

Probabilistic modeling for sensor fusion with inertial measurements

Manon Kok

Cover illustration: Norm of the magnetic field estimated from experimental data using Gaussian process regression as described in Paper F. The experimental setup is shown in Figure 6 in this paper. The cover illustration is an adapted version of Figure 7c.

Linköping studies in science and technology. Dissertations.
No. 1814

Probabilistic modeling for sensor fusion with inertial measurements

Manon Kok

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

ISBN 978-91-7685-621-5

ISSN 0345-7524

Copyright © 2016 Manon Kok

Printed by LiU-Tryck, Linköping, Sweden 2016

To everyone who reads this

Abstract

In recent years, inertial sensors have undergone major developments. The quality of their measurements has improved while their cost has decreased, leading to an increase in availability. They can be found in stand-alone sensor units, so-called inertial measurement units, but are nowadays also present in for instance any modern smartphone, in Wii controllers and in virtual reality headsets.

The term inertial sensor refers to the combination of accelerometers and gyroscopes. These measure the external specific force and the angular velocity, respectively. Integration of their measurements provides information about the sensor's position and orientation. However, the position and orientation estimates obtained by simple integration suffer from drift and are therefore only accurate on a short time scale. In order to improve these estimates, we combine the inertial sensors with additional sensors and models. To combine these different sources of information, also called sensor fusion, we make use of probabilistic models to take the uncertainty of the different sources of information into account. The first contribution of this thesis is a tutorial paper that describes the signal processing foundations underlying position and orientation estimation using inertial sensors.

In a second contribution, we use data from multiple inertial sensors placed on the human body to estimate the body's pose. A biomechanical model encodes the knowledge about how the different body segments are connected to each other. We also show how the structure inherent to this problem can be exploited. This opens up for processing long data sets and for solving the problem in a distributed manner.

Inertial sensors can also be combined with time of arrival measurements from an ultrawideband (UWB) system. We focus both on calibration of the UWB setup and on sensor fusion of the inertial and UWB measurements. The UWB measurements are modeled by a tailored heavy-tailed asymmetric distribution. This distribution naturally handles the possibility of measurement *delays* due to multipath and non-line-of-sight conditions while not allowing for the possibility of measurements arriving *early*, i.e. traveling faster than the speed of light.

Finally, inertial sensors can be combined with magnetometers. We derive an algorithm that can calibrate a magnetometer for the presence of metallic objects attached to the sensor. Furthermore, the presence of metallic objects in the environment can be exploited by using them as a source of position information. We present a method to build maps of the indoor magnetic field and experimentally show that if a map of the magnetic field is available, accurate position estimates can be obtained by combining inertial and magnetometer measurements.

Populärvetenskaplig sammanfattning

När regissören Seth MacFarlane animerade teddybjörnen Ted i den storsäljande filmen med samma namn, lånade han ut inte bara sin röst utan också sin kropp till Ted. Genom att montera en mängd sensorer på kroppen, kan man lagra rörelsemönster digitalt, och sedan spela upp dem i exempelvis en animerad teddybjörn. Tekniken har använts inte bara i en stor mängd filmer, utan också av spelindustrin för att utveckla verklighetstroga avатарer, i medicinsk rehabilitering, och för att analysera och optimera rörelsemönster inom elit-idrott.

Denna avhandling behandlar en rad forskningsproblem kring denna typ av sensorer, av exakt samma modell som användes i filmen Ted. Sensorerna som används är en kombination av så kallade *tröghetssensorer* (eng. *inertial sensors*) sammansatta i små enheter. Varje enhet mäter acceleration inklusive tyngdaccelerationen med accelerometer och rotationshastigheter med gyroskop. Dessa sensorer kan tillsammans ge information om enhetens orientering och position. Här används även andra sensortyper och annan information, såsom matematiska modeller. Eftersom dessa modeller är en förenkling av verkligheten och sensormätningar aldrig är exakta, vill vi kombinera olika informationskällor, och ange hur mycket vi kan lita på varje källa. Detta kallas *sensorfusion* och kan göras med *probabilistiska modeller* som kan representera osäkerhet.

En sådan modell som används för att skatta kroppens rörelser är en biomekanisk modell som beskriver kroppens olika delar och hur dessa kan röra sig. I vår modell är dessa kroppsdelar sammankopplade. Vi antar alltså att personen inte förlorar kroppsdelar under experimenten. Denna typ av information kan användas för att animera teddybjörnen Ted eller för att skapa avатарer i dataspel. Om vi även vill att de ska interagera, till exempel hålla hand, behöver vi veta var de är. För att åstadkomma detta kan vi lägga till positionsmätningar.

