 m/s^2 and angular velocities up to 1 rad/s. Hence, the motion is quite aggressive and all six degrees of freedom are excited. As the displacement between images is limited to 15 pixels it is still possible to use vision-only tracking, which allows for a comparison between tracking with and without an IMU. To test the robustness of the system, the vision has been deactivated for 1 s.

The experiment starts with a synchronization motion, which is used to synchronize the ground truth data from the industrial robot with the estimates from the system. Time synchronization is relevant, since a small time offset between the signals will result in a significant error. After the synchronization, the eight-shaped trajectory (see Figure 6.6) is repeated several times, utilizing the accurate and repeatable motion provided by the industrial robot.

By comparing the estimates from the filter to the ground truth the tracking errors are determined. Examples of position and orientation errors (z, roll) are shown in Figure 6.7. The other positions (x, y) and orientations (yaw, pitch) exhibit similar behavior. The absolute accuracy (with vision available) is below 2 cm for position and below 1° for orientation. These values turn out to be typical for the performance of the system in the setup described above. Furthermore, the accuracy of the IMU is not affected by the speed of motion, resulting in a tracking accuracy which is rather independent of velocity, as illustrated by Figure 6.8 which shows the *root mean square error* (RMSE) tracking error of the eight-shaped trajectory

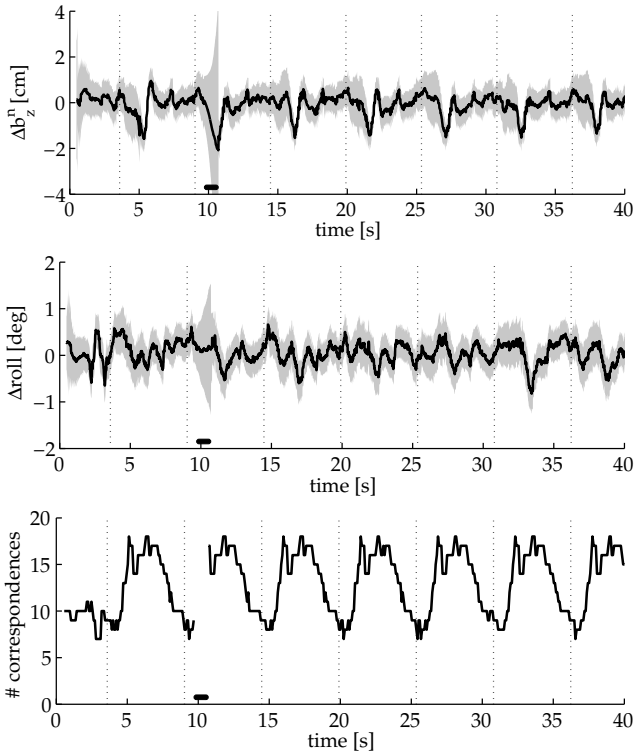


Figure 6.7: *Tracking error during multiple passes of the eight-shaped trajectory. The black line shows the position (z) and orientation (roll) errors, as well as the number of correspondences that were used. The gray band illustrates the 99% confidence intervals. Note that vision is deactivated from* 9.7 s *to* 10.7 s. *The vertical dotted lines mark the repetition of the motion.*

executed at various speeds. Other experiments, not described here, show similar

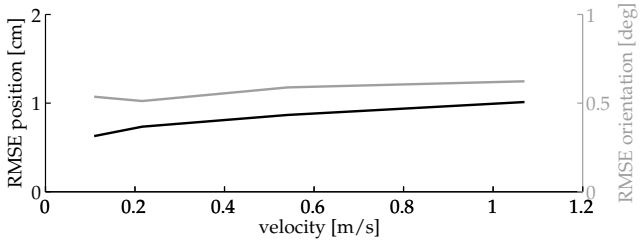


Figure 6.8: *Tracking error for several experiments executing the eight-shaped trajectory at different speeds.*

performance for various trajectories.

The measurements from the IMU result in better predictions of the feature positions in the acquired image. This effect is clearly illustrated in Figure 6.9, which provides a histogram of the feature prediction errors. The figure shows that the feature prediction errors are smaller and more concentrated in case the IMU measurement updates are used. This improvement is most significant when the cam-

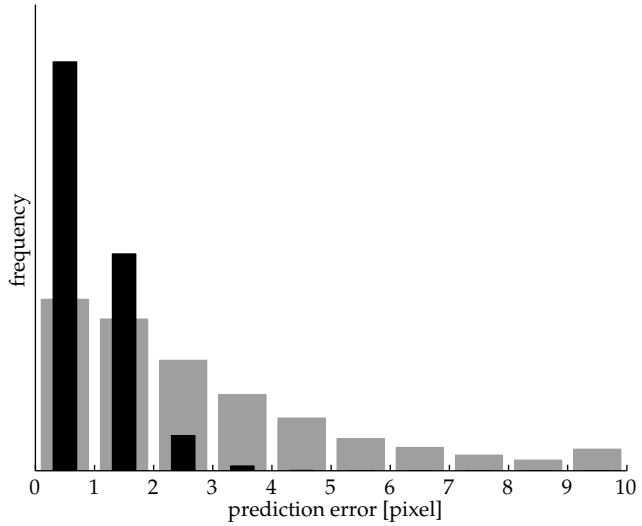


Figure 6.9: *Histogram of the prediction errors for the feature positions. Shown are feature predictions calculated using the latest vision pose (gray) and from the most recent inertial pose (black).*

era is moving fast or at lower frame rates. At lower speeds, the vision based feature predictions will improve and the histograms will become more similar.

The system tracks the camera during the entire experiment, including the period

where vision is deactivated. The motion during this period, indicated using gray segments in Figure 6.6, is actually quite significant. Vision-only tracking has no chance of dealing with such a gap and loses track. Indeed, such an extensive period where vision is deactivated is a little artificial. However, vision might be unavailable or corrupted, due to fast rotations, high velocity, motion blur, or simply too few visible features. These difficult, but commonly occurring, situations can be dealt with by using an IMU as well, clearly illustrating the benefits of having an IMU in the system. In this way, robust real-time tracking in realistic environments is made possible.

Augmented reality

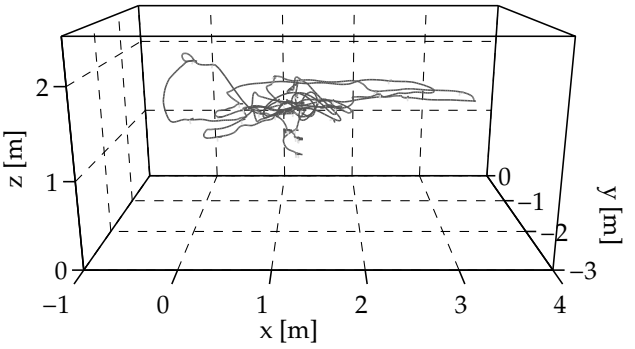
The system has been used to track the sensor unit in a relatively large room, approximately $5 \times 4 \times 2.5$ m in size, see Figure 6.10. The sensor unit is handheld

Figure 6.10: *The sensor unit is tracked in a large room. The monitor shows the live camera image augmented with a virtual character.*

and is allowed to move without constraints in this room, both close to and far away from the walls. The pose output of the pose estimation system is used to draw virtual graphics on top of the camera images in real-time. There is no ground-truth data available for this test, implying that the tracking performance has to be evaluated qualitatively from the quality of the augmentation.

The pose estimation system requires a 3D model of the environment. In this case, the model was not generated using the computer vision approaches described in Section 3.2.3, but created manually using a 3D modeling tool. This tool takes the geometry from a building model and uses digital photos to obtain textures for the surfaces. The resulting model, shown in Figure 6.11a, consists of the three main walls. The floor and roof do not contain sufficient features and are ignored, together with the fourth wall containing mostly windows.

(a) *3D model of the room.*



(b) *Camera trajectory.*

Figure 6.11: *Overview of the test setup.*

The system worked very well for the described setup. The augmentations showed no visible jitter or drift, even during fast motion. Tracking continued for extensive periods of time without deterioration or divergence. Furthermore, the system is capable to handle periods with few or no features at all, which pose difficulties for pure computer vision approaches. These situations occur for instance when the camera is close to a wall or during a 360° revolution. A reinitialization was required after 2 s without visible features. Beyond that period, the predicted feature positions were to far off to enable detection.

A sample trajectory of about 90 s is shown in Figure 6.11b. It contains acceleration up to 12 m/s^2 and angular velocity up to 9.5 rad/s. Furthermore, the trajectory involves several 360° rotations which include several views where the camera only observes the unmodeled window wall. An impression of the augmentation result is given by Figure 6.12. The overlaid graphics stay in the same location, regardless of the position and orientation of the camera. This is also the case when no features are available, for instance when only the unmodeled wall is in view, see Figure 6.12e. The above discussion has shown that the vision/IMU tracking system performs very well on a realistic AR scenario.

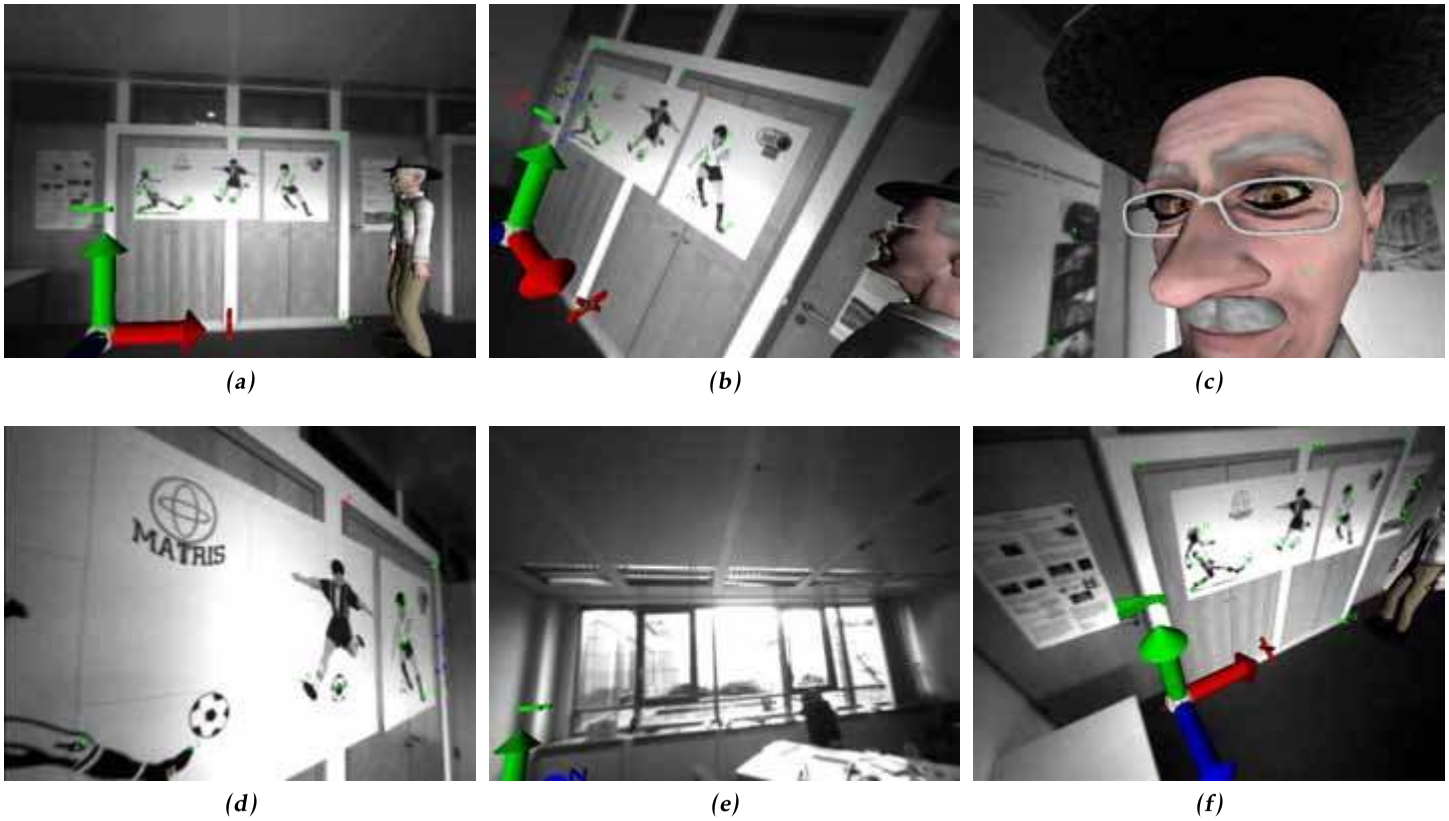


Figure 6.12: *Several frames taken from the sample trajectory. Shown are the camera images, located feature positions (+) and overlaid graphics (avatar, coordinate frames).*

6.3 Relative pose calibration

In this section we describe a flexible and easy-to-use calibration algorithm to calibrate the relative position and orientation between an IMU and a camera.

6.3.1 Calibration algorithm

For the setup discussed in Section 6.2 the camera and the IMU are rigidly connected, i.e., the relative position c^b and the relative orientation φ^{cb} are constant. The position of the camera is in this setting defined as the position of its optical center. Although this is a theoretically well-defined quantity, its physical location is rather hard to pinpoint without exact knowledge of the design of the optical system and typically a calibration algorithm has to be used to locate it. In this section we develop a calibration algorithm which provides high quality estimates of the relative translation and orientation. The proposed calibration algorithm is fast and more importantly, it is simple to use in practice. We also provide a quality measure for the estimates in terms of their covariance.

Current state-of-the-art when it comes to calibration of the relative translation and orientation between a camera and an IMU is provided by Lobo and Dias (2007) and Mirzaei and Roumeliotis (2008); Kelly and Sukhatme (2011). Lobo and Dias (2007) presents a rather labor-intensive two-step algorithm in which the relative orientation is determined first and then the relative position is determined using a turntable. The approach of Mirzaei and Roumeliotis (2008); Kelly and Sukhatme (2011) is to transform the parameter estimation problem into a state estimation problem by augmenting the state vector x_t used for pose estimation with the calibration parameters θ and then estimating the augmented vector using an EKF. Furthermore, the work of Foxlin and Naimark (2003) is worth mentioning, where a custom calibration rig is used together with a set of artificial landmarks. A closely related topic is coordinate frame alignment, see Section 4.4.

For our calibration algorithm we have taken a system identification approach, see Section 2.1. Combining (6.1)–(6.3) and (6.5) we obtain a discrete-time nonlinear state-space model

$$x_{t+1} = f(x_t, u_t, \theta) + w_t, \tag{6.7a}$$

$$y_t = h(x_t, \theta) + v_t. \tag{6.7b}$$

This model is parameterized by the state vector x_t and the parameter vector θ which are defined, with slight abuse of notation, as

$$x = \left(b^n, \dot{b}^n, q^{nb}\right)^T, \tag{6.8a}$$

$$\theta = \left(\varphi^{cb}, c^b, \delta^b_\omega, \delta^b_a, g^n\right)^T. \tag{6.8b}$$

The inertial bias terms δ^b_ω and δ^b_a are slowly time-varying and are therefore typically included in the process model (6.7a). However, for calibration purposes a few seconds of data are typically sufficient. Therefore, the biases are treated

as constants and included in the parameter vector θ. Besides the relative pose φ^{cb} and c^b, θ contains several parameters that we are not directly interested in, so-called nuisance parameters, for example the gravity in the n-frame g^n. Even though we are not directly interested in these nuisance parameters, they affect the estimated camera trajectory and they have to be taken into account to obtain accurate estimates of φ^{cb} and c^b.

For a given parameter vector θ, the state-space model (6.7a) is fully specified and can be used in sensor fusion methods such as the EKF, as is done in Section 6.2. As a part of its processing, the EKF computes the one-step ahead predictor $\hat{y}_{t|t-1}$ by applying the measurement model $h(\cdot)$ to the state prediction $\hat{x}_{t|t-1}$. This predictor,

$$\hat{y}_{t|t-1}(\theta) = h(\hat{x}_{t|t-1}(\theta), \theta), \tag{6.9}$$

whose dependence on θ is denoted explicitly, can be used to formulate the system identification problem (2.13) as

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{2} \sum_{t=1}^{N} \|y_t - \hat{y}_{t|t-1}(\theta)\|^2_{S_t^{-1}(\theta)} \tag{6.10}$$

where $S_t(\theta)$ is the covariance of the prediction error $\varepsilon_t = y_t - \hat{y}_{t|t-1}(\theta)$ calculated by the EKF. That is, the prediction errors ε_t, also called innovations, are weighted by their corresponding inverse covariance. This approach is a special case of gray-box system identification (Ljung, 1999; Graebe, 1990). An overview is given in Figure 6.13. Note that all the quantities in (6.10) are computed by processing the

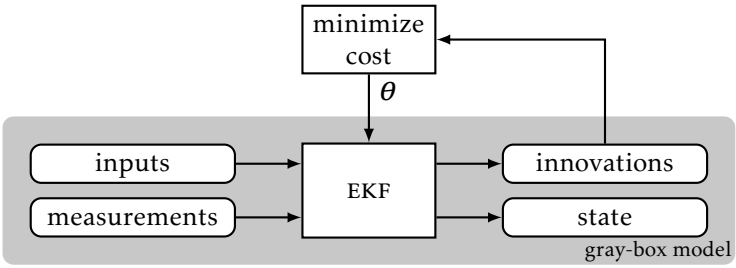


Figure 6.13: *Gray-box system identification using EKF innovations. The parameter vector θ is adjusted to minimize the cost function given in* (6.10).

complete dataset with the EKF, for a given value of θ. This makes our approach an offline calibration, which does not constrain its applicability. The covariance of the obtained estimate $\hat{\theta}$ is given by (2.17).

The measurement model (6.1) and hence our calibration algorithm works with any kind of correspondences. Without loss of generality we simplify the correspondence generation problem and work with checkerboard patterns of known size typically used for camera calibration. In this case, obtaining the correspondences is relatively easy due to the strong corners and simple planar geometry.

The required image processing is typically implemented in off-the-shelf camera calibration software (Bouguet, 2003; Scaramuzza et al., 2006).

We now can introduce Algorithm 6.2, a flexible algorithm for estimating the relative pose of the IMU and the camera. The dataset is captured without requiring any additional hardware, except for a standard camera calibration pattern of known size that can be produced with a standard printer. The motion of the sensor unit can be arbitrary, provided it contains sufficient rotational excitation. A convenient setup for the data capture is to mount the sensor unit on a tripod and pan, tilt and roll it, but hand-held sequences can be used equally well. Estimates of the relative position and orientation as well as nuisance parameters such as sensor biases and gravity are obtained by solving (6.10), see Section 2.3. The optimization is started, with slight abuse of notation, in

Algorithm 6.2 Relative Pose Calibration

1. Place a camera calibration pattern on a horizontal, level surface, e.g. a desk or the floor.
2. Acquire inertial measurements $\{\boldsymbol{y}_{\mathrm{a},t}\}_{t=1}^{N}, \{\boldsymbol{y}_{\omega,t}\}_{t=1}^{N}$ together with synchronized images $\{I_t\}_{t=1}^{N}$.
   - Rotate about all 3 axes, with sufficiently exciting angular velocities.
   - Always keep the calibration pattern in view.
3. Obtain the point correspondences between the 2D corner locations $\boldsymbol{p}_{t,k}^{t}$ and the corresponding 3D grid coordinates $\boldsymbol{p}_{t,k}^{n}$ of the calibration pattern for all images $\{I_t\}_{t=1}^{N}$.
4. Solve the gray-box identification problem (6.10), starting the optimization from $\boldsymbol{\theta}_0 = (\hat{\boldsymbol{\varphi}}_0^{cb}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \boldsymbol{g}_0^n)^T$. Here, $\boldsymbol{g}_0^n = (0, 0, -g)^T$ since the calibration pattern is placed horizontally and $\hat{\boldsymbol{\varphi}}_0^{cb}$ can be obtained using Algorithm 6.3.
5. Validate the calibration result by analyzing the obtained state trajectory, normalized innovations and parameter covariance (2.17). If necessary, start over from Step 2.

$$\boldsymbol{\theta}_0 = \left(\hat{\boldsymbol{\varphi}}_0^{cb}, \mathbf{0}, \mathbf{0}, \mathbf{0}, \boldsymbol{g}_0^n\right)^T \tag{6.11}$$

Here, $\boldsymbol{g}_0^n = (0, 0, -g)^T$ since the calibration pattern is placed horizontally and $\hat{\boldsymbol{\varphi}}_0^{cb}$ is an initial estimate of the relative orientation. It can be obtained by applying Theorem 4.1 on measurements from a slightly extended camera calibration, similar to Lobo and Dias (2007).

Performing a standard camera calibration as described in Section 3.2.2 with the calibration pattern placed on a horizontal, level surface, a vertical reference can be obtained from the calculated extrinsic parameters. Furthermore, when holding the sensor unit stationary, the accelerometers measure gravity, another vertical reference. From these two ingredients an initial orientation can be obtained using Theorem 4.1, resulting in Algorithm 6.3.