En typ av sensor som ofta kombineras med tröghetssensorer är magnetometrar. Dessa mäter magnetfältet och man kan likna den vid en kompass som tillhandahåller information om sensorns orientering. I denna avhandling används magnetometern även för att bestämma sensorns position. Magnetometern mäter om det finns magnetiskt material i till exempel möbler eller i byggnaden. Denna information kan man använda för att avgöra var i byggnaden sensorn befinner sig.

Utvecklingen av tröghetssensorer har gått snabbt de senaste åren. Kvaliteten på mätningarna har ökat samtidigt som kostnaden har minskat, vilket har lett till en ökad tillgänglighet. Idag finns de exempelvis i mobiltelefoner, handkontroller till Wii tv-spel och i virtual reality headsets. Allt detta öppnar upp möjligheter för flera spännande tillämpningar inom detta intressanta forskningsområde. Denna avhandling visar att bra information om orientering och position kan fås genom att kombinera olika sorters mätningar och modeller. Kanske kan det leda till att vi kan animera teddybjörnar i våra egna vardagsrum om några år!

Acknowledgments

“The only reason for time is so that everything doesn’t happen at once.”

– Often attributed to Albert Einstein

The quote above symbolizes one of the things that I sometimes forget: I often try to do everything at once. My supervisor, Thomas Schön, has often had to remind me that this is a bad plan and to tell me that I’m “not allowed” to do more things. The reason that I am trying to do all these things, however, is that I enjoy my work so much. The Automatic Control Group at Linköping University is a very inspiring environment and working in the group has been a truly wonderful experience. I am very grateful to Thomas Schön, Fredrik Gustafsson and Svante Gunnarsson for giving me the opportunity to join the group. I am also grateful for the financial support of MC Impulse, a European Commission, FP7 research project and of CADICS, a Linnaeus Center funded by the Swedish Research Council (VR).

Writing this thesis would not have been as easy without the L^AT_EX template developed and maintained by Henrik Tidefelt and Gustaf Hendeby. Gustaf, your help with L^AT_EX and any other work-related questions, even late in the evenings and during the weekends is very much appreciated. The thesis would not have been in such good shape without the help of my supervisors Thomas Schön, Jeroen Hol and Fredrik Gustafsson. I am also very thankful to Johan Dahlin and Hanna Nyqvist for proofreading my kappa and to Michael Roth, Erik Hedberg, Zoran Sjanic and Jonas Linder for being available to comment on parts of the thesis, even though I asked them only last-minute. The Swedish popular scientific abstract would have neither been as popular scientific nor as Swedish without the great help of Ylva Jung, Fredrik Gustafsson and Thomas Schön. Thanks a lot to you all! Thanks also to Michael Lorenz for sending me brain fuel in the form of Mozartkugeln to give me the energy to write a lot in the past weeks!

I would not have enjoyed these five years as much if I would not have had such a great supervisor. Thomas, your enthusiasm about your work has a great positive influence on the people around you. I am very grateful for all our meetings and all the time that you’ve always made available for me, even when you were busy or when I had to ask a bit too much of your time. You taught me a lot about how to do research, how to write papers, structure my work, etc, etc. Furthermore, the fact that you are so organized has helped structure my work and brought some order in the chaos that I can sometimes create. I am in the lucky situation to also have two co-supervisors, Jeroen Hol and Fredrik Gustafsson. Jeroen, thanks for all the nice technical discussions we have had and for always being welcoming when I was visiting Xsens! Fredrik, thanks for always being available when I needed your help!

I am grateful for having had great collaborations over the past years with Johan Dahlin, Kjartan Halvorsen, Anders Hansson, Alf Isaksson, Daniel Jönsson, Sina Khoshfetrat Pakazad, Joel Kronander, Lennart Ljung, Henk Luinge, Fredrik Olsson, Johan Sjöberg, Arno Solin, Andreas Svensson, Simo Särkkä, David Törnqvist, Jonas Unger, Adrian Wills and Niklas Wahlström. Thanks specifically to Simo and Arno for being so kind as to welcome me for a 2.5 month pre-doc in the Bayesian Methodology Group at Aalto University. Thanks also to Niklas for great

collaborations on our magnetic field papers. I am also very thankful for having gotten the opportunity to supervise the master thesis project by Michael Lorenz, together with Thomas Seel and Philipp Müller from TU Berlin. I hope to have the opportunity for further collaborations with you also in the future!