Algorithm 6.3 Initial Orientation

1. Place a camera calibration pattern on a horizontal, level surface, e.g. a desk or the floor.
2. Acquire images $\{I_k\}_{k=1}^K$ of the pattern while holding the sensor unit stationary in various poses, simultaneously acquiring accelerometer readings $\{\boldsymbol{y}_{\mathrm{a},k}\}_{k=1}^K$.
3. Perform a camera calibration using the images $\{I_k\}_{k=1}^K$ to obtain the orientations $\{R_k^{cn}\}_{k=1}^K$.
4. Compute an estimate $\hat{q}_0^{cb}$ from the vectors $\boldsymbol{g}_k^c = R_k^{cn}\boldsymbol{g}_0^n$ and $\boldsymbol{g}_k^b = -\boldsymbol{y}_{\mathrm{a},k}$ using Theorem 4.1. Here, $\boldsymbol{g}_0^n = (0, 0, -g)^T$ since the calibration pattern is placed horizontally.

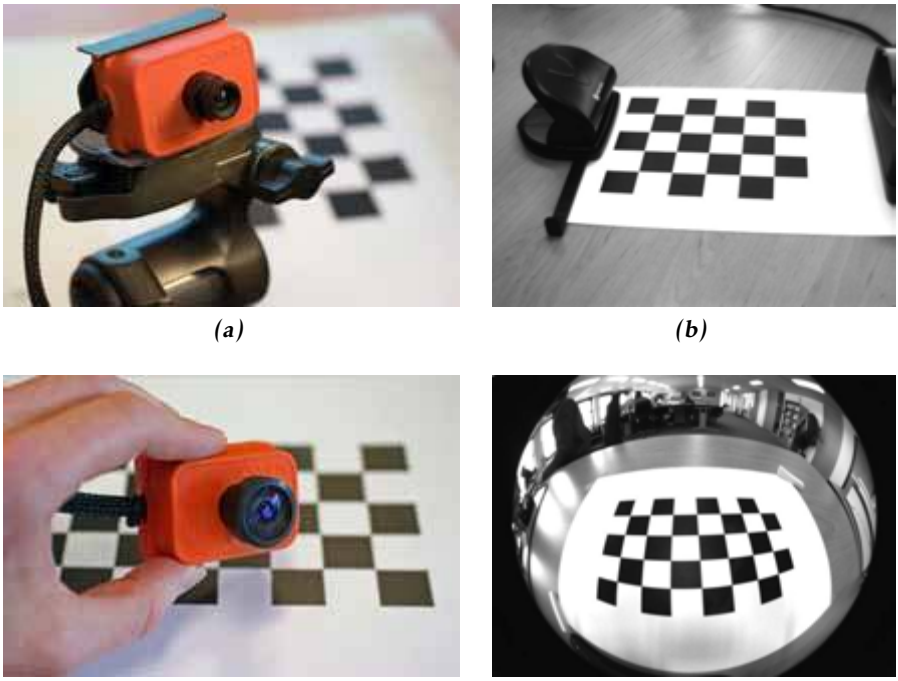
6.3.2 Experiments

The sensor unit has been equipped with both perspective and fisheye lenses, see Figure 6.14. In both configurations the sensor unit has been calibrated according to Algorithm 6.2, using nothing but a planar checkerboard pattern of known size as in a standard camera calibration setup. The calibration data was gathered according to the following protocol

1. The checkerboard pattern is placed on a horizontal, planar surface.
2. The sensor unit is held stationary in 8–12 different poses, similar to what is done during a standard camera calibration. For each pose, a single image is captured together with 1 s of inertial measurements at 100 Hz.
3. The sensor unit is subjected to 10–20 s of rotational motion around all three axes, while keeping the calibration pattern in view. The angular velocity during this rotational motion should be similar to the application being calibrated for. The inertial data is sampled at 100 Hz and the camera has a frame rate of 25 Hz. Due to the limited field of view, the sensor unit is mounted on a tripod and rotated in pan, tilt and roll direction, when equipped with the perspective lens. For the fisheye configuration, hand held sequences are used.

The measurements obtained in Step 2 are used in Algorithm 6.3 to determine an initial orientation. The measurements from Step 3 are used in Algorithm 6.2 to estimate the relative translation and orientation between the camera and the IMU. An example of a typical trajectory is given in Figure 6.15. To facilitate cross-validation, the measurements are split into an estimation part and a validation part (Ljung, 1999), both containing similar motion. The parameters are estimated from the estimation data and the quality of the estimates is assessed using the validation data.

A number of different sensor units and/or different lens configurations have been calibrated using the above protocol. The resulting estimates of the relative position and orientation of the camera and the IMU, $\boldsymbol{c}^b$ and $\boldsymbol{\varphi}^{cb}$, together with



(a) *(b)*

(c) *(d)*

Figure 6.14: *Two configurations of the sensor unit. In (a) and (b) a* 4 mm *perspective lens is used and in (c) and (d) a* 190° *fisheye lens is used.*

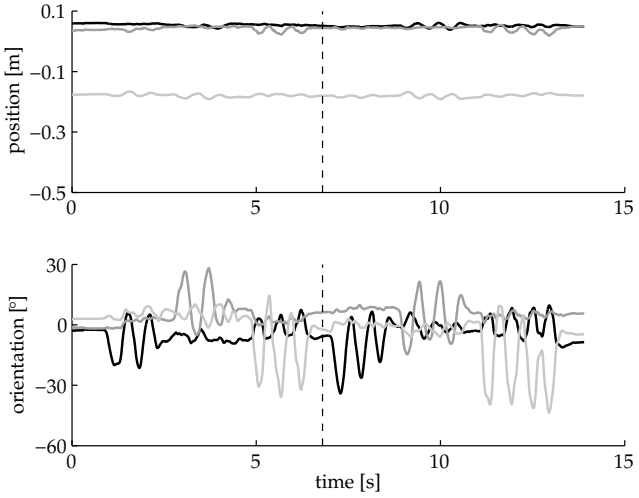


Figure 6.15: *Example trajectory of the sensor unit used for calibration. It contains both estimation data ($t < 6.8$ s) and validation data ($t \geq 6.8$ s), separated by the dashed line.*

their standard deviation calculated using Algorithm 6.2, are listed in Table 6.1. Table 6.1 shows that Algorithm 6.2 has been successfully applied to five differ-

Table 6.1: *Relative pose estimates and 99% confidence intervals for five different sensor units and several different lens configurations.*

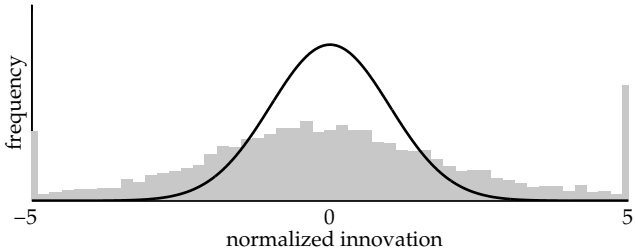
Unit	Lens	$\hat{\varphi}^{cb}$ (°)	$\hat{c}^b$ (mm)
1	4 mm	(-0.52, 0.43, 0.94) ± 0.04	(-17.6, -4.8, 22.1) ± 0.9
2	6 mm	(0.23, -0.34, 0.02) ± 0.05	(-17.6, -6.2, 28.3) ± 1.4
3	6 mm	(-0.53, 0.97, 0.29) ± 0.02	(-14.9, -6.7, 29.8) ± 0.5
4	6 mm	(-0.02, 0.21, -0.20) ± 0.04	(-18.1, -8.7, 31.0) ± 0.9
5	6 mm	(-0.27, 0.94, 0.09) ± 0.13	(-14.0, -7.0, 30.3) ± 1.3
5	Fisheye	(0.08, 0.17, 0.06) ± 0.14	(-17.4, -4.9, 38.7) ± 0.4
Reference[a]		(0, 0, 0)	(-14.5, -6.5, —)

[a] using the technical drawing of the sensor unit.

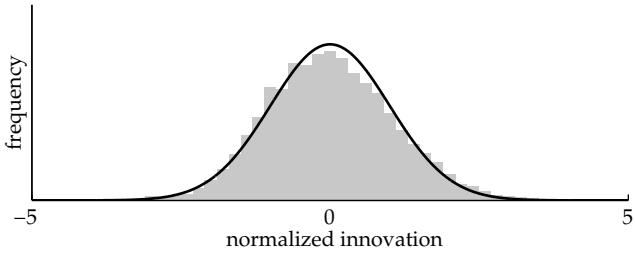
ent sensor units equipped with both perspective and fisheye lenses. Consistent results are obtained for multiple trials of the same configuration, which further reinforces the robustness and reliability of the proposed method. Table 6.1 also contains reference values obtained from the technical drawing. Note that the drawing defines the center of the image sensor, not the optical center of the lens. Hence, no height reference is available and some shifts can occur in the tangential directions.

In order to further validate the estimates, the normalized innovations of the EKF are studied. Histograms of the normalized innovations (for validation data) are given in Figure 6.16. Figure 6.16a and 6.16c show the effect of using wrong parameter vectors, in this case being the initial guess. After calibration, the normalized innovations are close to white noise, as shown in Figure 6.16b and 6.16d. This implies that the model with the estimated parameters and its assumptions appears to be correct, which in turn is a very good indication that reliable estimates $\hat{\boldsymbol{\varphi}}^{cb}$ and $\hat{\boldsymbol{c}}^{b}$ have been obtained.

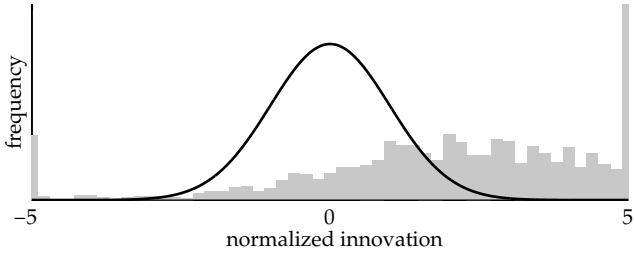
The calibration results shown in Table 6.1 are close to the reference values, but show individual differences between the different sensor units and lens configurations. These differences are significant, which is further illustrated in Figure 6.17. This figure illustrates the behavior when applying the calibration values of one sensor unit to a second sensor unit having the same type of lens. Notice the characteristic saw-tooth behavior present in the position plot. It is present in all three position channels and explains the big difference between the obtained normalized innovations and the theoretical distribution. When the correct calibration parameters are used this saw-tooth behavior is absent, which is illustrated in Figure 6.15. To summarize, the significant individual differences once more illustrate the need for an easy-to-use calibration method, since each sensor unit has to be individually calibrated for optimal performance.



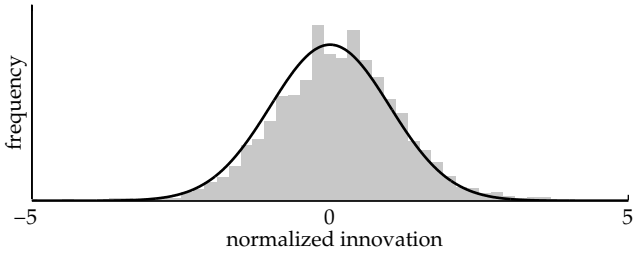
(a) *fisheye lens, using* $\boldsymbol{\theta}_0$



(b) *fisheye lens, using* $\hat{\boldsymbol{\theta}}$



(c) *perspective lens, using* $\boldsymbol{\theta}_0$



(d) *perspective lens, using* $\hat{\boldsymbol{\theta}}$

Figure 6.16: *Histograms of the normalized innovations, for validation data. Both the empirical distribution (gray bar) as well as the theoretical distribution (black line) are shown.*

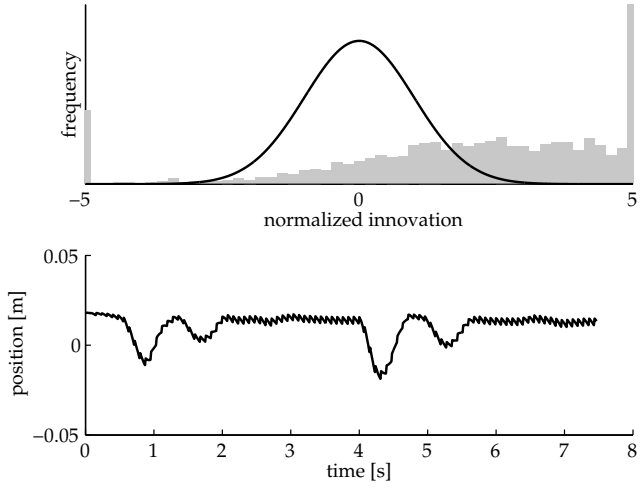


Figure 6.17: *Typical behavior obtained when using sensor unit A with calibration values of sensor unit B. The figure shows the empirical distribution (grey bar) and the theoretical distribution (black line) (top) as well as the x position trajectory (bottom).*

7

Inertial and UWB

The combination of *inertial measurement units* (IMUs) and *ultra-wideband* (UWB) technology is very suitable for pose estimation, especially for indoor applications. This chapter discusses sensor fusion using this combination of sensors, together with the associated problems of UWB positioning and calibration.

7.1 Problem formulation

Commercially available UWB asset tracking systems typically consist of a network of synchronized receivers which track a large number of small, battery powered and inexpensive transmitters. As already discussed in Section 3.3, reported indoor position accuracies lie in the order of decimeters, but suffer from multipath effects and *non line of sight* (NLOS) conditions. These effects are most prominent while tracking moving objects or persons and give rise to distorted and bumpy trajectories. Although the obtained performance is often sufficient for the aforementioned applications, many potential application areas have higher performance requirements.

To improve the tracking performance (especially the positioning accuracy) we propose to combine UWB with a low-cost IMU consisting of a *three dimensional* (3D) rate gyroscope and a 3D accelerometer. The main justification for adding an IMU — providing accurate position tracking for short periods of time, but drift prone for longer timescales — is to obtain a robust system, capable of detecting and rejecting multipath effects and NLOS situations. Additional benefits of adding an IMU include improved tracking results, especially for dynamic quantities like velocity, and that the orientation becomes observable as well. This results in a system providing a 6 *degrees of freedom* (DOF) general purpose tracking so-

lution for indoor applications.

Besides a good sensor fusion method to combine inertial and UWB measurements, two auxiliary problems need to be addressed to successfully apply the proposed tracking technology in real applications. Firstly, the UWB setup needs to be calibrated. Current calibration algorithms require surveying positions of receivers and transmitters. This makes calibration a very labor-intensive procedure which is only suitable for permanent setups. A new calibration method which does not have this limitation is very desirable, since it allows deployment of portable, temporary UWB systems and opens up many new application areas. The second auxiliary problem is that a robust UWB positioning algorithm is required to initialize the sensor fusion algorithm. The UWB measurements suffer from outliers due to multipath and NLOS conditions. These outliers occur quite frequently in practice and a multilateration algorithm which accounts for this is required to provide an initial position.

The remainder of this chapter is organized as follows: the UWB calibration problem is discussed in Section 7.2, UWB multilateration with outlier rejection is discussed in Section 7.3 and the sensor fusion method used to obtain pose estimates from inertial and UWB measurements is discussed in Section 7.4. Earlier versions of the work in this chapter have been published in Hol et al. (2009, 2010c).

7.2 UWB calibration

In this section we describe the calibration method as employed for commercially available UWB hardware and show how we extend it to a flexible and easy-to-use calibration algorithm.

7.2.1 Existing calibration algorithms

The UWB setup consists of a network of synchronized and stationary (rigidly fixed, mounted) receivers, all taking very precise *time of arrival* (TOA) measurements of signals originating from a mobile transmitter, see Section 3.3. The TOA measurement y_m is the time (according to its local clock) when receiver m receives a pulse from the transmitter. Repeating (3.18), it can be modeled as

$$y_{\mathrm{u},m} = \tau + \|\boldsymbol{r}_m - \boldsymbol{t}\|_2 + \Delta\tau_m + \delta_{\mathrm{u},m} + e_{\mathrm{u},m}, \tag{7.1}$$

where τ is the time of transmission of the pulse, $\boldsymbol{t}$ is the position of the transmitter, $\boldsymbol{r}_m$ is the position of the m-th receiver and $\Delta\tau_m$ is the clock-offset of the m-th receiver. $\delta_{\mathrm{u},m} \geq 0$ is a possibly nonzero delay due to NLOS or multipath and $e_{\mathrm{u},m}$ is *independently and identically distributed* (i.i.d.) Gaussian noise. For calibration, we assume $\delta_{\mathrm{u},m} = 0$.

Current state-of-the-art calibration methods focus exclusively on estimating the receiver clock-offset $\Delta\tau_m$. They require a dataset consisting of K transmissions from $L \geq 1$ transmitters and M receivers. Both the position of the transmitters and the position of the receivers are assumed to be known with values $\bar{\boldsymbol{t}}_l$ and

$\bar{\boldsymbol{r}}_m$, respectively. In that case, the clock-offset differences $\Delta\tau_m - \Delta\tau_1$ are straightforwardly given from (averaged) *time difference of arrival* (TDOA) measurements (3.18),

$$\begin{aligned} z_{\mathrm{u},mlk} &= y_{\mathrm{u},mlk} - y_{\mathrm{u},1lk} \\ &= \|\boldsymbol{r}_m - \boldsymbol{t}_{lk}\|_2 - \|\boldsymbol{r}_1 - \boldsymbol{t}_{lk}\|_2 + \Delta\tau_m - \Delta\tau_1 + e_{\mathrm{u},mlk} - e_{\mathrm{u},1lk}, \end{aligned} \tag{7.2}$$

since the ranges are known and can be eliminated. Slightly more advanced methods use multiple transmitters and estimate the clock-offset taking into account the correlated noise. The downside of this algorithm is that all involved positions have to be specified. The surveying of positions is typically a time-consuming and error-prone process which requires additional equipment. Such an elaborate procedure is only feasible for permanent setups, severely limiting the deployment of a UWB positioning system.

The above procedure can equivalently be formulated as a constrained *maximum likelihood* (ML) problem (2.1). To do so, we first define the parameter vector $\boldsymbol{\theta}$ as

$$\boldsymbol{\theta} = \Big(\{\boldsymbol{t}_l, \{\tau_{lk}\}_{k=1}^{K}\}_{l=1}^{L}, \{\boldsymbol{r}_m, \Delta\tau_m\}_{m=1}^{M}\Big). \tag{7.3}$$

Assuming Gaussian measurement noise, the *probability density function* (PDF) of the measurements is given as (2.2a)

$$p(y_{\mathrm{u},mlk}; \boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi\sigma_{\mathrm{u}}^2}} \exp\Big(-\tfrac{1}{2}\epsilon_{\mathrm{u},mlk}^2(\boldsymbol{\theta})\Big), \tag{7.4}$$

where in correspondence with (3.18) the normalized residuals $\epsilon_{\mathrm{u},mlk}$ are given as

$$\epsilon_{\mathrm{u},mlk}(\boldsymbol{\theta}) = \sigma_{\mathrm{u}}^{-1}\Big(\tau_{lk} + \|\boldsymbol{r}_m - \boldsymbol{t}_{lk}\|_2 + \Delta\tau_m - y_{\mathrm{u},mlk}\Big). \tag{7.5}$$

Using (7.4) and (7.5), the ML problem for estimating $\boldsymbol{\theta}$ becomes

$$\min_{\boldsymbol{\theta}} \quad \frac{1}{2}\sum_{m=1}^{M}\sum_{l=1}^{L}\sum_{k=1}^{K} \epsilon_{\mathrm{u},mlk}^2(\boldsymbol{\theta}) \tag{7.6a}$$

$$\text{s.t.} \quad \boldsymbol{t}_l - \bar{\boldsymbol{t}}_l = \boldsymbol{0}, \quad l = 1, \ldots, L \tag{7.6b}$$

$$\boldsymbol{r}_m - \bar{\boldsymbol{r}}_m = \boldsymbol{0}, \quad m = 1, \ldots, M \tag{7.6c}$$

$$\Delta\tau_1 = 0. \tag{7.6d}$$

The constraints (7.6b) – (7.6d) specify the surveyed positions of the receivers and the transmitters, and without loss of generality the clock of the first receiver is selected to be the central clock. The latter is necessary since a common constant can be added to the clock-offsets and subtracted from the time of transmissions without affecting the normalized residuals. Note that although the problem (7.6) is formulated using a large parameter vector, the constraints (7.6b) and (7.6c) reduce it to finding $\{\tau_{lk}\}$ and $\{\Delta\tau_m\}$.

7.2.2 Proposed calibration algorithm

To arrive at our proposed calibration method, note that the constraints (7.6b) and (7.6c) have been introduced out of convenience since they allow the clock-offset

to be read off directly from (7.2). In fact, they are not essential to solve the problem and can be removed. It is possible to estimate the complete parameter vector $\boldsymbol{\theta}$, including the receiver and transmitter positions, from the dataset $\{y_{\mathrm{u},mlk}\}$ and skip the surveying step completely. Similarly to selecting a central clock, we now have to define the coordinate system in which the positions are expressed. Without loss of generality, we define it using three (non-collinear) receivers. Summarizing the above, we propose to formulate the calibration problem as

$$\min_{\boldsymbol{\theta}} \quad \frac{1}{2}\sum_{m=1}^{M}\sum_{l=1}^{L}\sum_{k=1}^{K}\epsilon_{\mathrm{u},mlk}^2(\boldsymbol{\theta}) \tag{7.7a}$$

$$\text{s.t.} \quad A_m \boldsymbol{r}_m = \mathbf{0}, \quad m = 1,2,3 \tag{7.7b}$$

$$\Delta\tau_1 = 0. \tag{7.7c}$$

The constraint (7.7b) defines the coordinate system using the matrices

$$A_1 = [\boldsymbol{e}_1\ \boldsymbol{e}_2\ \boldsymbol{e}_3]^T, \qquad A_2 = [\boldsymbol{e}_2\ \boldsymbol{e}_3]^T, \qquad A_3 = \boldsymbol{e}_3^T, \tag{7.8}$$

where $\{\boldsymbol{e}_i\}_{i=1}^3$ is the standard basis for $\mathbb{R}^3$. The optimization problem (7.7) is a nonlinear least squares problem with equality constraints. This class of problems and methods for solving it have been discussed in Section 2.3.