I consider my colleagues in the group not so much as colleagues but more as my friends. I want to thank all of you for creating such an amazing work environment! Specifically, I would like to thank our head Svante Gunnarsson for making sure that there is always a good atmosphere and Ninna Stensgård for always being there for help with administrative tasks.

I have shared some amazing time with my colleagues over the years, both at work and outside of work. It has always been a great experience to go to conferences. Thanks everyone for also being up for so many fun activities around the conferences such as going on safari in Kruger park, wine tasting around Cape Town, whale watching in Vancouver, diving in Malaysia, and eating lots of dumplings in Beijing! Thanks also for all the nice times we shared in Linköping. Thanks to Sina Khoshfetrat Pakazad for always being the one to arrange fun things to do during weekends and evenings, thanks to Zoran Sjanic for always arranging for drinking beer on Wednesday's and to Marek Sylдатk for making our corridor more lively. Thanks Johan Dahlin for being a great friend and for always being there to answer all of my questions. Thanks also to Zoran Sjanic, Ylva Jung, Jonas Linder, Niklas Wahlström and Erik Hedberg for being there for me when I needed someone to talk to. Last but not least, I would like to thank Hanna Nyqvist, Clas Veibäck, Michael Roth, George Mathai, Gustaf Hendeby, Emre Özkan, Daniel Petersson, Emina Alickovic, Mahdieh Sadabadi, André Carvalho Bittencourt, Martin Lindfors and all the other people from the Automatic Control Group for a wonderful time in the past years!

I would also like to thank my friends from outside our group. Our friendship is an important reason that Linköping feels like home! Furthermore, I would like to thank my family and friends back in the Netherlands for always being welcoming when I go back home and for making sure that I always have a great time when I'm visiting.

Unfortunately, my time as a PhD student is coming to an end. I have learned so much and met so many great people over the past years ... thank you all for this! I'm looking forward to seeing what the future brings!

Linköping, December 2016

Manon Kok

Contents

Notation	xvii
----------	------

I Background

1 Introduction	3
1.1 Background	3
1.2 Additional sensors and models	9
1.2.1 Magnetometers	9
1.2.2 Ultrawideband	10
1.2.3 Biomechanical models	10
1.3 Main contributions	11
1.4 Outline	12
2 Probabilistic models	19
2.1 Models for position and orientation estimation	21
2.2 Maps of the magnetic field	24
2.3 Visualizing the resulting model structures	25
3 Inference	29
3.1 Building maps of the magnetic field	29
3.2 Estimating position and orientation	30
3.3 Estimating calibration parameters	33
4 Conclusions and future work	35
4.1 Position and orientation estimation using inertial sensors	35
4.2 Inertial sensor motion capture	36
4.3 Combining UWB with inertial sensors	38
4.4 Magnetometer calibration	38
4.5 Mapping and localization using magnetic fields	39
4.6 Concluding remarks	40
Bibliography	41

II Publications

A	Using inertial sensors for position and orientation estimation	49
1	Introduction	51
1.1	Background and motivation	52
1.2	Using inertial sensors for pose estimation	55
1.3	Tutorial content and its outline	57
2	Inertial sensors	58
2.1	Coordinate frames	59
2.2	Angular velocity	60
2.3	Specific force	60
2.4	Sensor errors	62
3	Probabilistic models	65
3.1	Introduction	66
3.2	Parametrizing orientation	69
3.3	Probabilistic orientation modeling	75
3.4	Measurement models	77
3.5	Choosing the state and modeling its dynamics	82
3.6	Models for the prior	83
3.7	Resulting probabilistic models	85
4	Estimating position and orientation	88
4.1	Smoothing in an optimization framework	88
4.2	Filtering estimation in an optimization framework	95
4.3	Extended Kalman filtering	97
4.4	Evaluation based on experimental and simulated data	103
4.5	Extending to pose estimation	119
5	Calibration	122
5.1	Maximum a posteriori calibration	123
5.2	Maximum likelihood calibration	124
5.3	Orientation estimation with an unknown gyroscope bias	126
5.4	Identifiability	127
6	Concluding remarks	128
A	Orientation parametrizations	130
A.1	Quaternion algebra	130
A.2	Conversions between different parametrizations	132
B	Pose estimation	133
B.1	Smoothing in an optimization framework	133
B.2	Filtering in an optimization framework	133
B.3	EKF with quaternion states	134
B.4	EKF with orientation deviation states	134
C	Gyroscope bias estimation	135
C.1	Smoothing in an optimization framework	135
C.2	Filtering in an optimization framework	135
C.3	EKF with quaternion states	135
C.4	EKF with orientation deviation states	136
	Bibliography	137