Since the proposed calibration method does not require any surveying of positions, there are no restrictions on the size of the dataset and it becomes feasible to collect a dataset with a large number ($L \gg M$) of transmitters. Furthermore, it turns out to be convenient to use only one transmission per transmitter, i.e. $K = 1$. Such a dataset of L stationary transmitters, each transmitting a single pulse, is equivalent to that of a single transmitter transmitting L pulses at different positions. That is, the dataset can be collected by simply moving a single transmitter in the measurement volume. This yields an efficient and simple procedure to collect a large dataset in a short amount of time.

The solvers described in Section 2.3 can efficiently solve (7.7), but require a reasonable initial estimate to converge to the correct optimum since the problem is non-convex. Our approach is to find a starting point for (7.7) by solving a series of optimization problems.

The first step is to collect a dataset using $L = M$ transmitters that are placed in close proximity of the M receivers. This implies that $\boldsymbol{r}_m \approx \boldsymbol{t}_m$, which allows us to extend (7.7) with this additional information and solve

$$\min_{\boldsymbol{\theta}} \quad \frac{1}{2}\sum_{m=1}^{M}\sum_{l=1}^{M}\sum_{k=1}^{K}\epsilon_{\mathrm{u},mlk}^2(\boldsymbol{\theta}) \tag{7.9a}$$

$$\text{s.t.} \quad A_m \boldsymbol{r}_m = \mathbf{0}, \quad m = 1,2,3 \tag{7.9b}$$

$$\Delta\tau_1 = 0 \tag{7.9c}$$

$$\boldsymbol{r}_m - \boldsymbol{t}_m = \mathbf{0}, \quad m = 1,\ldots,M. \tag{7.9d}$$

Solving this problem also requires a starting point, but when started in an arbi-

Algorithm 7.1 UWB calibration

1. Construct a setup consisting of M stationary receivers.
2. Place M transmitters in close proximity to the receiver antennas and collect a dataset $\mathcal{D}_1 = \{y_{\mathrm{u},mlk}\}$.
3. Solve (7.9) for $\mathcal{D}_1$ to obtain $\{\boldsymbol{r}_{m,0}, \Delta\tau_{m,0}\}_{m=1}^{M}$. The optimization is started in
$$\boldsymbol{\theta}_0 = \left(\{\tilde{\boldsymbol{r}}_l, \{0\}_{k=1}^{K}\}_{l=1}^{M}, \{\tilde{\boldsymbol{r}}_m, 0\}_{m=1}^{M}\right),$$
where $\{\tilde{\boldsymbol{r}}_m\}_{m=1}^{M}$ is a noisy, scaled and rotated estimate of the set of receiver positions provided by the user.
4. Collect a second dataset $\mathcal{D}_2 = \{y_{\mathrm{u},mn1}\}$ while moving a single transmitter through the measurement volume.
5. Apply multilateration (3.19) on $\mathcal{D}_2$ using the calibration values of Step 3 to obtain $\{\boldsymbol{t}_{l,0}, \tau_{l,0}\}_{l=1}^{L}$.
6. Solve (7.7) for $\mathcal{D}_2$. The optimization is started in
$$\boldsymbol{\theta}_0 = \left(\{\boldsymbol{t}_{l,0}, \tau_{l,0}\}_{l=1}^{L}, \{\boldsymbol{r}_{m,0}, \Delta\tau_{m,0}\}_{m=1}^{M}\right),$$
using the results from Steps 3 and 5.
7. Calculate the calibration accuracy using (2.39).

trary disjoint receiver configuration, i.e. $\boldsymbol{r}_i \neq \boldsymbol{r}_j$, it converges to the correct configuration or a (partially) mirrored version. To prevent the latter from happening, we start the optimization (7.9) in a user-specified initial receiver configuration: a noisy, rotated and scaled estimate of the set of receiver positions.

The calibration parameters $\{\boldsymbol{r}_m, \Delta\tau_m\}_{m=1}^{M}$ obtained from solving (7.9) are biased, since the constraint (7.9d) only holds approximately. However, they provide a viable starting point for solving (7.7), and they can be used to determine initial values for the transmitter parameters $\{\boldsymbol{t}_l, \tau_l\}_{l=1}^{L}$ using multilateration (3.19). With this approach, a good starting point for (7.7) has been specified and the methods described in Section 2.3 can be used to solve it.

Combining the steps discussed above, the proposed calibration algorithm is obtained. The complete algorithm is summarized in Algorithm 7.1.

7.2.3 Experiments

Algorithm 7.1 has been used to calibrate a number of UWB setups of varying dimensions, number of receivers and receiver configuration. In this and upcoming sections, results from a UWB setup deployed in a relatively large room, approximately $8 \times 6 \times 2.5$ m in size, are presented. In this room an optical reference system is present which provides ground truth positioning data in a relatively small part and allows for a performance evaluation. The UWB setup consists of a total of 10 synchronized receivers, 5 are attached to the ceiling and 5 are placed on the floor. Each receiver has a 1 Hz transmitter integrated in its housing. A 10 Hz transmitter is used to move throughout the measurement volume.

In this section we present the calibration results from a 6 s dataset of the transmitters integrated in the receivers ($\mathcal{D}_1$) and a 50 s dataset where the transmitter

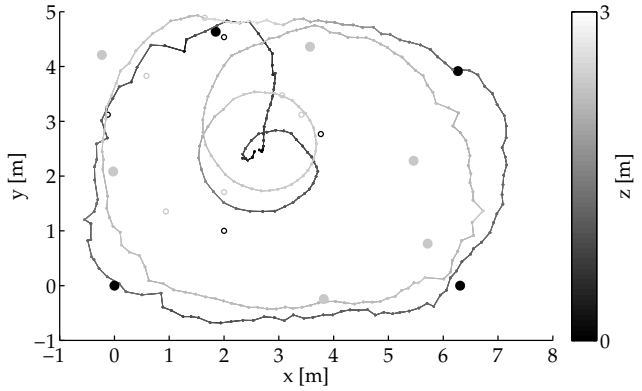


Figure 7.1: *Top view of the receiver locations and the transmitter trajectory of $\mathcal{D}_2$. Shown are the user-specified initial receiver configuration $\tilde{\boldsymbol{r}}_m$ (∘), the calibrated receiver positions $\boldsymbol{r}_m$ (•) and the transmitter trajectory $\boldsymbol{t}_l$ (–).*

is moving through the measurement volume ($\mathcal{D}_2$). The transmitter trajectory is circular with a small and a large radius at two height levels. Figure 7.1 shows the receiver positions before and after calibration using Algorithm 7.1, together with the estimated transmitter trajectory. The user-specified receiver configuration $\tilde{\boldsymbol{r}}_m$ used in Step 3 of Algorithm 7.1 clearly has a wrong scale and is rotated approximately 45°. Nevertheless, it provides a viable starting point for Algorithm 7.1: the transmitter trajectory clearly shows the executed motion and the calibrated positions $\boldsymbol{r}_m$ do agree very well with their surveyed positions.

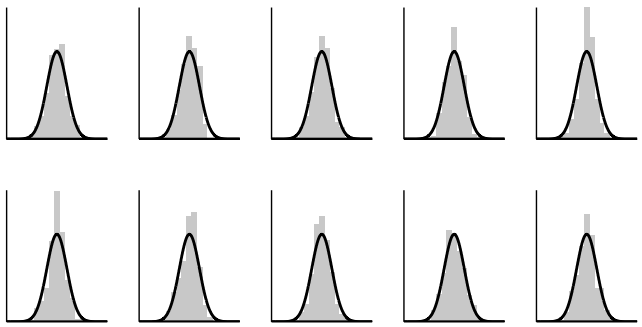


Figure 7.2: *Normalized residuals $\epsilon_{\mathrm{u},mn}$ of $\mathcal{D}_2$ for each of the M receivers after calibration. Both the empirical distribution (bar) and the theoretical distribution (line) are shown.*

In order to further validate the calibration results, the normalized residuals ϵ of the moving transmitter dataset $\mathcal{D}_2$ are studied. Figure 7.2 shows the normalized residuals of $\mathcal{D}_2$ after Step 6 of Algorithm 7.1, i.e. using the calibrated values $\{\boldsymbol{r}_m, \Delta\tau_m\}_{m=1}^{M}$. Notice that the residuals are unbiased and that their distribution

is very close to that of white noise. Hence, the model, with the estimated parameters and its assumptions, appears to be correct. This is in turn a very good indication that correct calibration parameters have been obtained.

Other experiments confirm that the proposed calibration method is an easy-to-use method capable of accurately calibrating a UWB setup within minutes. This is a major speedup compared to previous state-of-the-art calibration methods which rely on the time-consuming and error-prone process of surveying the receiver positions. Hence, the proposed calibration method enables deployment of non-permanent, portable UWB setups, thereby enabling many new application areas.

7.2.4 Sensitivity analysis

The calibration parameters obtained using Algorithm 7.1 are mainly used for multilateration purposes, see Section 3.3.3. Intuitively, the calculated positions will be affected by calibration errors. Figure 7.3a–c contains simulation results showing the influence of the calibration trajectory on the multilateration accuracy. The simulations are made with 8 receivers and a helix shaped calibration trajectory consisting of $L = 100$ transmissions. As shown in Figure 7.3, a better positioning accuracy is obtained with a larger radius of the calibration trajectory. This behavior can be explained using geometric arguments: the position of a transmitter can be estimated more accurately when it is surrounded by receivers. Similarly, the position of the receivers can be estimated more accurately when they are surrounded by transmitters. The simulations in Figure 7.3 show that a severe loss of position accuracy can occur when an inaccurate calibration is used. Therefore, care should be exerted to make sure that the calibration results are of sufficient accuracy. The estimated covariance of the calibration parameters of Step 7 in Algorithm 7.1 can be used for this purpose.

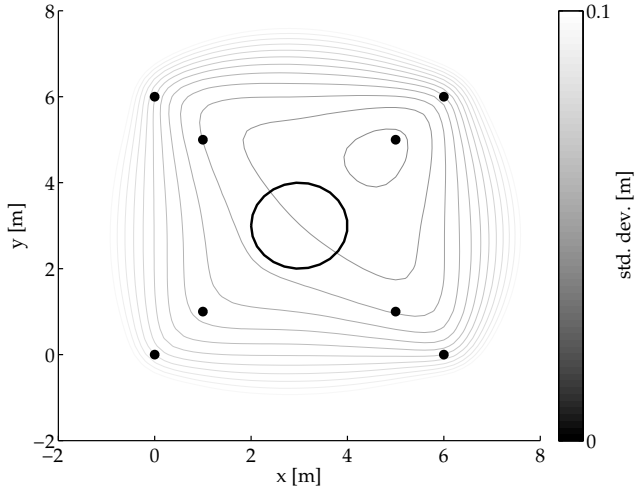
7.3 UWB multilateration

In this section we introduce a novel UWB multilateration method based on *maximum a posteriori* (MAP) estimation. Using an exponential prior for the time delays a triangulation method is obtained which detects and ignores measurements affected by non-zero time delays.

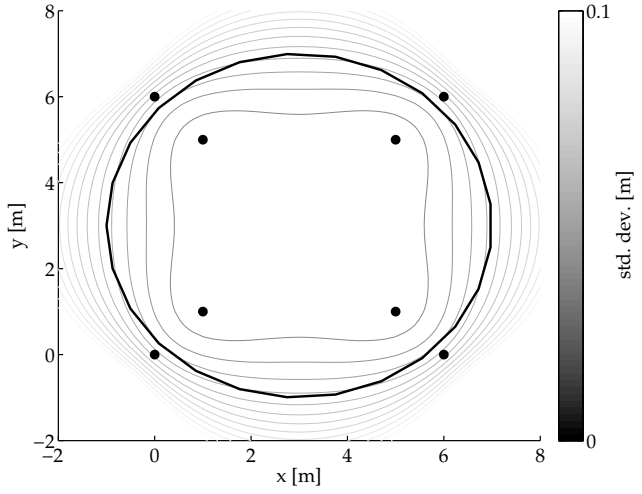
7.3.1 ℓ_1-regularization

The multilateration problem, see Section 3.3.3, can be approached using an ML approach similar to Section 7.2. Besides the position of the transmitter $\boldsymbol{t}$, the quantity of interest, the time of transmission τ needs to be considered in multilateration. Furthermore, the time delays $\delta_{\mathrm{u},m}$ are considered explicitly. That is, we first define the parameter vector $\boldsymbol{\theta}$ as

$$\boldsymbol{\theta} = \left(\boldsymbol{t}, \tau, \{\delta_{\mathrm{u},m}\}_{m=1}^{M}\right). \tag{7.10}$$

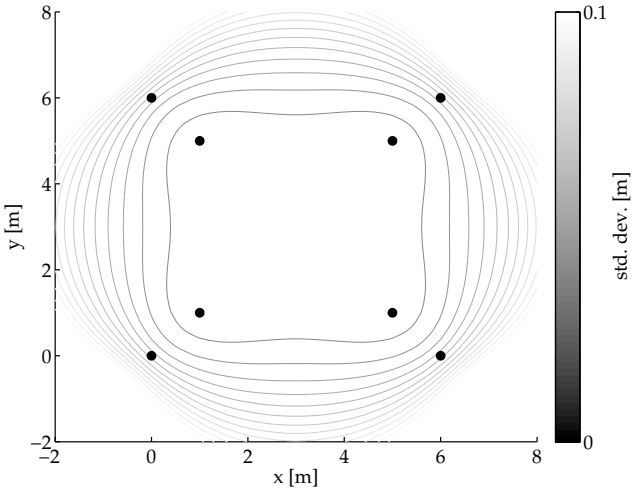


(a) *Calibration using a small trajectory.*



(b) *Calibration using a large trajectory.*

Figure 7.3: *The effect of calibration on UWB multilateration accuracy. Shown are the 3D position accuracy (contour lines), the receiver positions (•) and the calibration trajectory (–) for three calibration scenarios.*



(c) *Perfect calibration.*

Figure 7.3 (continued): *The effect of calibration on UWB multilateration accuracy. Shown are the 3D position accuracy (contour lines), the receiver positions (•) and the calibration trajectory (–) for three calibration scenarios.*

Assuming Gaussian measurement noise, the PDF of the TOA measurements is given as

$$p(y_{\mathrm{u},m};\boldsymbol{\theta}) = \frac{1}{\sqrt{2\pi\sigma_{\mathrm{u}}^2}} \exp\left(-\tfrac{1}{2}\epsilon_{\mathrm{u},m}^2(\boldsymbol{\theta})\right), \tag{7.11}$$

where in correspondence with (3.18) the normalized residuals $\epsilon_{\mathrm{u},m}$ are given as

$$\epsilon_{\mathrm{u},m}(\boldsymbol{\theta}) = \sigma_{\mathrm{u}}^{-1}\Big(\tau + \|\boldsymbol{r}_m - \boldsymbol{t}\|_2 + \Delta\tau_m + \delta_{\mathrm{u},m} - y_{\mathrm{u},m}\Big). \tag{7.12}$$

Using (7.11) and (7.12), the ML problem for estimating $\boldsymbol{\theta}$ becomes

$$\min_{\boldsymbol{\theta}} \quad \frac{1}{2}\sum_{m=1}^{M} \epsilon_{\mathrm{u},m}^2(\boldsymbol{\theta}) \tag{7.13a}$$

$$\text{s.t.} \quad \delta_{\mathrm{u},m} = 0, \quad m = 1,\ldots,M \tag{7.13b}$$

where the standard, but often incorrect assumption of zero time delays is included explicitly as the constraint (7.13b).

As an alternative to assuming zero time delays, it is possible to use an exponential prior for the time delays. This distribution captures a number of important characteristics of the time delays: its samples can only be positive, most probability weight is placed on the value zero, but large values are possible. The PDF of

this exponential prior is given by

$$p(\delta_{\mathrm{u},m}) = \begin{cases} \lambda \exp(-\lambda \delta_{\mathrm{u},m}), & \delta_{\mathrm{u},m} \geq 0 \\ 0, & \delta_{\mathrm{u},m} < 0. \end{cases} \tag{7.14}$$

Using (7.11), (7.12) and (7.14), the MAP problem (2.5) for estimating $\boldsymbol{\theta}$ now becomes

$$\min_{\boldsymbol{\theta}} \quad \frac{1}{2} \sum_{m=1}^{M} \epsilon_{\mathrm{u},m}^2(\boldsymbol{\theta}) + \lambda \sum_{m=1}^{M} \delta_{\mathrm{u},m} \tag{7.15a}$$

$$\text{s.t.} \quad \delta_{\mathrm{u},m} \geq 0 \quad m = 1, \ldots, M \tag{7.15b}$$

The MAP problem (7.15) can be classified as an ℓ_1-regularized least squares problem (Kim et al., 2007). It contains an additional regularization term in its objective function and the equality constraint is replaced with a non-negative constraint compared to the ML problem (7.13). ℓ_1-regularization has the property that it typically yields sparse parameters vectors, i.e., it tries to make as few elements as possible nonzero. This property explains the recent attention for the methods in the upcoming field of compressed sensing (Candès et al., 2006; Donoho, 2006). It is also exactly the desired behavior to deal with time delays in multilateration: the time delays are by default pushed toward zero, however, when inconsistencies occur there is freedom to assign a few nonzero time delays. In the latter case, the corresponding TOA measurements are effectively removed from the problem.

7.3.2 Experiments

To evaluate the proposed multilateration system it has been used to track the motion of a test-subject walking in an indoor environment. The experiments are performed with the setup used in Section 7.2. That is, we use a UWB setup consisting of 10 receivers deployed in a room of approximately $8 \times 6 \times 2.5$ m in size. The setup has been calibrated using Algorithm 7.1. The sensor unit provides 10 Hz UWB transmissions.

In this section we present result of a 25 s trial, where the test-subject is walking a circular path. The sensor unit has been attached to the foot of a test-subject, a position with relatively high dynamics. Regular occurring NLOS conditions due to occlusion by the body — a medium with a reduced speed of light — as well as multipath effects from signals reflected by the floor result in difficulties while determining position when outliers are accounted for. Figure 7.4 shows a comparison of the estimates to an optical reference. It can be concluded that (7.15) yields accurate results. Figure 7.5 shows a histogram of estimated time delays δ_k. Note that a significant part is non-zero, which shows that occlusions and multipath effects are indeed present in the dataset. Furthermore, the distribution does resemble the assumed exponential distribution.

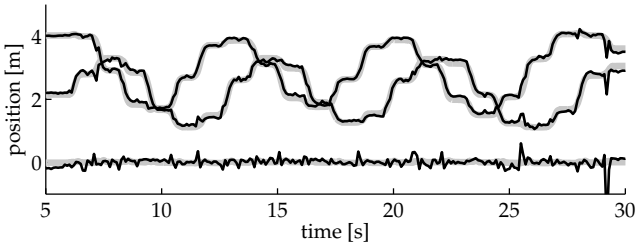


Figure 7.4: *Estimated position of the sensor unit $\boldsymbol{b}^n$. Shown is the estimated trajectory (–) together with reference from an optical system (–).*

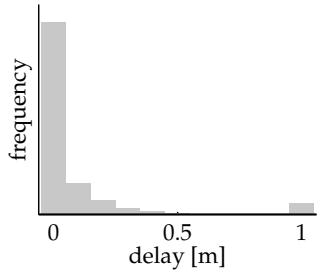


Figure 7.5: *Histogram of estimated time delays.*

7.4 Pose estimation

In this section we discuss a 6 DOF tracking system, estimating both position and orientation, based on tightly coupled sensor fusion of UWB and inertial sensors.

7.4.1 Sensor fusion

Our setup is based on a commercially available UWB asset tracking system. However, instead of working with a UWB transmitter only, we integrated it with an IMU in a single unit, shown in Figure 7.6. The devices are synchronized at hardware level, significantly simplifying the signal processing.

To the best of the author's knowledge there are only a few reports in the literature on combining UWB and inertial sensors. The more recent contributions include a hybrid *two dimensional* (2D) positioning tracking algorithm (Sczyslo et al., 2008) and an *extended Kalman filter* (EKF) for pedestrian tracking (Pittet et al., 2008). Both approaches are loosely coupled and only estimate a limited number of DOF. By a loosely coupled approach we refer to a solution where the measurements from one or several of the individual sensors are pre-processed before they are used to compute the final result. A tightly coupled approach on the other hand refers to an approach where all the measurements are used directly to compute the final result. In this section we propose a full 6 DOF tracker, estimating both the position $\boldsymbol{b}^n$ and the orientation q^{nb} of the sensor unit based on

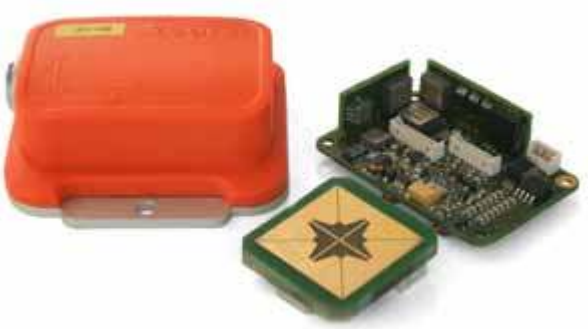


Figure 7.6: *An xsens prototype sensor unit, integrating an IMU and an UWB transmitter into a single housing.*

tightly coupled sensor fusion of UWB and inertial sensors. Tightly coupled sensor

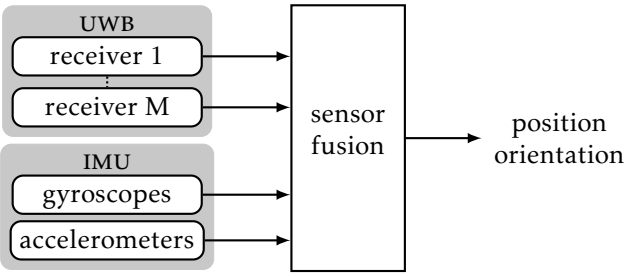


Figure 7.7: *Tightly coupled sensor fusion. The 'raw' measurements from the UWB receivers and the IMU are directly used for sensor fusion.*

fusion, illustrated by Figure 7.7, implies that the 'raw' accelerometer, gyroscope and TOA measurements are directly used for sensor fusion, instead of already filtered output quantities like position or acceleration. Hence, there is no explicit trilateration module as typically found in (loosely coupled) UWB positioning systems. Instead, the trilateration of position is implicitly performed by the sensor fusion algorithm.

The advantages of using a tightly coupled approach are two-fold: Firstly, preprocessing of measurements typically results in a loss of information. This is mainly due to approximations of statistical distributions, but in extreme cases measurements are ignored, for instance when there are not enough TOA measurements for trilateration. By directly using the sensor measurements nothing has to be disregarded and maximal advantage is taken of the available information. Secondly, tightly coupled sensor fusion can perform hypothesis testing for the individual sensors and efficiently deal with outliers. This is especially useful for UWB measurements, where outliers occur regularly due to multipath effects

and/or NLOS conditions. Tightly coupled sensor fusion can disregard the affected measurements while still utilizing the remaining ones. Hence, a tightly coupled system is more robust.

As discussed in Chapter 2 state-space models are one of the key ingredients of sensor fusion. For the sensor unit introduced earlier this section, such a model can be constructed from the models in Chapter 3 and Chapter 4. They are repeated below for the reader's convenience.

As in Section 3.3.1, the TOA measurements of the UWB system are modeled as

$$y_{\mathrm{u},m,t} = \tau_t + \|\boldsymbol{r}_m^n - \boldsymbol{t}_t^n\|_2 + \Delta\tau_m + \delta_{\mathrm{u},m,t} + e_{\mathrm{u},m,t}, \tag{7.16}$$

where τ is the time of transmission of the pulse, $\boldsymbol{t}$ is the position of the transmitter, $\boldsymbol{r}_m$ is the position of the m-th receiver and $\Delta\tau_m$ is the clock-offset of the m-th receiver. Furthermore, $\delta_{\mathrm{u},m} \geq 0$ is a possibly nonzero delay due to NLOS or multipath and $e_{\mathrm{u},m}$ is i.i.d. Gaussian noise. For pose estimation, we assume $\delta_{\mathrm{u},m} = 0$.

Following Section 4.3, we model the position and orientation of the IMU using the inertial measurements,

$$\boldsymbol{b}_{t+1}^n = \boldsymbol{b}_t^n + T\dot{\boldsymbol{b}}_t^n + \tfrac{T^2}{2}\ddot{\boldsymbol{b}}_t^n, \tag{7.17a}$$

$$\dot{\boldsymbol{b}}_{t+1}^n = \dot{\boldsymbol{b}}_t^n + T\ddot{\boldsymbol{b}}_t^n, \tag{7.17b}$$

$$q_{t+1}^{nb} = q_t^{nb} \odot \exp(\tfrac{T}{2}\boldsymbol{\omega}_{nb,t}^b), \tag{7.17c}$$

where $\boldsymbol{b}^n$ and $\dot{\boldsymbol{b}}^n$ denote the position and velocity of the body frame (b-frame) resolved in the navigation frame (n-frame), q^{nb} is a unit quaternion describing the orientation of the n-frame relative to the b-frame and T denotes the sampling interval. The acceleration $\ddot{\boldsymbol{b}}^n$ and the angular velocity $\boldsymbol{\omega}_{nb}^b$ are calculated from the accelerometer measurements $\boldsymbol{y}_{\mathrm{a}}$ and the gyroscope measurements $\boldsymbol{y}_\omega$ using (3.2) and (3.3),

$$\ddot{\boldsymbol{b}}_t^n = R_t^{nb}\left(\boldsymbol{y}_{\mathrm{a},t} - \boldsymbol{\delta}_{\mathrm{a},t}^b - \boldsymbol{e}_{\mathrm{a},t}^b\right) - 2\boldsymbol{\omega}_{ie}^n \times \dot{\boldsymbol{b}}_t^n + \boldsymbol{g}^n, \tag{7.18a}$$

$$\boldsymbol{\omega}_{nb,t}^b = \boldsymbol{y}_{\omega,t} - R_t^{bn}\boldsymbol{\omega}_{ie}^n - \boldsymbol{\delta}_{\omega,t}^b - \boldsymbol{e}_{\omega,t}^b. \tag{7.18b}$$

The inertial bias terms $\boldsymbol{\delta}_{\mathrm{a}}^b$ and $\boldsymbol{\delta}_{\omega}^b$ are slowly time-varying. Hence, they are modeled as random walks,

$$\boldsymbol{\delta}_{\mathrm{a},t+1}^b = \boldsymbol{\delta}_{\mathrm{a},t}^b + \boldsymbol{e}_{\delta_{\mathrm{a}},t}^b, \tag{7.19a}$$

$$\boldsymbol{\delta}_{\omega,t+1}^b = \boldsymbol{\delta}_{\omega,t}^b + \boldsymbol{e}_{\delta_{\omega},t}^b, \tag{7.19b}$$

where $\boldsymbol{e}_{\delta_{\mathrm{a}}}^b$ and $\boldsymbol{e}_{\delta_{\omega}}^b$ are i.i.d. Gaussian noises.

The transmitter sends in regular intervals. Taking into account the clock drift and the clock jitter, the time of transmission τ is modeled as an integrated random

walk

$$\tau_{t+1} = \tau_t + T\dot{\tau}_t + e_{\tau,t}, \tag{7.20a}$$
$$\dot{\tau}_{t+1} = \dot{\tau}_t + e_{\dot{\tau},t}, \tag{7.20b}$$

where e_τ and $e_{\dot{\tau}}$ are i.i.d.. Gaussian noises.

The TOA measurement model (7.16) uses the transmitter position $\boldsymbol{t}^n$. However, the sensor unit motion (7.17) is modeled using the pose of IMU, q^{nb}, $\boldsymbol{b}^n$. Since the transmitter is rigidly attached to the IMU, the expression

$$\boldsymbol{t}^n = \boldsymbol{b}^n + R^{nb}\boldsymbol{t}^b, \tag{7.21}$$

can be used to calculate $\boldsymbol{t}^n$.

Combining (7.16)–(7.21) we obtain a discrete-time nonlinear state-space model whose state vector is, with slight abuse of notation, given by

$$\boldsymbol{x} = \left(\boldsymbol{b}^n, \dot{\boldsymbol{b}}^n, q^{nb}, \delta_{\mathrm{a}}^b, \delta_{\omega}^b, \tau, \dot{\tau}\right)^T. \tag{7.22}$$

It is used in an EKF to fuse the TOA and the inertial measurements, see Section 2.2. The EKF handles the different sample rates and a varying number of measurements straightforwardly. It runs at the high data rate of the IMU and the UWB updates are only performed when measurements are available. Outliers from NLOS and/or multipath effects violate the assumption of $\delta_{\mathrm{u}} = 0$. They can be detected using hypothesis testing on the residuals/innovations of the EKF, see Example 2.3. Since the IMU gives very accurate short-term predictions of the transmitter position, it is relatively easy to detect non-zero delays in the TOA measurements and ignore the affected ones.

The above discussion is summarized in Algorithm 7.2.

Algorithm 7.2 Sensor fusion using Inertial and UWB measurements

1. Perform an initialization and set $p(\boldsymbol{x}_0)$.
2. Do a time update. Propagate $p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1})$ to $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})$ using the process model (7.17), (7.18), (7.19) and (7.20) with the inertial measurements as input signals.
3. If new UWB measurements are available, do a measurement update with outlier detection. Use the measurements in combination with the measurement model (7.16) and (7.21) to correct $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})$ and obtain $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$.
4. Set $t := t + 1$ and iterate from Step 2.

7.4.2 Experiments

To evaluate the proposed tracking system it has been used to track the motion of a test-subject walking in an indoor environment. The experiments are performed with the setup used in Section 7.2 and Section 7.3. That is, we use a UWB setup consisting of 10 receivers deployed in a room of approximately $8 \times 6 \times 2.5$ m in size. The setup has been calibrated using Algorithm 7.1. The sensor unit provides

120 Hz inertial measurements and 10 Hz UWB transmissions.

In this section we present result of a 35 s trial, where test-subject is walking a circular path. The sensor unit has been attached to the head of a test-subject for optimal visibility of the receivers. Figure 7.8 shows an overview of the experiment. The circular path is clearly recognizable. It only occupies a small part of the measurement volume of the UWB/IMU tracking system so that a performance comparison with an optical reference system is possible. Figures 7.9, 7.10 and 7.11 show the estimated position, velocity and orientation of the sensor unit. Comparing the estimates to the optical reference it can be concluded that the system provides drift free and accurate estimates for all quantities at a high output frequency. In fact, the comparison shows 5 cm *root mean square error* (RMSE) for position and 1° RMSE for orientation, see Table 7.1.

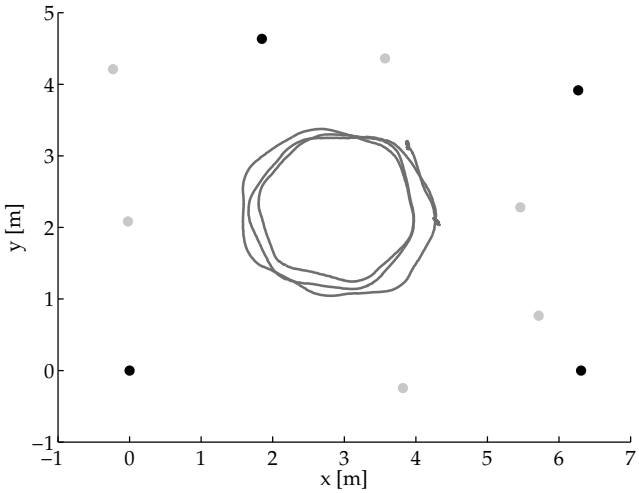


Figure 7.8: *Top view of the trial where the test-subject walked a circular trajectory. Shown are the positions of the low, respectively the high receivers* $\boldsymbol{r}^n_m$ (●, ●) *and the estimated trajectory of the sensor unit* $\boldsymbol{b}^n$ *(–).*

Table 7.1: *RMSE for position and orientation estimates.*

		x	y	z
position	m	0.05	0.04	0.03
orientation	°	0.65	0.46	0.85

Experiments reported in Hol et al. (2009) show the performance of the tracking system when the sensor unit is mounted on the foot. This is a more challenging environment for UWB with regular occurring NLOS conditions due to occlusion by the body — a medium with a reduced speed of light — as well as multipath effects from signals reflected by the floor. Because of efficient outlier rejection,

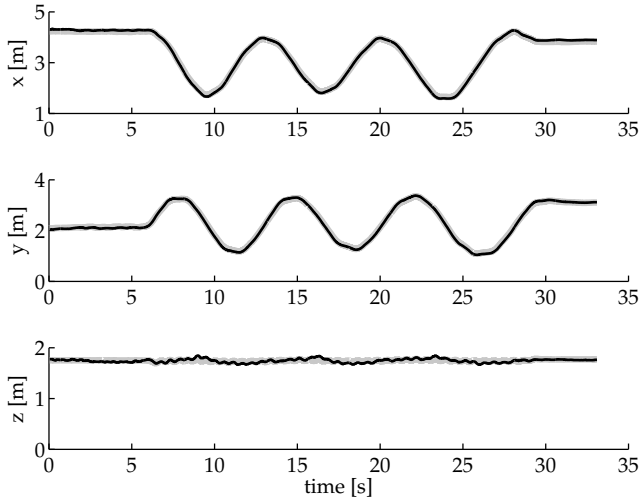


Figure 7.9: *Estimated position of the sensor unit $\mathbf{b}^n$. Shown is the estimated trajectory (–) together with reference from an optical system (–).*

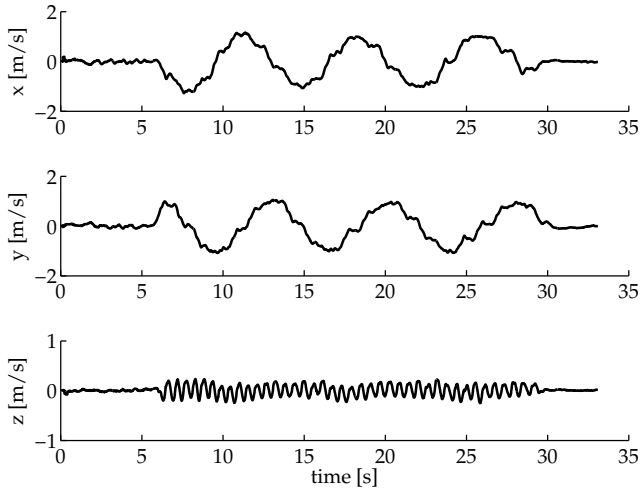


Figure 7.10: *Estimated velocity of the sensor unit $\dot{\mathbf{b}}^n$.*

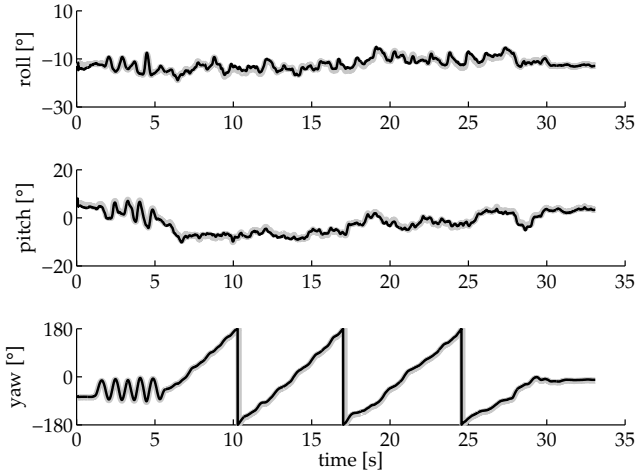


Figure 7.11: *Estimated orientation of the sensor unit q^{nb} expressed in Euler angles. Shown is the estimated trajectory (–) together with reference from an optical system (–).*

the affected TOA measurements are ignored and a robust and accurate tracking solution is obtained.

The above discussion has shown that the UWB/IMU tracking system performs very well on a realistic indoor tracking scenario. This performance cannot exclusively be attributed to effective sensor fusion; accurate calibration of the UWB system as well as robust UWB multilateration are of equal importance for the final result.

8

Inertial and GPS

The combination of the *global positioning system* (GPS) with inertial sensors has become standard for many outdoor applications. This chapter discusses the sensor fusion problem for single as well as dual receiver configurations.

8.1 Problem formulation

Many outdoor applications require both real-time position and orientation estimates. Examples include navigation of autonomous vehicles for aerial, ground or marine applications where the sensors are used to determine the position and orientation of the platform. This is in turn used in the control loop to stabilize the platform and to make sure that it follows a predetermined path.

GPS is an excellent source for positioning information. It provides reasonable accuracy at low cost and is almost always available. Combining GPS with inertial sensors not only improves the position estimates, but also enables estimation of orientation. Furthermore, these estimates are available at high sampling rates. The position estimates from GPS are used to stabilize the inertial solution, whereas the inertial sensors are used to bridge short periods with bad or no GPS reception due to for instance shadowing or multipath.

For heading to become observable, some excitation with linear acceleration is required. Typically this is no problem, as velocity changes, in magnitude or direction, provide sufficient acceleration. However, during longer periods with constant acceleration heading drift might occur. In an automotive setting, driving on a straight highway is an example of a situation where this problem occurs. To prevent this, a magnetometer or a second GPS receiver can be used. The sensor fusion problems for both cases will be discussed.

8.2 Single receiver pose estimation

In this section we discuss a 6 *degrees of freedom* (DOF) tracking system, estimating both position and orientation, based on loosely coupled sensor fusion of a single GPS receiver and inertial sensors.

8.2.1 Sensor fusion

Sensor fusion of inertial and GPS measurements is a well known topic in literature (Kaplan and Hegarty, 2006; Brown and Hwang, 1996). Both loosely coupled as well as tightly coupled approaches have been proposed. In this section we discuss loosely coupled sensor fusion for a small, low-cost sensor unit integrating *micromachined electromechanical system* (MEMS) inertial sensors with a single band GPS receiver in a single package, shown in Figure 8.1. The devices are synchro-

Figure 8.1: *An Xsens MTi-G, integrating an IMU and a L1 GPS receiver into a single housing. A barometer is included as well.*

nized at hardware level and have a common clock, which significantly simplifies the signal processing.

Although there are potential benefits for tight coupling in combination with high accuracy applications, loosely coupled sensor fusion of GPS and inertial sensors already gives very good performance. In this approach, illustrated in Figure 8.2, the GPS pseudorange and Doppler measurements are used to calculate a *position, velocity and time* (PVT) solution. The obtained position and velocity are then fused with the inertial measurements.

As discussed in Chapter 2, state-space models are one of the key ingredients of sensor fusion. For the sensor unit introduced earlier in this section, such a model can be constructed from the models in Chapter 3 and Chapter 4. They are repeated below for the reader's convenience.

As discussed in Section 3.4.3, a GPS receiver can provide a PVT solution. In a

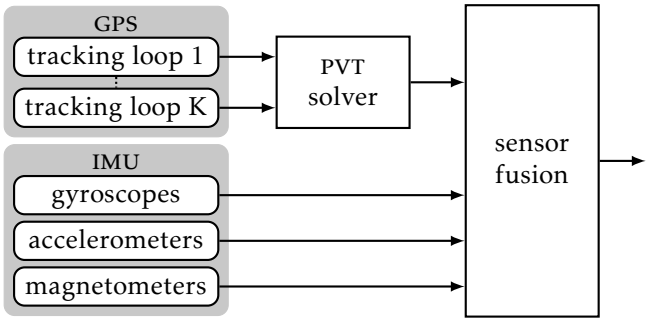


Figure 8.2: *Loosely coupled sensor fusion. The 'raw' GPS measurements from the satellites are used to solve for position and velocity before using them with the IMU measurement for sensor fusion.*

loosely coupled approach, this solution is used as a measurement. That is, the GPS measurements are modeled as

$$\boldsymbol{y}_{\mathrm{p},t} = \boldsymbol{r}_t^n + \boldsymbol{e}_{\mathrm{p},t}^n, \tag{8.1a}$$

$$\boldsymbol{y}_{\mathrm{v},t} = \dot{\boldsymbol{r}}_t^n + \boldsymbol{e}_{\mathrm{v},t}^n, \tag{8.1b}$$

where $\boldsymbol{r}$ and $\dot{\boldsymbol{r}}$ are the receiver position and velocity, $\boldsymbol{e}_{\mathrm{p}}$ and $\boldsymbol{e}_{\mathrm{v}}$ are *independently and identically distributed* (i.i.d.) Gaussian noises. Note that the measurements have been expressed in the navigation frame (n-frame) which is defined to be stationary *with respect to* (w.r.t.) the earth frame (e-frame).

Assuming a magnetically undistorted environment, the magnetometer measurement model (3.4) becomes

$$\boldsymbol{y}_{\mathrm{m},t} = R_t^{bn} \boldsymbol{m}_{\mathrm{e}}^n + \boldsymbol{e}_{\mathrm{m},t}^b, \tag{8.2}$$

where $\boldsymbol{m}_{\mathrm{e}}$ is the earth magnetic field vector and $\boldsymbol{e}_{\mathrm{m}}$ is i.i.d. Gaussian noise. In a typical outdoor environments, the only persistent magnetic distortions are local to the sensor and can be compensated for with a magnetic field calibration. Furthermore, the earth magnetic field vector can be retrieved from a map when the position of the *inertial measurement unit* (IMU) is known.