B	An optimization-based approach to motion capture using inertial sensors	145
1	Introduction	147
2	Problem formulation	149
3	Biomechanical model	150
4	Dynamic and sensor models	153
4.1	Dynamic model	153
4.2	Sensor model	154
5	Resulting algorithm	155
6	Experiments	155
7	Conclusions and future work	159
	Bibliography	161
C	A scalable and distributed solution to the inertial motion capture problem	163
1	Introduction	165
2	Problem formulation	168
3	Model	169
3.1	Dynamics of the state $x_t^{S_i}$	170
3.2	Placement of the sensors on the body segments	170
3.3	Biomechanical constraints	171
4	Problem reformulation enabling structure exploitation	171
4.1	Reordering based on time	172
4.2	Reordering based on sensors and body segments	173
5	Tree structure in coupled problems and message passing	175
6	Scalable and distributed solutions using message passing	177
7	Results and discussion	179
8	Conclusions and future work	180
	Bibliography	182
D	Indoor positioning using ultrawideband and inertial measurements	185
1	Introduction	187
2	Related work	189
3	Problem formulation	191
4	Sensor models	192
4.1	Modeling the ultrawideband measurements	193
4.2	Modeling the inertial measurements	194
5	Ultrawideband calibration	195
5.1	Initial estimate: step I	197
5.2	Initial estimate: step II - multilateration	198
5.3	Resulting calibration algorithm	199
6	Sensor fusion	199
7	Experimental results	201
7.1	Experimental validation of the asymmetric noise distribution	201
7.2	Calibration	202
7.3	Pose estimation	205

8	Conclusions and future work	210
	Bibliography	211
E	Magnetometer calibration using inertial sensors	215
1	Introduction	217
2	Related work	219
3	Problem formulation	220
4	Magnetometer measurement model	221
5	Calibration algorithm	224
5.1	Optimization algorithm	225
5.2	Evaluation of the cost function	225
5.3	The parameter vector θ	226
6	Finding good initial estimates	227
6.1	Ellipsoid fitting	227
6.2	Determine misalignment of the inertial and magnetometer sensor axes	229
7	Experimental results	230
7.1	Experimental setup	230
7.2	Calibration results	230
7.3	Heading estimation	233
8	Simulated heading accuracy	236
9	Conclusions	238
	Bibliography	240
F	Modeling and interpolation of the ambient magnetic field by Gaus- sian processes	243
1	Introduction	245
2	Related work	248
3	The ambient magnetic field	250
4	Modeling the magnetic field using Gaussian process priors	251
4.1	Gaussian process regression	252
4.2	Interpolation of magnetic fields	254
4.3	Separate modeling of the magnetic field components	254
4.4	Modeling the magnetic field as the gradient of a scalar po- tential	256
5	Efficient GP modeling of the magnetic field	257
5.1	Reduced-rank GP modeling	257
5.2	Batch estimation	259
5.3	Sequential estimation	261
5.4	Spatio-temporal modeling	261
6	Experiments	263
6.1	Simulated experiment	263
6.2	Empirical proof-of-concept data	264
6.3	Mapping the magnetic field in a building	268
6.4	Online mapping	270
7	Discussion	273

8	Conclusion	275
	Bibliography	277
G	MEMS-based inertial navigation based on a magnetic field map	283
1	Introduction	285
2	Models	286
	2.1 Dynamical model	287
	2.2 Magnetometer measurement model	289
	2.3 Some additional words about the magnetic field model	290
3	Computing the estimate	290
	3.1 RBPF-MAP	291
4	Experimental results	293
	4.1 Experimental setup	293
	4.2 Results	294
5	Conclusions and future work	294
	Bibliography	296