Following Section 4.3, we model the position and orientation of the IMU using the inertial measurements,

$$\boldsymbol{b}_{t+1}^n = \boldsymbol{b}_t^n + T\dot{\boldsymbol{b}}_t^n + \tfrac{T^2}{2}\ddot{\boldsymbol{b}}_t^n, \tag{8.3a}$$

$$\dot{\boldsymbol{b}}_{t+1}^n = \dot{\boldsymbol{b}}_t^n + T\ddot{\boldsymbol{b}}_t^n, \tag{8.3b}$$

$$q_{t+1}^{nb} = q_t^{nb} \odot \exp(\tfrac{T}{2}\boldsymbol{\omega}_{nb,t}^b), \tag{8.3c}$$

where $\boldsymbol{b}^n$ and $\dot{\boldsymbol{b}}^n$ denote the position and velocity of the body frame (b-frame) resolved in the n-frame, q^{nb} is a unit quaternion describing the orientation of the n-frame relative to the b-frame and T denotes the sampling interval. The acceleration $\ddot{\boldsymbol{b}}^n$ and the angular velocity $\boldsymbol{\omega}_{nb}^b$ are calculated from the accelerometer

measurements $\boldsymbol{y}_{\mathrm{a}}$ and the gyroscope measurements $\boldsymbol{y}_{\omega}$ using (3.2) and (3.3),

$$\ddot{\boldsymbol{b}}_t^n = R_t^{nb}\left(\boldsymbol{y}_{\mathrm{a},t} - \boldsymbol{\delta}_{\mathrm{a},t}^b - \boldsymbol{e}_{\mathrm{a},t}^b\right) - 2\boldsymbol{\omega}_{ie}^n \times \dot{\boldsymbol{b}}_t^n + \boldsymbol{g}^n, \tag{8.4a}$$

$$\boldsymbol{\omega}_{nb,t}^b = \boldsymbol{y}_{\omega,t} - R_t^{bn}\boldsymbol{\omega}_{ie}^n - \boldsymbol{\delta}_{\omega,t}^b - \boldsymbol{e}_{\omega,t}^b. \tag{8.4b}$$

The inertial bias terms $\boldsymbol{\delta}_{\mathrm{a}}^b$ and $\boldsymbol{\delta}_{\omega}^b$ are slowly time-varying. Hence, they are modeled as random walks,

$$\boldsymbol{\delta}_{\mathrm{a},t+1}^b = \boldsymbol{\delta}_{\mathrm{a},t}^b + \boldsymbol{e}_{\delta_{\mathrm{a}},t}^b, \tag{8.5a}$$

$$\boldsymbol{\delta}_{\omega,t+1}^b = \boldsymbol{\delta}_{\omega,t}^b + \boldsymbol{e}_{\delta_{\omega},t}^b, \tag{8.5b}$$

where $\boldsymbol{e}_{\delta_{\mathrm{a}}}^b$ and $\boldsymbol{e}_{\delta_{\omega}}^b$ are i.i.d. Gaussian noises.

The GPS measurement model (8.1) uses the receiver position $\boldsymbol{r}^n$ and the receiver velocity $\dot{\boldsymbol{r}}^n$. However, the sensor unit motion (8.3) is modeled using the pose of the IMU, q^{nb} and $\boldsymbol{b}^n$. Since the receiver is rigidly attached to the IMU, the expressions

$$\boldsymbol{r}_t^n = \boldsymbol{b}_t^n + R_t^{nb}\boldsymbol{r}^b, \qquad \dot{\boldsymbol{r}}_t^n = \dot{\boldsymbol{b}}_t^n + R_t^{nb}(\boldsymbol{\omega}_{nb,t}^b \times \boldsymbol{r}^b), \tag{8.6}$$

can be used to calculate $\boldsymbol{r}^n$ and $\dot{\boldsymbol{r}}^n$.

Combining (8.1)–(8.6) we obtain a discrete-time nonlinear state-space model. Its state vector is, with slight abuse of notation, given by

$$\boldsymbol{x} = \left(\boldsymbol{b}^n, \dot{\boldsymbol{b}}^n, q^{nb}, \boldsymbol{\delta}_{\mathrm{a}}^b, \boldsymbol{\delta}_{\omega}^b\right)^T. \tag{8.7}$$

The model is used in an *extended Kalman filter* (EKF) to fuse the GPS and the inertial measurements, see Section 2.2. The EKF handles the different sample rates straightforwardly. It runs at the high data rate of the IMU and the GPS updates are only performed when measurements are available. In navigation applications, the traveled distance might become large enough to motivate a redefinition of the n-frame. This ensures that the n-frame remains aligned to the local tangent plane with unit vectors pointing north, west and up. This redefinition is achieved with the state transformation

$$\tilde{\boldsymbol{x}} = \begin{pmatrix} \boldsymbol{b}^{\tilde{n}} \\ \dot{\boldsymbol{b}}^{\tilde{n}} \\ q^{\tilde{n}b} \\ \boldsymbol{\delta}_{\mathrm{a}}^b \\ \boldsymbol{\delta}_{\omega}^b \end{pmatrix} = \begin{pmatrix} R^{\tilde{n}n}\boldsymbol{b}^n + \boldsymbol{n}^{\tilde{n}} \\ R^{\tilde{n}n}\dot{\boldsymbol{b}}^n \\ q^{\tilde{n}n} \odot q^{nb} \\ \boldsymbol{\delta}_{\mathrm{a}}^b \\ \boldsymbol{\delta}_{\omega}^b \end{pmatrix} = f(\boldsymbol{x}), \tag{8.8}$$

where the transformation f and the redefined frame $\tilde{n}$ are chosen such that the position estimate in the redefined frame becomes zero, i.e., $\hat{\boldsymbol{b}}^{\tilde{n}} = \boldsymbol{0}$.

The above discussion is summarized in Algorithm 8.1.

8.2.2 Experiments

The GPS/IMU combination has been used to track an Aero L-29 Delfin jet aircraft, see Figure 8.3. In this section we present results from a flight of approximately

Algorithm 8.1 Sensor fusion using inertial and GPS measurements

1. Perform an initialization and set $p(\boldsymbol{x}_0)$.
2. Do a time update. Propagate $p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1})$ to $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})$ using the process model (8.3), (8.4) and (8.5) with the inertial measurements as input signals.
3. Do a measurement update with the magnetometer measurements and, when available, the GPS measurement. Use the measurements in combination with the measurement models (8.1), (8.2) and (8.6) to correct $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})$ and obtain $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$.
4. When the estimated position becomes too large, redefine the n-frame. Use the state transformation (8.8) to obtain $p(\tilde{\boldsymbol{x}}_t|\boldsymbol{y}_{1:t})$ and set $\boldsymbol{x}_t := \tilde{\boldsymbol{x}}_t$.
5. Set $t := t + 1$ and iterate from Step 2.

Figure 8.3: *The Aero L-29 Delphin jet used to evaluate the GPS/inertial sensor unit.*

35 min in which the airplane performed several maneuvers, including banked curves and barrel rolls as well as periods with constant velocity. The airplane is equipped with a Honeywell HG 1150 IMU and a high grade, dual band GPS receiver providing a very accurate reference.

During periods of constant velocity, the heading of the airplane is not observable using GPS only. To prevent this, the magnetometers in the IMU are used. However, since the airplane is constructed from metal a magnetic field calibration, see Section 5.4, is essential.

Figure 8.4 shows an overview of the experiment. It shows the trajectory of the aircraft performing a number of dynamic maneuvers. Figures 8.5, 8.6 and 8.7 show the estimated position, velocity and orientation of the sensor unit obtained using Algorithm 8.1 together with their references. The magnetometers are used with and without a magnetic field calibration. The differences are not visible on the scale of Figures 8.5 and 8.6, but do show in Figure 8.7. It can be concluded that the system provides drift free and accurate estimates for all quantities at a

high output frequency when using a properly calibrated magnetometer. In fact, the numerical comparison in Table 8.1 shows less than 2° *root mean square error* (RMSE) for orientation in that case. The position and velocity estimates are as accurate as can be expected when using a single band GPS receiver.

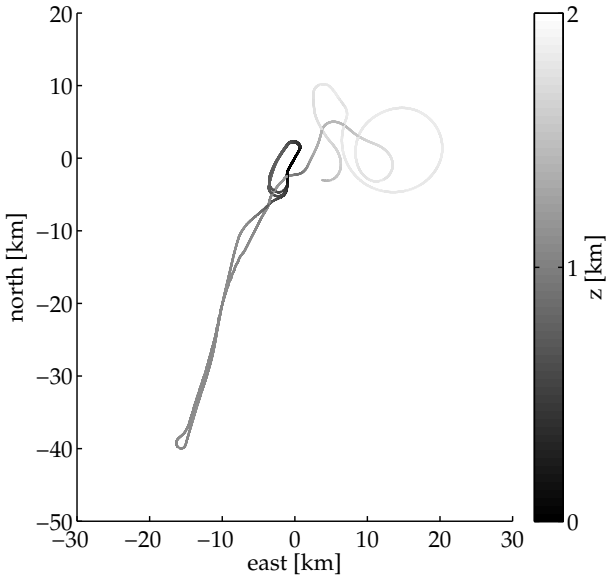


Figure 8.4: *Top view of the trajectory of the airplane.*

Table 8.1: *RMSE for position, velocity and orientation estimates with and without magnetic field calibration.*

		not calibrated			calibrated		
		x	y	z	x	y	z
position	m	6.1	5.9	21.6	6.1	5.3	18.7
velocity	m/s	4.6	4.5	3.7	4.4	3.9	3.5
orientation	°	7.3	6.9	43.2	0.9	0.9	1.8

The above discussion shows that loosely coupled sensor fusion of GPS and inertial measurements provides accurate tracking of position and orientation, even under dynamic situations. A magnetic field calibration is essential to obtain good results. Without it performance, especially in orientation, is decreased significantly.

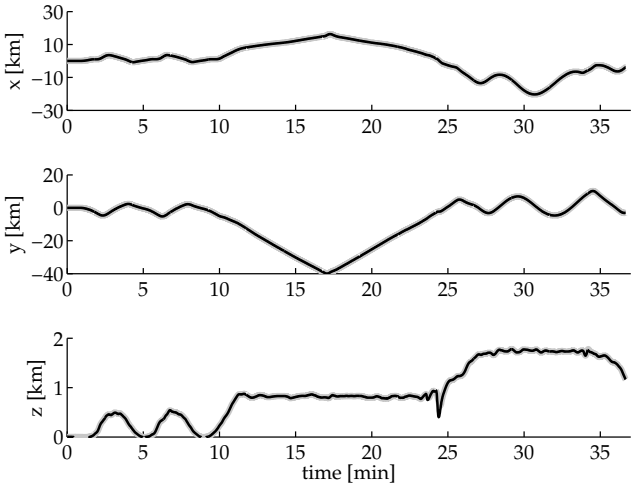


Figure 8.5: *Estimated position of the sensor unit $\boldsymbol{b}^n$. Shown is the estimated trajectory (–) together with the dual band GPS reference (–).*

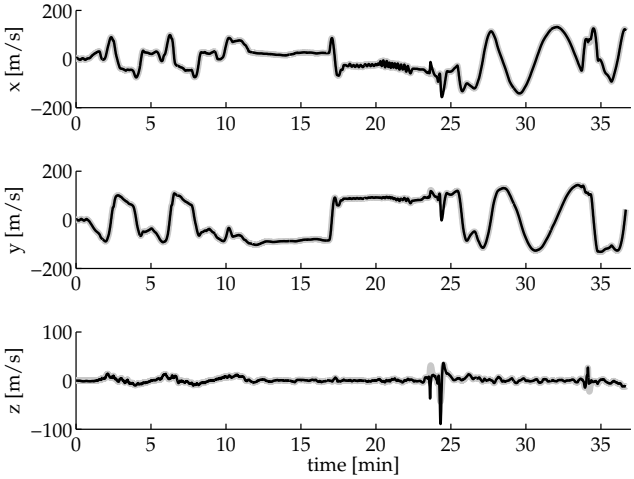


Figure 8.6: *Estimated velocity of the sensor unit $\dot{\boldsymbol{b}}^n$. Shown is the estimated trajectory (–) together with the dual band GPS reference (–).*

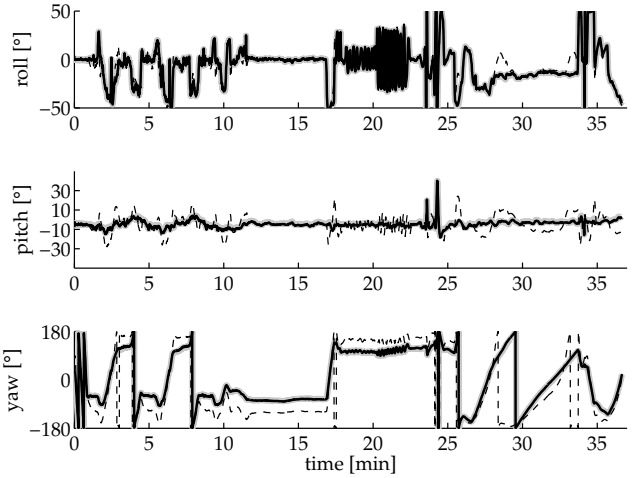


Figure 8.7: *Estimated orientation of the sensor unit q^{nb} expressed in Euler angles. Shown is the estimated trajectory before (- -) and after (–) magnetic calibration together with the navigation grade inertial reference (–).*

8.3 Dual receiver pose estimation

In this section we discuss a 6 DOF tracking system, estimating both position and orientation, based on sensor fusion of a pair of GPS receivers and inertial sensors.

8.3.1 Sensor fusion

A single GPS receiver configuration requires the use of a magnetometer to make heading observable under low acceleration conditions. Adding a second GPS receiver to the platform is an alternative method to achieve this. As discussed in Section 3.4, a setup with two receivers can be used to obtain double difference carrier phase measurements. When both receivers are rigidly attached to the platform, the baseline vector measured by the double difference measurements is directly dependent on the orientation of the platform and heading becomes observable.

For a dual receiver configuration, both receivers can provide a position and velocity solution, although the pseudorange and Doppler measurements can also be used directly. Similar to (8.1), the GPS measurements are modeled as

$$\boldsymbol{y}_{\mathrm{p},a,t} = \boldsymbol{r}^n_{a,t} + \boldsymbol{e}_{\mathrm{p},a,t}, \qquad \boldsymbol{y}_{\mathrm{p},b,t} = \boldsymbol{r}^n_{b,t} + \boldsymbol{e}_{\mathrm{p},b,t}, \tag{8.9a}$$

$$\boldsymbol{y}_{\mathrm{v},a,t} = \dot{\boldsymbol{r}}^n_{a,t} + \boldsymbol{e}_{\mathrm{v},a,t}, \qquad \boldsymbol{y}_{\mathrm{v},b,t} = \dot{\boldsymbol{r}}^n_{b,t} + \boldsymbol{e}_{\mathrm{v},b,t}, \tag{8.9b}$$

where $\boldsymbol{r}_a$ and $\boldsymbol{r}_b$ are the positions of receiver a and receiver b, $\boldsymbol{e}_{\mathrm{p}}$ and $\boldsymbol{e}_{\mathrm{v}}$ are i.i.d. Gaussian noises. Note that the measurements have been expressed in the n-frame which is defined to be stationary w.r.t. the e-frame.

As in Section 3.4.1, the double difference carrier phase measurement from the satellites k and l are modeled as

$$y_{\phi,t} = (\boldsymbol{u}_t^e)^T(\boldsymbol{r}_{a,t}^e - \boldsymbol{r}_{b,t}^e) + \lambda N_{kl} + e_{\phi,t}, \tag{8.10}$$

where $\boldsymbol{u}_{kl} = \boldsymbol{u}_k - \boldsymbol{u}_l$ is the difference of unit vectors from the receiver to the satellite, $\boldsymbol{r}$ is a receiver position, λ is the wavelength, $N_{kl} \in \mathbb{Z}$ is an integer number of carrier cycles and $e_{\phi,kl}$ is i.i.d. Gaussian noise with correlations due to the differencing. As long as the tracking loops maintain lock, the integer ambiguity N_{kl} remains constant. To simplify notation, we assign a number n for each unique pair kl.

The position and orientation of the IMU is modeled identically as in Section 8.2 and is given by (8.3)–(8.5). The GPS measurement models (8.9) and (8.10) use the receiver position $\boldsymbol{r}^n$ and the receiver velocity $\dot{\boldsymbol{r}}^n$ of receivers a and b. However, the sensor unit motion (8.3) is modeled using the pose of IMU, $q^{nb}, \boldsymbol{b}^n$. Since the receivers are rigidly attached to the IMU, the expressions

$$\boldsymbol{r}_{a,t}^n = \boldsymbol{b}^n + R_t^{nb}\boldsymbol{r}_a^b, \qquad \dot{\boldsymbol{r}}_{a,t}^n = \dot{\boldsymbol{b}}^n + R_t^{nb}(\boldsymbol{\omega}_{nb,t}^b \times \boldsymbol{r}_a^b), \tag{8.11}$$

$$\boldsymbol{r}_{b,t}^n = \boldsymbol{b}^n + R_t^{nb}\boldsymbol{r}_b^b, \qquad \dot{\boldsymbol{r}}_{b,t}^n = \dot{\boldsymbol{b}}^n + R_t^{nb}(\boldsymbol{\omega}_{nb,t}^b \times \boldsymbol{r}_b^b), \tag{8.12}$$

can be used to calculate $\boldsymbol{r}^n$ and $\dot{\boldsymbol{r}}^n$ for receivers a and b.

Combining (8.3)–(8.5) and (8.9)–(8.11) we obtain a discrete-time nonlinear state-space model. Its state vector is, with slight abuse of notation, given by

$$\boldsymbol{x} = \left(\boldsymbol{b}^n, \dot{\boldsymbol{b}}^n, q^{nb}, \boldsymbol{\delta}_{\mathrm{a}}^b, \boldsymbol{\delta}_{\omega}^b\right)^T. \tag{8.13}$$

In case the integer ambiguities in (8.10) are known, it would be straightforward to use the model in an EKF to fuse the GPS and the inertial measurements Hirokawa and Ebinuma (2009). When they are unknown, a modified measurement update has to be used. As shown in Example 2.3, the measurement update of an EKF is equivalent to solving a nonlinear least squares problem. Using this optimization formulation, it is straightforward to add integer constraints, resulting in

$$\begin{aligned} \min_{\boldsymbol{x}_t, N_{1:N}} \quad & \frac{1}{2}\|\boldsymbol{e}_0\|_{\Sigma_{t|t-1}} + \frac{1}{2}\sum_{n=1}^{N}\|\boldsymbol{e}_n\|_{\Sigma_{\phi,n}} \\ \text{s.t.} \quad & \boldsymbol{e}_0 = \boldsymbol{x}_t - \hat{\boldsymbol{x}}_{t|t-1} \\ & \boldsymbol{e}_n = y_{\phi,n,t} - (\boldsymbol{u}_{n,t}^e)^T R_t^{nb}(\boldsymbol{r}_a^b - \boldsymbol{r}_b^b) + \lambda N_n, \quad n = 1,\ldots,N \\ & N_n \in \mathbb{Z}, \qquad\qquad\qquad\qquad\qquad\quad n = 1,\ldots,N \end{aligned} \tag{8.14}$$

This is an integer constrained nonlinear least squares problem, which is generally hard to solve. However, in the context of GPS ambiguity resolution, efficient methods have been derived (Hassibi and Boyd, 1998; Teunissen, 1995) which can be used to solve (8.14). These methods first calculate a so-called float solution without the integer constraint, then perform a de-correlation step and finally perform a local grid search.

The above discussion is summarized in Algorithm 8.2. Preliminary tests on a

Algorithm 8.2 Sensor fusion of inertial and dual GPS measurements

1. Perform an initialization and set $p(\boldsymbol{x}_0)$.
2. Do a time update. Propagate $p(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1})$ to $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})$ using the process model (8.3), (8.4) and (8.5) with the inertial measurements as input signals.
3. When available, use the GPS measurements to correct $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t-1})$ and obtain $p(\boldsymbol{x}_t|\boldsymbol{y}_{1:t})$. That is,
   - Perform a standard EKF measurement update using the PVT measurement model (8.9).
   - Perform an integer constrained measurement update (8.14) using the double difference carrier phase measurement model (8.10).
4. When the estimated position becomes too large, redefine the n-frame. Use the state transformation (8.8) to obtain $p(\tilde{\boldsymbol{x}}_t|\boldsymbol{y}_{1:t})$ and set $\boldsymbol{x}_t := \tilde{\boldsymbol{x}}_t$.
5. Set $t := t + 1$ and iterate from Step 2.

stationary setup have done with success. However, it remains future work to perform experiments and validate the performance of Algorithm 8.2.

9

Concluding remarks

This thesis discusses sensor fusion and calibration for the combination of inertial sensors with vision, *ultra-wideband* (UWB) and *global positioning system* (GPS). These applications are discussed individually, starting from a common theoretical background. The aim has been to show how to combine the information obtained from the different sensors as well as how the calibration issues when working with multiple sensors can be solved using simple and efficient procedures. The conclusions of this thesis are given in Section 9.1. Section 9.2 contains some directions for future research. Both sections follow the outline of this thesis.

9.1 Conclusions

Part I of this thesis contains a concise overview of inertial sensors, vision, UWB and GPS and the kinematics linking their measurements.

Part II discusses sensor fusion for a number of sensor combinations, together with associated calibration problems. Chapter 5 is about standalone inertial sensors. It contains ideas on applying an optimization based smoothing method to a multi-segment inertial motion capture system. Furthermore, a novel calibration method has been developed to calibrate the magnetic field measurements of an *inertial measurement unit* (IMU) mounted close to a ferro-magnetic object. The method has been demonstrated by using it to calibrate an IMU mounted in a jet airplane.

Estimating position and orientation in real-time using measurements from vision and inertial sensors is discussed in Chapter 6. A system has been developed to solve this problem in unprepared environments, assuming that a map or scene model is available. The system has been tested in an *augmented reality* (AR) setup

and its performance has been evaluated in experiments with an industrial robot. Additionally, an easy-to-use calibration method to determine the relative position and orientation of a rigidly connected camera and IMU is derived. The method has been successfully applied to a number of sensor units with both perspective and fisheye lenses.

In Chapter 7, a calibration method to determine the clock parameters and the receiver positions of a UWB positioning system has been developed and tested. It allows the user to deploy a UWB system and calibrate it within minutes. Furthermore, a tightly coupled sensor fusion algorithm has been developed to estimate real-time position and orientation using inertial and UWB measurements. Its performance has been evaluated using a comparison to an optical reference system.

Finally, Chapter 8 contains a loosely coupled sensor fusion algorithm for combining an inertial sensor with GPS. Its performance has been evaluated on a flight of a jet aircraft. This experiment also motivates the need for magnetic field calibration. Additionally, ideas on a sensor fusion method for inertial sensors and dual GPS are discussed.

9.2 Future work

Some directions for future research have been mentioned in the previous chapters. Of particular interest is the combination of simultaneous sensor fusion and calibration. The optimization approach to smoothing described in Chapter 2 seems to be very suitable for this purpose. Although formulated as a post-processing method, recent work in the *simultaneous localization and mapping* (SLAM) community has shown that similar problems can be solved in almost real-time. The challenge here is to use these ideas and obtain efficient implementations for combined sensor fusion and calibration problems which can be used in real-time applications.

Chapter 5 contains ideas on applying an optimization based smoothing algorithm to multi-segment systems. This approach is very interesting in the context of e.g. clinical applications, which tend to allow for post-processing. A natural continuation here is to further develop and implement the solver and demonstrate its performance experimentally on a relevant application.

For the combination of inertial sensors with dual GPS, a sensor fusion algorithm which takes care of the integer ambiguity has been discussed in Chapter 8. The contribution here is rather theoretical and the logical next step is to test the method on experimental data and validate its performance.

Solutions to calibration problems are essential to obtain efficient sensor fusion methods. Two calibration problems not investigated in this thesis are how to obtain the lever arms in the context of both inertial / GPS sensor fusion and multi-segment systems. These calibration problems are very relevant and form interesting topics for further research.

A

Linear Algebra

This appendix provides a short introduction to selected topics of linear algebra. More details can be found in the references Magnus and Neudecker (1999); Golub and Van Loan (1996); Björk (1996); Kailath et al. (2000).

A.1 Matrix differentials

The vectorize operator (vec) transforms a matrix $A \in \mathbb{R}^{m \times n}$ into a $mn \times 1$ column vector by stacking the columns $\boldsymbol{a}_i$ of the matrix below another, that is,

$$\operatorname{vec} A = \begin{bmatrix} \boldsymbol{a}_1 \\ \vdots \\ \boldsymbol{a}_n \end{bmatrix}, \qquad \boldsymbol{a}_i = \begin{bmatrix} a_{1i} \\ \vdots \\ a_{mi} \end{bmatrix}. \tag{A.1}$$

It can be shown that for matrices of matching dimensions it holds that

$$\operatorname{vec} ABC = (C^T \otimes A) \operatorname{vec} B, \tag{A.2}$$

were $\otimes$ denotes the Kronecker product. For any two matrices $A \in \mathbb{R}^{m \times n}$ and $B \in \mathbb{R}^{p \times q}$ the Kronecker product is defined as the $mp \times nq$ matrix

$$A \otimes B = \begin{bmatrix} a_{11}B & \dots & a_{1n}B \\ \vdots & & \vdots \\ a_{m1}B & \dots & a_{mn}B \end{bmatrix}. \tag{A.3}$$

Using the vec operator, the Jacobian matrix of a function f is defined as

$$[\mathrm{D}_{\boldsymbol{x}} f](\boldsymbol{a}) = \left.\frac{\partial \operatorname{vec} f(\boldsymbol{x})}{\partial (\operatorname{vec} \boldsymbol{x})^T}\right|_{\boldsymbol{x}=\boldsymbol{a}}. \tag{A.4}$$

This definition always provides a well defined Jacobian, regardless of the dimensions of the function f or its argument $\boldsymbol{x}$. They can be any combination of scalars, vectors and matrices. As an alternative to evaluating each partial derivative in (A.4), the Jacobian $\mathrm{D}_{\boldsymbol{x}} f$ can be identified from the (vectorized) differential of f:

$$\mathrm{d} \operatorname{vec} f(\boldsymbol{x}) = A(\boldsymbol{x}) \, \mathrm{d} \operatorname{vec} \boldsymbol{x} \quad \Leftrightarrow \quad [\mathrm{D}_{\boldsymbol{x}} f](\boldsymbol{a}) = A(\boldsymbol{a}). \tag{A.5}$$

Since computations with differentials are relatively easy, this is a rather useful and powerful approach. The Jacobian of a composite function $h = g \circ f$ is given by the chain rule,

$$[\mathrm{D}_{\boldsymbol{x}} h](\boldsymbol{a}) = [\mathrm{D}_{\boldsymbol{x}} (g \circ f)](\boldsymbol{a}) = [\mathrm{D}_{\boldsymbol{y}} g](\boldsymbol{b}) \, [\mathrm{D}_{\boldsymbol{x}} f](\boldsymbol{a}), \tag{A.6}$$

where $\boldsymbol{b} = f(\boldsymbol{a})$.

A.2 Special matrices

Triangular and block (diagonal) matrices are examples of frequently occurring matrices which have a special structure. Exploiting their properties, it is possible to obtain useful results. Selected results are presented below.

Triangular matrices

A matrix $A \in \mathbb{R}^{n \times n}$ is lower triangular if $a_{ij} = 0$ for $i < j$. Its inverse $B = A^{-1}$ is also lower triangular,

$$\begin{bmatrix} a_{11} & 0 & \dots & 0 \\ a_{21} & a_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}^{-1} = \begin{bmatrix} b_{11} & 0 & \dots & 0 \\ b_{21} & b_{22} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nn} \end{bmatrix}, \tag{A.7}$$

where the entries on the first two diagonals are given by

$$b_{ii} = a_{ii}^{-1}, \quad b_{i,i-1} = -\frac{a_{i,i-1}}{a_{i-1,i-1} a_{ii}}.$$

Block matrices

Consider a matrix $M \in \mathbb{R}^{n \times n}$ partitioned as

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix}, \tag{A.8}$$

where the blocks A, B, C, D are matrices of appropriate dimensions. Assuming the inverses exist, M can be factored as

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} I & 0 \\ CA^{-1} & I \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & \Delta_{\mathrm{A}} \end{bmatrix} \begin{bmatrix} I & A^{-1}B \\ 0 & I \end{bmatrix}, \tag{A.9}$$

where $\Delta_{\mathrm{A}} = D - CA^{-1}B$ is called the Schur complement of A in M. Among other things, this implies that determinant of M can be factored as

$$\det(M) = \det(A)\det(\Delta_{\mathrm{A}}). \tag{A.10}$$

Furthermore, when M is invertible, M^{-1} can be obtained by inverting (A.9)

$$\begin{aligned} M^{-1} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} &= \begin{bmatrix} I & -A^{-1}B \\ 0 & I \end{bmatrix} \begin{bmatrix} A^{-1} & 0 \\ 0 & \Delta_{\mathrm{A}}^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -CA^{-1} & I \end{bmatrix} \\ &= \begin{bmatrix} A^{-1} + A^{-1}B\Delta_{\mathrm{A}}^{-1}CA^{-1} & -A^{-1}B\Delta_{\mathrm{A}}^{-1} \\ -\Delta_{\mathrm{A}}^{-1}CA^{-1} & \Delta_{\mathrm{A}}^{-1} \end{bmatrix}. \end{aligned} \tag{A.11}$$

Permuting the rows and columns of M and applying (A.9), an alternative factorization is obtained

$$M = \begin{bmatrix} A & B \\ C & D \end{bmatrix} = \begin{bmatrix} I & BD^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} \Delta_{\mathrm{D}} & 0 \\ 0 & D \end{bmatrix} \begin{bmatrix} I & 0 \\ D^{-1}C & I \end{bmatrix}, \tag{A.12}$$

where $\Delta_{\mathrm{D}} = A - BD^{-1}C$ is called the Schur complement of D in M. This implies that M^{-1} can also be written as

$$\begin{aligned} M^{-1} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} &= \begin{bmatrix} I & 0 \\ -D^{-1}C & I \end{bmatrix} \begin{bmatrix} \Delta_{\mathrm{D}}^{-1} & 0 \\ 0 & D^{-1} \end{bmatrix} \begin{bmatrix} I & -BD^{-1} \\ 0 & I \end{bmatrix} \\ &= \begin{bmatrix} \Delta_{\mathrm{D}}^{-1} & -\Delta_{\mathrm{D}}^{-1}BD^{-1} \\ -D^{-1}C\Delta_{\mathrm{D}}^{-1} & D^{-1} + D^{-1}C\Delta_{\mathrm{D}}^{-1}BD^{-1} \end{bmatrix}. \end{aligned} \tag{A.13}$$

Equating the (1,1)-block of (A.11) and (A.13), we obtain the identity

$$\begin{aligned} \Delta_{\mathrm{D}}^{-1} &= A^{-1} + A^{-1}B\Delta_{\mathrm{A}}^{-1}CA^{-1} \\ (A - BD^{-1}C)^{-1} &= A^{-1} + A^{-1}B(D - CA^{-1}B)^{-1}CA^{-1}. \end{aligned} \tag{A.14a}$$

Substituting $D = C^{-1}$ and $C = -D$, the well-known matrix inversion lemma,

$$(A + BCD)^{-1} = A^{-1} + A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1}, \tag{A.14b}$$

is obtained.

A.3 Matrix factorizations

The general approach to solving a linear system $A\boldsymbol{x} = \boldsymbol{b}$ is to express A as a product of matrices

$$A = A_1 A_2 \dots A_m. \tag{A.15}$$

The factors A_i are typically chosen such that the subproblems $A_i \boldsymbol{z}_i = \boldsymbol{z}_{i-1}$ are easy to solve, e.g., they are (block) diagonal, triangular or permutation matrices.

For sparse matrices, a number of matrix factorizations are especially useful. They are introduced below.

Cholesky factorization

Every symmetric, positive definite matrix $A \in \mathbb{R}^{n \times n}$ can be factored as

$$PAP^T = LL^T, \tag{A.16}$$

where P is an arbitrary permutation matrix and L is a nonsingular, lower triangular matrix with positive diagonal elements. For every choice of P there exists a unique associated factor L, but the choice can affect the sparsity of the associated factor L.

LDL^T factorization

Every symmetric nonsingular matrix $A \in \mathbb{R}^{n \times n}$ can be factored as

$$PAP^T = LDL^T, \tag{A.17}$$

where P is a permutation matrix, L is a nonsingular, lower triangular matrix with unit diagonal and D is a block diagonal matrix consisting of nonsingular 1×1 and 2×2 blocks.

A.4 Partial inverses

Calculating the inverse of a matrix A is a challenging task in terms of floating point operations, numerical accuracy as well as storage. Especially for large matrices the situation tends to become problematic, which is why explicit evaluation of the inverse is avoided whenever possible. However, in some cases the inverse has to be calculated. Covariance analysis of a least squares problem is such a case.

Instead of calculating the entire matrix inverse A^{-1} — which is a computationally very intensive task — parts of it can be computed very efficiently given a sparse factorization of A. This is the case for the elements corresponding to the nonzero elements of the factorization, which include the diagonal elements. The next sections describe the procedure for two factorizations.

Cholesky factorization

Factoring a symmetric, positive definite matrix $A = A^T \succ 0$ as $A = LL^T$ using a Cholesky factorization, its inverse B is given as

$$B \triangleq A^{-1} = (LL^T)^{-1} = L^{-T}L^{-1}. \tag{A.18}$$

As noticed by Golub and Plemmons (1980), (A.18) can be rearranged to

$$L^T B = L^{-1}. \tag{A.19}$$

Since L is an lower triangular matrix, L^{-1} is a lower triangular matrix with known diagonal. Indeed, using (A.7) we have $(L^{-1})_{ij} = 0$ for $i < j$ and $(L^{-1})_{ii} = l_{ii}^{-1}$. That is, the upper triangular elements of the right-hand side of (A.19) are known and we obtain the following system of equations

$$\begin{cases} \sum_{k=i}^{n} l_{ki} b_{kj} = 0 & i < j \\ \sum_{k=i}^{n} l_{ki} b_{kj} = \dfrac{1}{l_{ii}} & i = j \end{cases} \tag{A.20}$$

which fully determines $B = B^T$. Its solution is given by

$$\begin{cases} b_{ii} = \dfrac{1}{l_{ii}} \left(\dfrac{1}{l_{ii}} - \sum_{k=i+1}^{n} l_{ki} b_{ki} \right) \\ b_{ij} = -\dfrac{1}{l_{ii}} \left(\sum_{k=i+1}^{j} l_{ki} b_{kj} + \sum_{k=j+1}^{n} l_{ki} b_{jk} \right) \quad i < j \end{cases} \tag{A.21}$$

which can be used to recursively calculate the elements of B, starting with its last column. Closer analysis shows that only the b_{ij} for which $l_{ij} \neq 0$ are required to complete the recursion. Hence, for sparse factorizations, (A.21) becomes a very efficient method to calculate these elements of the inverse.

LDL^T factorization

Factoring a symmetric, non-singular matrix $A = A^T$ as $A = LDL^T$ using a LDL^T factorization, its inverse B is given as

$$B \triangleq A^{-1} = (LDL^T)^{-1} = L^{-T} D^{-1} L^{-1}. \tag{A.22}$$

Similar to (A.19), (A.22) can be arranged as

$$L^T B = D^{-1} L^{-1} = T. \tag{A.23}$$

Since L is an lower triangular matrix with unit diagonal, L^{-1} also is a lower triangular matrix. Indeed, using (A.7) we have $(L^{-1})_{ij} = 0$ for $i < j$, $(L^{-1})_{ii} = 1$ and $(L^{-1})_{i,i-1} = -l_{i,i-1}$. Furthermore, D is a block diagonal matrix consisting of 1×1 and 2×2 blocks which is straightforward to invert. Hence, expanding the product $T \triangleq D^{-1} L^{-1}$, we obtain $t_{ij} = 0$ for $i + 1 < j$ and

$$t_{i,i+1} = (D^{-1})_{i,i+1}, \qquad t_{ii} = (D^{-1})_{ii} - l_{i+1,i} (D^{-1})_{i,i+1}. \tag{A.24}$$

That is, upper triangular part of the right-hand side of (A.23) is known and we obtain the following system of equations

$$\begin{cases} \sum_{k=i}^{n} l_{ki} b_{kj} = 0 & i < j-1 \\ \sum_{k=i}^{n} l_{ki} b_{kj} = t_{ij} & i \in [j-1, j] \end{cases} \tag{A.25}$$

which fully determines $B = B^T$. Its solution is given by

$$\begin{cases} b_{ii} = t_{ii} - \sum_{k=i+1}^{n} l_{ki} b_{ki} & \\ b_{ij} = t_{ij} - \sum_{k=i+1}^{j} l_{ki} b_{kj} - \sum_{k=j+1}^{n} l_{ki} b_{jk} & i = j-1 \\ b_{ij} = -\sum_{k=i+1}^{j} l_{ki} b_{kj} - \sum_{k=j+1}^{n} l_{ki} b_{jk} & i < j-1 \end{cases} \tag{A.26}$$

which can be used to recursively calculate the elements of B, starting with its last column. Closer analysis shows that only the b_{ij} for which $l_{ij} \neq 0$ or $d_{ij} \neq 0$ are required to complete the recursion. Hence, for sparse factorizations, (A.26) becomes a very efficient method to calculate these elements of the inverse.

B

Quaternion Algebra

This appendix provides a very short introduction to quaternions and their properties. Only the most basic operations are stated, mostly without proof. For more details, see e.g. Kuipers (1999); Hamilton (1844).

B.1 Operators and properties

A quaternion $q \in \mathbb{R}^4$ is a 4-tuple of real numbers, denoted as $q = (q_0, q_1, q_2, q_3)$. An alternative notation is $q = (q_0, \boldsymbol{q})$, where q_0 is called the scalar part and $\boldsymbol{q}$ the vector part of a quaternion. Special quaternions sets are $\mathcal{Q}_s = \{q \in \mathbb{R}^4 : \boldsymbol{q} = \boldsymbol{0}\}$, $\mathcal{Q}_v = \{q \in \mathbb{R}^4 : q_0 = 0\}$ and $\mathcal{Q}_1 = \{q \in \mathbb{R}^4 : \|q\|_2 = 1\}$. When clear from the context, conversions to and from scalars and vectors are performed implicitly.

For quaternions the following operators are defined:

addition	$p + q \triangleq (p_0 + q_0, \boldsymbol{p} + \boldsymbol{q}),$	(B.1)
multiplication	$pq = p \odot q \triangleq (p_0 q_0 - \boldsymbol{p} \cdot \boldsymbol{q}, p_0 \boldsymbol{q} + q_0 \boldsymbol{p} + \boldsymbol{p} \times \boldsymbol{q}),$	(B.2)
conjugation	$q^c \triangleq (q_0, -\boldsymbol{q}),$	(B.3)
norm	$\|q\|_2 \triangleq (q_0^2 + \boldsymbol{q} \cdot \boldsymbol{q})^{\frac{1}{2}} = \sqrt{(q \odot q^c)_0}$	(B.4)
inverse	$q^{-1} \triangleq \|q\|_2^{-2} q^c,$	(B.5)
inner product	$p \cdot q \triangleq -\frac{1}{2}(p \odot q + q \odot p),$	(B.6)
cross product	$p \times q \triangleq \frac{1}{2}(p \odot q - q \odot p).$	(B.7)

Associative and distributive properties hold, but only additions are commutative.

$$p + (q + r) = (p + q) + r, \tag{B.8}$$
$$p + q = q + p, \tag{B.9}$$
$$p \odot (q \odot r) = (p \odot q) \odot r, \tag{B.10}$$
$$p \odot (q + r) = p \odot q + p \odot r. \tag{B.11}$$

Multiplication with a scalar $\lambda \in \mathcal{Q}_s$ is commutative,

$$\lambda q = (\lambda, \mathbf{0}) \odot (q_0, \boldsymbol{q}) = (\lambda q_0, \lambda \boldsymbol{q}) = q\lambda, \tag{B.12}$$

but, in general quaternion multiplications do not commutate

$$p \odot q \neq q \odot p. \tag{B.13}$$

Furthermore, the following identities are useful,

$$(p \odot q)^c = q^c \odot p^c, \tag{B.14}$$
$$(p \odot q)^{-1} = q^{-1} \odot p^{-1}, \tag{B.15}$$
$$\|p \odot q\|_2 = \|p\|_2 \|q\|_2. \tag{B.16}$$

B.2 Multiplication

The multiplication of quaternions is a bilinear operator. This property gives rise to introduction of the left and the right multiplication operators, $\cdot^L$, $\cdot^R$,

$$\begin{aligned} p \odot q &= (p_0 q_0 - \boldsymbol{p} \cdot \boldsymbol{q}, p_0 \boldsymbol{q} + q_0 \boldsymbol{p} + \boldsymbol{p} \times \boldsymbol{q}) \\ &= \underbrace{\begin{bmatrix} p_0 & -\boldsymbol{p}^T \\ \boldsymbol{p} & p_0 I_3 + [\boldsymbol{p}\times] \end{bmatrix}}_{\triangleq p^L} \begin{bmatrix} q_0 \\ \boldsymbol{q} \end{bmatrix} = \underbrace{\begin{bmatrix} q_0 & -\boldsymbol{q}^T \\ \boldsymbol{q} & q_0 I_3 - [\boldsymbol{q}\times] \end{bmatrix}}_{\triangleq q^R} \begin{bmatrix} p_0 \\ \boldsymbol{p} \end{bmatrix}. \end{aligned} \tag{B.17}$$

The structure of these operators and their Jacobians is very simple,

$$q^L = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & -q_3 & q_2 \\ q_2 & q_3 & q_0 & -q_1 \\ q_3 & -q_2 & q_1 & q_0 \end{bmatrix}, \qquad \mathrm{D}_q\, q^L = \begin{bmatrix} e_0^R \\ e_1^R \\ e_2^R \\ e_3^R \end{bmatrix}, \tag{B.18a}$$

$$q^R = \begin{bmatrix} q_0 & -q_1 & -q_2 & -q_3 \\ q_1 & q_0 & q_3 & -q_2 \\ q_2 & -q_3 & q_0 & q_1 \\ q_3 & q_2 & -q_1 & q_0 \end{bmatrix}, \qquad \mathrm{D}_q\, q^R = \begin{bmatrix} e_0^L \\ e_1^L \\ e_2^L \\ e_3^L \end{bmatrix}. \tag{B.18b}$$

Here, $\{e_i\}_{i=0}^3$ is the standard basis of $\mathbb{R}^4$. The operators satisfy the following identities,

$$p^L q^R = q^R p^L, \tag{B.19}$$
$$(q^c)^L = (q^L)^T, \quad (q^c)^R = (q^R)^T. \tag{B.20}$$

The multiplication operators turn out to be very useful for simplifying expressions, see Example B.1 and Example B.2 below.