Notation

ABBREVIATIONS

Abbreviation	Meaning
BFGS	Broyden-Fletcher-Goldfarb-Shanno
EKF	Extended Kalman filter
GP	Gaussian process
GPS	Global positioning system
IMU	Inertial measurement unit
KF	Kalman filter
LS	Least squares
MAP	Maximum a posteriori
MEKF	Multiplicative extended Kalman filter
MEMS	Micro-machined electromechanical system
MHE	Moving horizon estimation
ML	Maximum likelihood
NLOS	Non-line-of-sight
NLS	Nonlinear least squares
PDF	Probability density function
PDR	Pedestrian dead-reckoning
PF	Particle filter
PF-MAP	Maximum a posteriori estimate for the particle filter
RMS	Root mean square
RMSE	Root mean square error
RTS	Rauch-Tung-Striebel
RBPF	Rao-Blackwellized particle filter
RBPF-MAP	Maximum a posteriori estimate for the Rao-Blackwellized particle filter
SLAM	Simultaneous localization and mapping
SQP	Sequential quadratic programming
TOA	Time of arrival
TDOA	Time difference of arrival
UWB	Ultrawideband
VR	Virtual reality

SYMBOLS AND OPERATORS

Notation	Meaning
x_t	Vector x at time t
$x_{1:N}$	Vector x from time $t = 1$ to $t = N$
$\hat{x}$	Estimate of x
x^u	Vector x expressed in the u -frame
R^{uv}	Rotation matrix from the v -frame to the u -frame
$\odot$	Quaternion multiplication
q^L	Left quaternion multiplication of the quaternion q
q^R	Right quaternion multiplication of the quaternion q
q_v	Vector part of the quaternion q
$\mathbb{R}$	Set of real numbers
$\text{SO}(3)$	Special orthogonal group in three dimensions
$\det A$	Determinant of the matrix A
$\text{Tr } A$	Trace of the matrix A
A^T	Transpose of the matrix A
$\times$	Cross product
$[a \times]$	Cross product matrix of the vector a
$\otimes$	Kronecker product
A^{-1}	Inverse of the matrix A
$A^\dagger$	Pseudo-inverse of the matrix A
$\mathcal{N}(\mu, \sigma^2)$	Gaussian distribution with mean μ and covariance σ^2
$\text{Cauchy}(\mu, \gamma)$	Cauchy distribution with location parameter μ and scale parameter γ
$\mathcal{U}(a, b)$	Uniform distribution on the interval $[a, b]$
$\mathcal{GP}(\mu, k)$	Gaussian process with mean μ and covariance function k
$p(\cdot)$	Probability density function
$p(a b)$	Conditional probability of a given b
$p(a, b)$	Joint probability of a and b
$\sim$	Is distributed according to
$\mathbb{E}$	Expected value
cov	Covariance
$\mathcal{I}_n$	Identity matrix of size $n \times n$
$0_{m \times n}$	Zero matrix of size $m \times n$
$\triangleq$	Defined as
$\emptyset$	Empty set
$\in$	Is a member of
$A \subseteq B$	A is a subset of or is included in B
$\arg \max$	Maximizing argument
$\arg \min$	Minimizing argument
$\ a\ _2$	Two-norm of the vector a

Part I

Background

1

Introduction

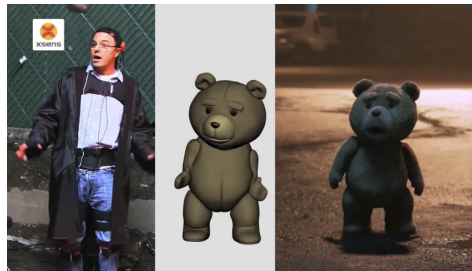
In this thesis, we consider the problem of estimating position and orientation using inertial sensors. In Section 1.1, we give some example applications and introduce what inertial sensors are and what their measurements look like. We will also discuss why inertial sensors typically need to be combined with additional sensors or models to obtain accurate position and orientation information. Examples of additional sensors and models used in this thesis are given in Section 1.2. In Sections 1.3 and 1.4, we will introduce the contributions of the thesis and give an outline of the rest of the thesis.