B.1 Example: Differentiation

Rewriting the quaternion product as $p \odot q = q^R p = p^L q$, it becomes trivial to derive its Jacobians

$$\mathrm{D}_p\, p \odot q = \mathrm{D}_p\, q^R p = q^R, \qquad \mathrm{D}_q\, p \odot q = \mathrm{D}_q\, p^L q = p^L.$$

B.2 Example: Rotation

The rotation of a vector $v \in \mathcal{Q}_v$ by a quaternion $q \in \mathcal{Q}_1$ can be simplified as

$$\begin{aligned} q \odot v \odot q^c &= q^L (q^R)^T v \\ &= \begin{bmatrix} q_0 & -\boldsymbol{q}^T \\ \boldsymbol{q} & q_0 I_3 + [\boldsymbol{q}\times] \end{bmatrix} \begin{bmatrix} q_0 & \boldsymbol{q}^T \\ -\boldsymbol{q} & q_0 I_3 + [\boldsymbol{q}\times] \end{bmatrix} \begin{bmatrix} 0 \\ \boldsymbol{v} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0_{1\times 3} \\ 0_{3\times 1} & \boldsymbol{q}\boldsymbol{q}^T + q_0^2 I_3 + 2q_0[\boldsymbol{q}\times] + [\boldsymbol{q}\times]^2 \end{bmatrix} \begin{bmatrix} 0 \\ \boldsymbol{v} \end{bmatrix} = \begin{bmatrix} 0 \\ R\boldsymbol{v} \end{bmatrix}. \end{aligned}$$

That is, rotation is a linear operator and it only affects the vector part.

B.3 Exponential and logarithm

The quaternion exponential is defined using a power series:

$$\exp q \triangleq \sum_{n=0}^{\infty} \frac{q^n}{n!}. \tag{B.21}$$

Here the quaternion power is recursively defined as

$$q^n = q \odot q^{n-1} = q^{n-1} \odot q, \qquad q^0 = 1. \tag{B.22}$$

With the above definitions the quaternion exponential (B.21) satisfies a number of useful relations, including the differential equation

$$\frac{\partial}{\partial t} \exp(tq) = q \odot \exp(tq) = \exp(tq) \odot q \tag{B.23}$$

for scalar t. The first equality follows from

$$\frac{\partial}{\partial t} \exp(tq) = \sum_{n=1}^{\infty} \frac{t^{n-1} q^n}{(n-1)!} = q \odot \left(\sum_{m=0}^{\infty} \frac{(tq)^m}{m!} \right) = q \odot \exp(tq),$$

and the latter equality can be derived using a similar argument. Furthermore, the quaternion exponential commutes with rotation and conjugation. For rotation we

have that

$$\exp(p \odot q \odot p^{-1}) = \sum_{n=0}^{\infty} \frac{(p \odot q \odot p^{-1})^n}{n!} = p \odot \left(\sum_{n=0}^{\infty} \frac{q^n}{n!} \right) \odot p^{-1} = p \odot \exp(q) \odot p^{-1}. \tag{B.24}$$

Similarly, conjugation of the quaternion exponent yields

$$\exp(q^c) = \sum_{n=0}^{\infty} \frac{(q^c)^n}{n!} = \left(\sum_{n=0}^{\infty} \frac{q^n}{n!} \right)^c = (\exp q)^c. \tag{B.25}$$

The exponential of a vector $v = (0, \boldsymbol{v}) \in \mathcal{Q}_v$ deserves some more attention. This restriction implies that $v^2 = v \odot v = (0 \cdot 0 - \boldsymbol{v} \cdot \boldsymbol{v}, 0\boldsymbol{v} + 0\boldsymbol{v} + \boldsymbol{v} \times \boldsymbol{v}) = (-\|\boldsymbol{v}\|_2^2, \boldsymbol{0})$. Hence, (B.21) simplifies to

$$\begin{aligned} \exp v &= \sum_{n=0}^{\infty} \frac{v^n}{n!} = \sum_{n=0}^{\infty} \frac{v^{2n}}{2n!} + \sum_{n=0}^{\infty} \frac{v^{2n+1}}{(2n+1)!} \\ &= \left(\sum_{n=0}^{\infty} (-1)^n \frac{\|\boldsymbol{v}\|_2^{2n}}{(2n)!}, \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|_2} \sum_{n=0}^{\infty} (-1)^n \frac{\|\boldsymbol{v}\|_2^{2n+1}}{(2n+1)!} \right) \\ &= \left(\cos \|\boldsymbol{v}\|_2, \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|_2} \sin \|\boldsymbol{v}\|_2 \right) \in \mathcal{Q}_1. \end{aligned} \tag{B.26}$$

The exponential of a vector returns a unit quaternion as $\|\exp v\|_2 = 1$. The Jacobian of (B.26) is given by

$$\mathrm{D}_{\boldsymbol{v}} \exp v = \begin{bmatrix} -\frac{\boldsymbol{v}^T \sin \|\boldsymbol{v}\|_2}{\|\boldsymbol{v}\|_2} \\ \frac{I_3 \sin \|\boldsymbol{v}\|_2}{\|\boldsymbol{v}\|_2} - \frac{\boldsymbol{v}\boldsymbol{v}^T \sin \|\boldsymbol{v}\|_2}{\|\boldsymbol{v}\|_2^3} + \frac{\boldsymbol{v}\boldsymbol{v}^T \cos \|\boldsymbol{v}\|_2}{\|\boldsymbol{v}\|_2^2} \end{bmatrix}. \tag{B.27}$$

The logarithm for unit quaternions $q = (q_0, \boldsymbol{q}) \in \mathcal{Q}_1$ can be defined[1] as

$$\log q \triangleq \frac{q_0}{|q_0|} \frac{\boldsymbol{q}}{\|\boldsymbol{q}\|_2} \arcsin \|\boldsymbol{q}\|_2. \tag{B.28}$$

This definition is such that the relation $\log \exp(0, \boldsymbol{v}) = \boldsymbol{v}$ behaves as expected. The Jacobian of (B.28) is given by

$$\mathrm{D}_{\boldsymbol{q}} \log q = \begin{bmatrix} 0_{3\times 1} & \frac{I_3 \|\boldsymbol{v}\|_2}{\sin \|\boldsymbol{v}\|_2} - \frac{\boldsymbol{v}\boldsymbol{v}^T}{\|\boldsymbol{v}\|_2 \sin \|\boldsymbol{v}\|_2} + \frac{\boldsymbol{v}\boldsymbol{v}^T}{\|\boldsymbol{v}\|_2^2 \cos \|\boldsymbol{v}\|_2} \end{bmatrix}, \tag{B.29}$$

where $\boldsymbol{v} = \log q$.

[1] An alternative definition is $\log q = \frac{\boldsymbol{q}}{\|\boldsymbol{q}\|_2} \arccos q_0$, whose Jacobian is singular at e_0.

C

Orientation conversion

Orientations can be described with several interchangeable parameterizations. This appendix gives conversions between unit quaternions, rotation vectors, rotation matrices and Euler angles.

C.1 Rotation vectors

A rotation around axis $\boldsymbol{n}$ by angle α has a rotation vector $\boldsymbol{\phi} = \boldsymbol{n}\alpha$. The conversion to and from a unit quaternion is given by

$$q^{uv} = \exp -\tfrac{1}{2}\boldsymbol{\phi}^{uv}, \qquad \boldsymbol{\phi}^{uv} = -2\log q^{uv}, \tag{C.1}$$

where exp and log denote the quaternion exponent and the quaternion logarithm defined in Appendix B.3.

C.2 Rotation matrices

A unit quaternion q^{uv} can be converted to a rotation matrix R^{uv} using

$$\begin{aligned} R^{uv} &= \boldsymbol{q}\boldsymbol{q}^T + q_0^2 I_3 + 2q_0[\boldsymbol{q}\times] + [\boldsymbol{q}\times]^2 \\ &= \begin{bmatrix} 2q_0^2 + 2q_1^2 - 1 & 2q_1q_2 - 2q_0q_3 & 2q_1q_3 + 2q_0q_2 \\ 2q_1q_2 + 2q_0q_3 & 2q_0^2 + 2q_2^2 - 1 & 2q_2q_3 - 2q_0q_1 \\ 2q_1q_3 - 2q_0q_2 & 2q_2q_3 + 2q_0q_1 & 2q_0^2 + 2q_3^2 - 1 \end{bmatrix}, \end{aligned} \tag{C.2}$$

where the annotation ^{uv} has been left out for readability. The conversion of a rotation matrix R^{uv} to a unit quaternion $q^{uv} = (q_0, \boldsymbol{q})$ is given by

$$q_0 = \sqrt{\frac{1 + \operatorname{tr} R}{4}}, \qquad \boldsymbol{q} = \frac{1}{2q_0}\begin{bmatrix} R_{23} - R_{32} \\ R_{31} - R_{13} \\ R_{12} - R_{21} \end{bmatrix}, \tag{C.3}$$

where the annotation ^{uv} has been left out for readability. If $1 + \operatorname{tr} R^{uv} \ll 1$ numerical difficulties might arise when evaluation (C.3). In this case a slightly modified conversion should be used.

C.3 Euler angles

Euler angles define a rotation using a series of three rotations around a coordinate axis. In case of the popular aerospace sequence, one rotates first an angle ψ about the z-axis, then an angle θ around the y-axis and finally an angle ϕ around the x-axis. Here, the convention is to call ψ heading or yaw, θ elevation or pitch and ϕ bank or roll. Using (C.1), the Euler angles $\boldsymbol{e}^{uv} = (\phi, \theta, \psi)$ are converted to a unit quaternion using

$$q^{uv} = \exp(-\tfrac{\phi}{2}\boldsymbol{e}_1) \odot \exp(-\tfrac{\theta}{2}\boldsymbol{e}_2) \odot \exp(-\tfrac{\psi}{2}\boldsymbol{e}_3), \tag{C.4}$$

where $\{\boldsymbol{e}_i\}_{i=1}^3$ is the standard basis in $\mathbb{R}^3$. The rotation matrix equivalent is

$$\begin{aligned} R^{uv} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}\begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix}\begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix}. \end{aligned} \tag{C.5}$$

The conversion to Euler angles is given by

$$\psi = \tan^{-1}\left(\frac{R_{12}}{R_{11}}\right) = \tan^{-1}\left(\frac{2q_1q_2 - 2q_0q_3}{2q_0^2 + 2q_1^2 - 1}\right), \tag{C.6a}$$

$$\theta = -\sin^{-1}(R_{13}) = -\sin^{-1}(2q_1q_3 + 2q_0q_2), \tag{C.6b}$$

$$\phi = \tan^{-1}\left(\frac{R_{23}}{R_{33}}\right) = \tan^{-1}\left(\frac{2q_2q_3 - 2q_0q_1}{2q_0^2 + 2q_3^2 - 1}\right). \tag{C.6c}$$

Bibliography

Analog Devices, 2010. [Online]. Available: `http://www.analog.com/`.

L. Armesto, J. Tornero, and M. Vincze. Fast ego-motion estimation with multi-rate fusion of inertial and vision. *International Journal of Robotics Research*, 26(6):577–589, 2007. doi: 10.1177/0278364907079283.

M. Aron, G. Simon, and M.-O. Berger. Use of inertial sensors to support video tracking. *Computer Animation and Virtual Worlds*, 18(1):57–68, 2007. doi: 10.1002/cav.161.

M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, Feb. 2002. doi: 10.1109/78.978374.

T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (SLAM): Part II. *IEEE Robotics & Automation Magazine*, 13(3):108–117, Sept. 2006. doi: 10.1109/MRA.2006.1678144.

H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Journal of Computer Vision and Image Understanding*, pages 346–359, June 2008. doi: 10.1016/j.cviu.2007.09.014.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Information Science and Statistics. Springer-Verlag, New York, USA, 2006. ISBN 9780387310732.

Å. Björk. *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia, 1996. ISBN 0898713609.

G. Bleser and D. Stricker. Advanced tracking through efficient image processing and visual-inertial sensor fusion. In *Proceedings of IEEE Virtual Reality Conference*, pages 137–144, Reno, NE, USA, Mar. 2008. doi: 10.1109/VR.2008.4480765.

J.-Y. Bouguet. Camera calibration toolbox for matlab, 2003. [Online]. Available: `http://www.vision.caltech.edu/bouguetj/calib_doc/`.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. ISBN 0521833787.

R. G. Brown and P. Y. C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. John Wiley & Sons, Ltd, 3rd edition, 1996.

E. J. Candès, J. Romberg, and T. Tao. Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on Information Theory*, 52(2):489–509, Feb. 2006. doi: 10.1109/TIT.2005.862083.

Y. Chan and K. Ho. A simple and efficient estimator for hyperbolic location. *IEEE Transactions on Signal Processing*, 42(8):1905–1915, Aug. 1994. doi: 10.1109/78.301830.

J. Chandaria, G. Thomas, B. Bartczak, K. Koeser, R. Koch, M. Becker, G. Bleser, D. Stricker, C. Wohlleber, M. Felsberg, F. Gustafsson, J. D. Hol, T. B. Schön, J. Skoglund, P. J. Slycke, and S. Smeitz. Real-time camera tracking in the MATRIS project. *SMPTE Motion Imaging Journal*, 116(7–8):266–271, Aug. 2007a.

J. Chandaria, G. A. Thomas, and D. Stricker. The MATRIS project: real-time markerless camera tracking for augmented reality and broadcast applications. *Journal of Real-Time Image Processing*, 2(2):69–79, Nov. 2007b. doi: 10.1007/s11554-007-0043-z.

A. Chatfield. *Fundamentals of High Accuracy Inertial Navigation*, volume 174. American Institute of Aeronautics and Astronautics, USA, 3rd edition, 1997. ISBN 1563472430.

S. G. Chroust and M. Vincze. Fusion of vision and inertial data for motion and structure estimation. *Journal of Robotics Systems*, 21(2):73–83, 2004. doi: 10.1002/rob.10129.

P. Corke, J. Lobo, and J. Dias. An introduction to inertial and visual sensing. *International Journal of Robotics Research*, 26(6):519–535, 2007. doi: 10.1177/0278364907079279.

J. L. Crassidis, F. L. Markley, and Y. Cheng. Survey of nonlinear attitude estimation methods. *Journal of Guidance, Control, and Dynamics*, 30(1):12–28, jan 2007. doi: 10.2514/1.22452.

K. Daniilidis. Hand-eye calibration using dual quaternions. *International Journal of Robotics Research*, 18(3):286–298, Mar. 1999. ISSN 0278-3649. doi: 10.1177/02783649922066213.

A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, June 2007. doi: 10.1109/TPAMI.2007.1049.

D. L. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, Apr. 2006. doi: 10.1109/TIT.2006.871582.

H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping (SLAM): Part I. *IEEE Robotics & Automation Magazine*, 13(2):99–110, June 2006. doi: 10.1109/MRA.2006.1638022.

N. El-Sheimy, H. Hou, and X. Niu. Analysis and modeling of inertial sensors using Allan variance. *IEEE Transactions on Instrumentation and Measurement*, 57(1):140–149, jan 2008. doi: 10.1109/TIM.2007.908635.

F. Ferraris, U. Grimaldi, and M. Parvis. Procedure for effortless in-field calibration of three-axial rate gyro and accelerometers. *Sensors and Materials*, 7(5):311–330, 1995. ISSN 0914-4935.

M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981. doi: 10.1145/358669.358692.

E. Foxlin. Pedestrian tracking with shoe-mounted inertial sensors. *IEEE Computer Graphics and Applications*, 25(6):38–46, nov 2005. doi: 10.1109/MCG.2005.140.

E. Foxlin and L. Naimark. Miniaturization, calibration & accuracy evaluation of a hybrid self-tracker. In *Proceedings of 2nd International Symposium on Mixed and Augmented Reality*, pages 151– 160, Tokyo, Japan, Oct. 2003. doi: 10.1109/ISMAR.2003.1240698.

W. Gander, G. H. Golub, and R. Strebel. Least-squares fitting of circles and ellipses. *BIT Numerical Mathematics*, 4(4):558–578, Dec. 1994. doi: 10.1007/BF01934268.

P. Garofalo. *Development of motion analysis protocols based on inertial sensors*. PhD thesis, University of Bologna, 2010.

P. Gemeiner, P. Einramhof, and M. Vincze. Simultaneous motion and structure estimation by fusion of inertial and vision data. *International Journal of Robotics Research*, 26(6):591–605, 2007. doi: 10.1177/0278364907080058.

S. Gezici, Z. Tian, G. Giannakis, H. Kobayashi, A. Molisch, H. Poor, and Z. Sahinoglu. Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks. *IEEE Signal Processing Magazine*, 22(4):70–84, July 2005. doi: 10.1109/MSP.2005.1458289.

G. H. Golub and R. J. Plemmons. Large-scale geodetic least-squares adjustment by dissection and orthogonal decomposition. *Linear Algebra and its Applications*, 34:3–28, dec 1980. doi: 10.1016/0024-3795(80)90156-1.

G. H. Golub and C. F. Van Loan. *Matrix computations*. Johns Hopkins University Press, Baltimore, MD, USA, 3rd edition, 1996. ISBN 0801854148.

N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proceedings on Radar and Signal Processing*, 140(2):107–113, Apr. 1993. ISSN 0956-375X.

S. Graebe. *Theory and Implementation of Gray Box Identification*. PhD thesis, Royal Institute of Technology, Stockholm, Sweden, June 1990.

M. Grewal and A. Andrews. How good is your gyro. *IEEE Control Systems Magazine*, 30(1):12–86, feb 2011. doi: 10.1109/MCS.2009.935122.

G. Grisetti, R. Kümmerle, C. Stachniss, U. Frese, and C. Hertzberg. Hierarchical optimization on manifolds for online 2d and 3d mapping. In *Proceedings of IEEE International Conference on Robotics and Automation*, pages 273–278, Anchorage, Alaska, May 2010. doi: 10.1109/ROBOT.2010.5509407.

F. Gustafsson. *Statistical Sensor Fusion*. Studentlitteratur, Lund, Sweden, 2010. ISBN 9789144054896.

F. Gustafsson, T. B. Schön, and J. D. Hol. Sensor fusion for augmented reality. In *Proceedings of 17th International Federation of Automatic Control World Congress*, Seoul, South Korea, July 2008. doi: 10.3182/20080706-5-KR-1001.3059.

W. R. Hamilton. On quaternions; or on a new system of imaginaries in algebra. *Philosophical Magazine*, xxv, 1844.

C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, pages 147–151, Manchester, UK, 1988.

R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, 2004. ISBN 0521540518.

A. Hassibi and S. Boyd. Integer parameter estimation in linear models with applications to GPS. *IEEE Transactions on Signal Processing*, 46(11):2938–295, nov 1998. doi: 10.1109/78.726808.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, 2nd edition, 2009. ISBN 978-0-387-84857-0.

G. Hendeby, J. D. Hol, R. Karlsson, and F. Gustafsson. A graphics processing unit implementation of the particle filter. In *Proceedings of European Signal Processing Conference*, Poznań, Poland, Sept. 2007.

R. Hirokawa and T. Ebinuma. A low-cost tightly coupled GPS/INS for small UAVs augmented with multiple GPS antennas. *Navigation*, 56(1):35–44, 2009.

J. D. Hol. *Pose Estimation and Calibration Algorithms for Vision and Inertial Sensors*. Licentiate thesis no 1379, Department of Electrical Engineering, Linköping University, Sweden, May 2008. ISBN 978-91-7393-862-4.

J. D. Hol, T. B. Schön, and F. Gustafsson. On resampling algorithms for particle filters. In *Proceedings of Nonlinear Statistical Signal Processing Workshop*, Cambridge, UK, Sept. 2006a. doi: 10.1109/NSSPW.2006.4378824.

J. D. Hol, T. B. Schön, F. Gustafsson, and P. J. Slycke. Sensor fusion for augmented reality. In *Proceedings of 9th International Conference on Information Fusion*, Florence, Italy, July 2006b. doi: 10.1109/ICIF.2006.301604.

J. D. Hol, T. B. Schön, H. Luinge, P. J. Slycke, and F. Gustafsson. Robust real-time tracking by fusing measurements from inertial and vision sensors. *Journal of Real-Time Image Processing*, 2(2):149–160, Nov. 2007. doi: 10.1007/s11554-007-0040-2.

J. D. Hol, T. B. Schön, and F. Gustafsson. Relative pose calibration of a spherical camera and an IMU. In *Proceedings of 8th International Symposium on Mixed and Augmented Reality*, pages 21–24, Cambridge, UK, Sept. 2008a. doi: 10.1109/ISMAR.2008.4637318.

J. D. Hol, T. B. Schön, and F. Gustafsson. A new algorithm for calibrating a combined camera and IMU sensor unit. In *Proceedings of 10th International Conference on Control, Automation, Robotics and Vision*, pages 1857–1862, Hanoi, Vietnam, Dec. 2008b. doi: 10.1109/ICARCV.2008.4795810.

J. D. Hol, F. Dijkstra, H. Luinge, and T. B. Schön. Tightly coupled UWB/IMU pose estimation. In *Proceedings of IEEE International Conference on Ultra-Wideband*, pages 688–692, Vancouver, Canada, Sept. 2009. doi: 10.1109/ICUWB.2009.5288724. Best student paper award.

J. D. Hol, H. J. Luinge, and P. J. Slycke. Positioning system calibration. Patent Application, Mar. 2010a. US 12/749494.

J. D. Hol, T. B. Schön, and F. Gustafsson. Modeling and calibration of inertial and vision sensors. *International Journal of Robotics Research*, 29(2):231–244, feb 2010b. doi: 10.1177/0278364909356812.

J. D. Hol, T. B. Schön, and F. Gustafsson. Ultra-wideband calibration for indoor positioning. In *Proceedings of IEEE International Conference on Ultra-Wideband*, volume 2, pages 1–4, Nanjing, China, Sept. 2010c. doi: 10.1109/ICUWB.2010.5616867.

J. D. Hol, F. Dijkstra, H. J. Luinge, and P. J. Slycke. Tightly coupled UWB/IMU pose estimation system and method. Patent Application, Feb. 2011. US 2011/0025562 A1.

B. K. P. Horn. Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America A*, 4(4):629–642, Apr. 1987. doi: 10.1364/JOSAA.4.000629.

J. Humpherys and J. West. Kalman filtering with Newton's method. *IEEE Control Systems Magazine*, 30(6):101–106, dec 2010. doi: 10.1109/MCS.2010.938485.

IEEE Std 1559. IEEE standard for inertial systems terminology. Technical report, IEEE, 2009. doi: 10.1109/IEEESTD.2009.5226540.

A. H. Jazwinski. *Stochastic processes and filtering theory*. Mathematics in science and engineering. Academic Press, New York, USA, 1970. ISBN 978-0123815507.

E. S. Jones and S. Soatto. Visual–inertial navigation, mapping and localization: A

scalable real-time causal approach. *International Journal of Robotics Research*, 30(4):407–430, 2011. doi: 10.1177/0278364910388963.

M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 25(6):1365–1378, dec 2008. doi: 10.1109/TRO.2008.2006706.

T. Kailath, A. H. Sayed, and B. Hassibi. *Linear Estimation*. Prentice-Hall, Inc, 2000. ISBN 0-13-022464-2.

R. E. Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME, Journal of Basic Engineering*, 82:35–45, 1960.

E. D. Kaplan and C. Hegarty, editors. *Understanding GPS: Principles and Applications*. Artech House, 2nd edition, 2006. ISBN 978-1580538947.

J. Kelly and G. S. Sukhatme. Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration. *International Journal of Robotics Research*, 30(1):56–79, 2011. doi: 10.1177/027836491038280.

S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An interior-point method for large-scale ℓ_1-regularized least squares. *IEEE Journal of Selected Topics in Signal Processing*, 1(4):606–617, Dec. 2007. ISSN 1932-4553. doi: 10.1109/JSTSP.2007.910971.

G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proceedings of 6th International Symposium on Mixed and Augmented Reality*, pages 225–234, Nara, Japan, Nov. 2007. doi: 10.1109/ISMAR.2007.4538852.

G. S. W. Klein and T. W. Drummond. Tightly integrated sensor fusion for robust visual tracking. *Image and Vision Computing*, 22(10):769–776, 2004. doi: 10.1016/j.imavis.2004.02.007.

K. Koeser, B. Bartczak, and R. Koch. Robust GPU-assisted camera tracking using free-form surface models. *Journal of Real-Time Image Processing*, 2(2):133–147, Nov. 2007. doi: 10.1007/s11554-007-0039-8.

J. B. Kuipers. *Quaternions and Rotation Sequences*. Princeton University Press, 1999. ISBN 0691102988.

L. Ljung. *System Identification: Theory for the User*. Prentice-Hall, Inc, Upper Saddle River, NJ, USA, 2nd edition, 1999. ISBN 0-13-656695-2.

L. Ljung. Perspectives on system identification. In *Proceedings of 17th International Federation of Automatic Control World Congress*, pages 7172–7184, Seoul, South Korea, July 2008. doi: 10.3182/20080706-5-KR-1001.4277.

J. Lobo and J. Dias. Inertial sensed ego-motion for 3D vision. *Journal of Robotics Systems*, 21(1):3–12, 2004. doi: 10.1002/rob.10122.

J. Lobo and J. Dias. Relative pose calibration between visual and inertial sensors. *International Journal of Robotics Research*, 26(6):561–575, 2007. doi: 10.1177/0278364907079276.

D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, Nov. 2004. doi: 10.1023/B:VISI.0000029664.99615.94.

Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry. *An invitation to 3-D vision – from images to geometric models*. Interdisciplinary Applied Mathematics. Springer-Verlag, 2006. ISBN 0387008934.

J. R. Magnus and H. Neudecker. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. John Wiley & Sons, Ltd, 2nd edition, 1999. ISBN 978-0471986331.

K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1):43–72, Nov. 2005. doi: 10.1007/s11263-005-3848-x.

F. M. Mirzaei and S. I. Roumeliotis. A Kalman filter-based algorithm for IMU-camera calibration: Observability analysis and performance evaluation. *IEEE Transactions on Robotics*, 24(5):1143–1156, Oct. 2008. doi: 10.1109/TRO.2008.2004486.

P. Misra and P. Enge. *Global Positioning System: Signals, Measurements, and Performance*. Ganga-Jamuna Press, Lincoln, MA, USA, 2nd edition, 2006. ISBN 0-9709544-1-7.

J. Nocedal and S. J. Wright. *Numerical optimization*. Springer-Verlag, New York, 2nd edition, 2006. ISBN 0387987932.

M. Ozuysal, P. Fua, and V. Lepetit. Fast keypoint recognition in ten lines of code. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1–8, Minneapolis, MI, June 2007. doi: 10.1109/CVPR.2007.38312.

S. Pittet, V. Renaudin, B. Merminod, and M. Kasser. UWB and MEMS based indoor navigation. *The Journal of Navigation*, 61(3):369–384, July 2008. doi: 10.1017/S0373463308004797.

V. Renaudin, M. H. Afzal, and G. Lachapelle. Complete triaxis magnetometer calibration in the magnetic domain. *Journal of Sensors*, 2010. doi: doi:10.1155/2010/967245.

M. Ribo, M. Brandner, and A. Pinz. A flexible software architecture for hybrid tracking. *Journal of Robotics Systems*, 21(2):53–62, 2004. doi: 10.1002/rob.10124.

D. Roetenberg and H. Luinge. In-use estimation of joint angles and segment lengths using inertial sensors. *Journal of Biomechanics*, 2011. submitted.

A. M. Sabatini, C. Martelloni, S. Scapellato, and F. Cavallo. Assessment of walking features from foot inertial sensing. *IEEE Transactions on Biomedical Engineering*, 52(3):486–494, mar 2005. doi: 10.1109/TBME.2004.840727.

Z. Sahinoglu, S. Gezici, and I. Güvenc. *Ultra-wideband Positioning Systems*. Cambridge University Press, 2008. ISBN 9780521873093.

P. G. Savage. Strapdown inertial navigation integration algorithm design part 1: Attitude algorithms. *Journal of Guidance, Control, and Dynamics*, 21(1):19–28, Jan. 1998. ISSN 0731-5090. doi: 10.2514/2.4228.

A. Sayed, A. Tarighat, and N. Khajehnouri. Network-based wireless location: challenges faced in developing techniques for accurate wireless location information. *IEEE Signal Processing Magazine*, 22(4):24–40, July 2005. doi: 10.1109/MSP.2005.1458275.

D. Scaramuzza, A. Martinelli, and R. Siegwart. A toolbox for easily calibrating omnidirectional cameras. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5695–5701, Beijing, China, Oct. 2006. doi: 10.1109/IROS.2006.282372.

H. M. Schepers, H. F. J. M. Koopman, and P. H. Veltink. Ambulatory assessment of ankle and foot dynamics. *IEEE Transactions on Biomedical Engineering*, 54 (5):895–902, may 2007. doi: 10.1109/TBME.2006.889769.

S. F. Schmidt. Application of state-space methods to navigation problems. *Advances in Control Systems*, 3:293–340, 1966.

S. Sczyslo, J. Schroeder, S. Galler, and T. Kaiser. Hybrid localization using UWB and inertial sensors. In *Proceedings of IEEE International Conference on Ultra-Wideband*, volume 3, pages 89–92, Hannover, Germany, Sept. 2008. doi: 10.1109/ICUWB.2008.4653423.

J. Shi and C. Tomasi. Good features to track. In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition*, pages 593–600, Seattle, WA, June 1994. doi: 10.1109/CVPR.1994.323794.

M. D. Shuster. A survey of attitude representations. *The Journal of the Astronautical Sciences*, 41(4):439–517, Oct. 1993.

Z. Sjanic, M. A. Skoglund, F. Gustafsson, and T. B. Schön. A nonlinear least squares approach to the slam problem. In *Proceedings of 20th International Federation of Automatic Control World Congress*, Milano, Italy, Aug. 2011. Accepted for publication.

I. Skog. *Low-Cost Navigation Systems*. PhD thesis, Royal Institute of Technology, Stockholm, Sweden, dec 2009.

I. Skog, P. Händel, J.-O. Nilsson, and J. Rantakokko. Zero-velocity detection -— an algorithm evaluation. *IEEE Transactions on Biomedical Engineering*, 57(11): 2657–2666, Nov. 2010. doi: 10.1109/TBME.2010.2060723.

STMicroelectronics, 2011. [Online]. Available: http://www.st.com/.

K. H. Strobl and G. Hirzinger. Optimal hand-eye calibration. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4647–4653, Beijing, China, Oct. 2006. doi: 10.1109/IROS.2006.282250.

P. J. G. Teunissen. The least-squares ambiguity decorrelation adjustment: a method for fast GPS integer ambiguity estimation. *Journal of Geodesy*, 70: 65–82, nov 1995. doi: 10.1007/BF00863419.

Time Domain, 2010. [Online]. Available: http://www.timedomain.com/.

D. H. Titterton and J. L. Weston. *Strapdown inertial navigation technology*. IEE radar, sonar, navigation and avionics series. Peter Peregrinus Ltd., Stevenage, UK, 1997. ISBN 0863413587.

H. L. V. Trees. *Detection, Estimation, and Modulation Theory, Part I*. John Wiley & Sons, Ltd, 1968. ISBN 978-0471095170.

B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon. Bundle adjustment — a modern synthesis. In A. Z. Bill Triggs and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, volume 1883 of *Lecture Notes in Computer Science*, pages 153–177. Springer-Verlag, 2000. doi: 10.1007/3-540-44480-7_21.

R. Y. Tsai and R. K. Lenz. A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *IEEE Transactions on Robotics and Automation*, 5(3):345–358, June 1989. doi: 10.1109/70.34770.

Ubisense, 2010. [Online]. Available: http://www.ubisense.net/.

J. Vasconcelos, G. Elkaim, C. Silvestre, P. Oliveira, and B. Cardeira. A geometric approach to strapdown magnetometer calibration in sensor frame. In *IFAC Workshop on Navigation, Guidance, and Control of Underwater Vehicles*, Ireland, Apr. 2008. doi: 10.3182/20080408-3-IE-4914.00031.

G. Welch and E. Foxlin. Motion tracking: no silver bullet, but a respectable arsenal. *IEEE Computer Graphics and Applications*, 22(6):24–38, nov 2002. doi: 10.1109/MCG.2002.1046626.

O. J. Woodman. An introduction to inertial navigation. Technical Report UCAM-CL-TR-696, University of Cambridge, Computer Laboratory, Aug. 2007.

Xsens Technologies, 2010. [Online]. Available: http://www.xsens.com/.

Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, Nov. 2000. doi: 10.1109/34.888718.

PhD Dissertations
Division of Automatic Control
Linköping University

M. Millnert: Identification and control of systems subject to abrupt changes. Thesis No. 82, 1982. ISBN 91-7372-542-0.

A. J. M. van Overbeek: On-line structure selection for the identification of multivariable systems. Thesis No. 86, 1982. ISBN 91-7372-586-2.

B. Bengtsson: On some control problems for queues. Thesis No. 87, 1982. ISBN 91-7372-593-5.

S. Ljung: Fast algorithms for integral equations and least squares identification problems. Thesis No. 93, 1983. ISBN 91-7372-641-9.

H. Jonson: A Newton method for solving non-linear optimal control problems with general constraints. Thesis No. 104, 1983. ISBN 91-7372-718-0.

E. Trulsson: Adaptive control based on explicit criterion minimization. Thesis No. 106, 1983. ISBN 91-7372-728-8.

K. Nordström: Uncertainty, robustness and sensitivity reduction in the design of single input control systems. Thesis No. 162, 1987. ISBN 91-7870-170-8.

B. Wahlberg: On the identification and approximation of linear systems. Thesis No. 163, 1987. ISBN 91-7870-175-9.

S. Gunnarsson: Frequency domain aspects of modeling and control in adaptive systems. Thesis No. 194, 1988. ISBN 91-7870-380-8.

A. Isaksson: On system identification in one and two dimensions with signal processing applications. Thesis No. 196, 1988. ISBN 91-7870-383-2.

M. Viberg: Subspace fitting concepts in sensor array processing. Thesis No. 217, 1989. ISBN 91-7870-529-0.

K. Forsman: Constructive commutative algebra in nonlinear control theory. Thesis No. 261, 1991. ISBN 91-7870-827-3.

F. Gustafsson: Estimation of discrete parameters in linear systems. Thesis No. 271, 1992. ISBN 91-7870-876-1.

P. Nagy: Tools for knowledge-based signal processing with applications to system identification. Thesis No. 280, 1992. ISBN 91-7870-962-8.

T. Svensson: Mathematical tools and software for analysis and design of nonlinear control systems. Thesis No. 285, 1992. ISBN 91-7870-989-X.

S. Andersson: On dimension reduction in sensor array signal processing. Thesis No. 290, 1992. ISBN 91-7871-015-4.

H. Hjalmarsson: Aspects on incomplete modeling in system identification. Thesis No. 298, 1993. ISBN 91-7871-070-7.

I. Klein: Automatic synthesis of sequential control schemes. Thesis No. 305, 1993. ISBN 91-7871-090-1.

J.-E. Strömberg: A mode switching modelling philosophy. Thesis No. 353, 1994. ISBN 91-7871-430-3.

K. Wang Chen: Transformation and symbolic calculations in filtering and control. Thesis No. 361, 1994. ISBN 91-7871-467-2.

T. McKelvey: Identification of state-space models from time and frequency data. Thesis No. 380, 1995. ISBN 91-7871-531-8.

J. Sjöberg: Non-linear system identification with neural networks. Thesis No. 381, 1995. ISBN 91-7871-534-2.

R. Germundsson: Symbolic systems – theory, computation and applications. Thesis No. 389, 1995. ISBN 91-7871-578-4.

P. Pucar: Modeling and segmentation using multiple models. Thesis No. 405, 1995. ISBN 91-7871-627-6.
H. Fortell: Algebraic approaches to normal forms and zero dynamics. Thesis No. 407, 1995. ISBN 91-7871-629-2.
A. Helmersson: Methods for robust gain scheduling. Thesis No. 406, 1995. ISBN 91-7871-628-4.
P. Lindskog: Methods, algorithms and tools for system identification based on prior knowledge. Thesis No. 436, 1996. ISBN 91-7871-424-8.
J. Gunnarsson: Symbolic methods and tools for discrete event dynamic systems. Thesis No. 477, 1997. ISBN 91-7871-917-8.
M. Jirstrand: Constructive methods for inequality constraints in control. Thesis No. 527, 1998. ISBN 91-7219-187-2.
U. Forssell: Closed-loop identification: Methods, theory, and applications. Thesis No. 566, 1999. ISBN 91-7219-432-4.
A. Stenman: Model on demand: Algorithms, analysis and applications. Thesis No. 571, 1999. ISBN 91-7219-450-2.
N. Bergman: Recursive Bayesian estimation: Navigation and tracking applications. Thesis No. 579, 1999. ISBN 91-7219-473-1.
K. Edström: Switched bond graphs: Simulation and analysis. Thesis No. 586, 1999. ISBN 91-7219-493-6.
M. Larsson: Behavioral and structural model based approaches to discrete diagnosis. Thesis No. 608, 1999. ISBN 91-7219-615-5.
F. Gunnarsson: Power control in cellular radio systems: Analysis, design and estimation. Thesis No. 623, 2000. ISBN 91-7219-689-0.
V. Einarsson: Model checking methods for mode switching systems. Thesis No. 652, 2000. ISBN 91-7219-836-2.
M. Norrlöf: Iterative learning control: Analysis, design, and experiments. Thesis No. 653, 2000. ISBN 91-7219-837-0.
F. Tjärnström: Variance expressions and model reduction in system identification. Thesis No. 730, 2002. ISBN 91-7373-253-2.
J. Löfberg: Minimax approaches to robust model predictive control. Thesis No. 812, 2003. ISBN 91-7373-622-8.
J. Roll: Local and piecewise affine approaches to system identification. Thesis No. 802, 2003. ISBN 91-7373-608-2.
J. Elbornsson: Analysis, estimation and compensation of mismatch effects in A/D converters. Thesis No. 811, 2003. ISBN 91-7373-621-X.
O. Härkegård: Backstepping and control allocation with applications to flight control. Thesis No. 820, 2003. ISBN 91-7373-647-3.
R. Wallin: Optimization algorithms for system analysis and identification. Thesis No. 919, 2004. ISBN 91-85297-19-4.
D. Lindgren: Projection methods for classification and identification. Thesis No. 915, 2005. ISBN 91-85297-06-2.
R. Karlsson: Particle Filtering for Positioning and Tracking Applications. Thesis No. 924, 2005. ISBN 91-85297-34-8.
J. Jansson: Collision Avoidance Theory with Applications to Automotive Collision Mitigation. Thesis No. 950, 2005. ISBN 91-85299-45-6.
E. Geijer Lundin: Uplink Load in CDMA Cellular Radio Systems. Thesis No. 977, 2005. ISBN 91-85457-49-3.
M. Enqvist: Linear Models of Nonlinear Systems. Thesis No. 985, 2005. ISBN 91-85457-64-7.
T. B. Schön: Estimation of Nonlinear Dynamic Systems — Theory and Applications. Thesis No. 998, 2006. ISBN 91-85497-03-7.

I. Lind: Regressor and Structure Selection — Uses of ANOVA in System Identification. Thesis No. 1012, 2006. ISBN 91-85523-98-4.

J. Gillberg: Frequency Domain Identification of Continuous-Time Systems Reconstruction and Robustness. Thesis No. 1031, 2006. ISBN 91-85523-34-8.

M. Gerdin: Identification and Estimation for Models Described by Differential-Algebraic Equations. Thesis No. 1046, 2006. ISBN 91-85643-87-4.

C. Grönwall: Ground Object Recognition using Laser Radar Data – Geometric Fitting, Performance Analysis, and Applications. Thesis No. 1055, 2006. ISBN 91-85643-53-X.

A. Eidehall: Tracking and threat assessment for automotive collision avoidance. Thesis No. 1066, 2007. ISBN 91-85643-10-6.

F. Eng: Non-Uniform Sampling in Statistical Signal Processing. Thesis No. 1082, 2007. ISBN 978-91-85715-49-7.

E. Wernholt: Multivariable Frequency-Domain Identification of Industrial Robots. Thesis No. 1138, 2007. ISBN 978-91-85895-72-4.

D. Axehill: Integer Quadratic Programming for Control and Communication. Thesis No. 1158, 2008. ISBN 978-91-85523-03-0.

G. Hendeby: Performance and Implementation Aspects of Nonlinear Filtering. Thesis No. 1161, 2008. ISBN 978-91-7393-979-9.

J. Sjöberg: Optimal Control and Model Reduction of Nonlinear DAE Models. Thesis No. 1166, 2008. ISBN 978-91-7393-964-5.

D. Törnqvist: Estimation and Detection with Applications to Navigation. Thesis No. 1216, 2008. ISBN 978-91-7393-785-6.

P-J. Nordlund: Efficient Estimation and Detection Methods for Airborne Applications. Thesis No. 1231, 2008. ISBN 978-91-7393-720-7.

H. Tidefelt: Differential-algebraic equations and matrix-valued singular perturbation. Thesis No. 1292, 2009. ISBN 978-91-7393-479-4.

H. Ohlsson: Regularization for Sparseness and Smoothness — Applications in System Identification and Signal Processing. Thesis No. 1351, 2010. ISBN 978-91-7393-287-5.

S. Moberg: Modeling and Control of Flexible Manipulators. Thesis No. 1349, 2010. ISBN 978-91-7393-289-9.

J. Wallén: Estimation-based iterative learning control. Thesis No. 1358, 2011. ISBN 978-91-7393-255-4.

D. Ankelhed: On design of low order H-infinity controllers. Thesis No. 1371, 2011. ISBN 978-91-7393-157-1.