1.1 Background

Sensors can be used to provide information about the position and orientation of a person or an object. For instance, it is possible to place sensors on a human body to see how the person moves. This information can be useful for rehabilitation or for improving sports performance. An example can be seen in Figure 1.1a where Olympic and world champion speed skating Ireen Wüst wears sensors on her body that give information about her posture while ice skating. One can imagine that she can use this information to analyze which angles her knees and hips should have to skate as fast as possible and if her posture changes when she gets more tired. It is also possible to use the information about how a person moves for motion capture in movies and games, as illustrated in Figure 1.1b, where the actor Seth MacFarlane wears sensors on his body that measure his movements to animate the bear Ted. Sensors can also be placed in or on objects, for example cars, to provide information about their position and orientation as illustrated in Figure 1.1c. This information is for instance useful for self-driving cars. There is a wide range of other examples that one can think of, such as using sensors to



(a) Left: Olympic and world champion speed skating Ireen Wüst wearing sensors on her body. Right: graphical representation of the estimated orientation and position of her body segments.

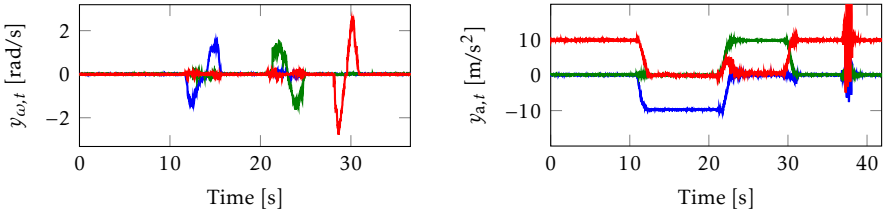


(b) Actor Seth MacFarlane wearing sensors on his body to capture his motion and animate the bear Ted.



(c) Sensors can be used to provide information about the position of the cars in a challenge on cooperative and autonomous driving.

Figure 1.1: Example applications of using sensors to obtain information about the position and orientation of cars and of the various body segments of a person. Courtesy of Xsens Technologies.



(a) Gyroscope measurements $y_{\omega,t}$ in the x - (blue), y - (green) and z -axis (red) of the sensor. (b) Accelerometer measurements $y_{a,t}$ in the x - (blue), y - (green) and z -axis (red) of the sensor.

Figure 1.2: Inertial measurements collected with a smartphone.

obtain information about the position and orientation of robots, unmanned aerial vehicles, trains and people.

The sensors placed on the people and in the cars in Figure 1.1 are *inertial sensors*. The term inertial sensor is used to refer to the combination of accelerometers and gyroscopes. A gyroscope measures the rate of change of the orientation of the sensor, called the *angular velocity*. The gyroscopes that we consider have three axes, implying that they measure the angular velocity in three directions. This is illustrated in Figure 1.2a, which shows gyroscope measurements collected with a Sony Xperia Z5 Compact smartphone using the app described in Hendeby et al. (2014). For the first 10 seconds, the smartphone was lying stationary on a table. Afterwards, the gyroscope was rotated back and forth around its x -, y - and z -axis. An accelerometer measures both the earth's gravity and the acceleration of the sensor. The accelerometers that we consider also have three axes as illustrated in Figure 1.2b. During the first 10 seconds, the smartphone was again lying stationary on a table. The accelerometer measurements can be seen to be around zero in the x - and y -axis, while the z -axis measures a value of around 10 m/s^2 which is due to the earth's gravity. When rotating the smartphone, the accelerometer measures the gravity in different axes. After around 37 seconds, the smartphone was shaken, resulting in a significant acceleration that is measured in addition to the earth's gravity.

Over recent years, inertial sensors have undergone major developments. They have become smaller, lighter and cheaper while providing more accurate measurements. Because of this, they are nowadays available in a large number of devices such as smartphones, Wii controllers and virtual reality (VR) headsets, as shown in Figure 1.3. They are also present in dedicated devices called inertial measurement units (IMUs). The sensor devices placed on the persons and in the cars in Figure 1.1 are IMUs.

Gyroscopes can be used to provide information about the orientation of the sensor, by adding up the changes in orientation over time. This process is called *integration* of the signal. Accelerometers can be used to provide information both about the position and about the orientation of the sensor. If the sensor is not accelerated, the accelerometer measurements can be used to provide information



(a) Left bottom: an Xsens MTx IMU (Xsens Technologies B.V., 2016). Left top: a Trivision Colibri Wireless IMU (Trivision Prototyping GmbH, 2016). Right: a Samsung Galaxy S4 mini smartphone.



(b) A Samsung gear VR.¹



(c) A Wii controller containing an accelerometer and a Motion-Plus expansion device containing a gyroscope.²

Figure 1.3: Examples of devices containing inertial sensors.

¹ 'Samsung Gear VR' available at flic.kr/photos/pestoverde/15247458515 under CC BY 2.0 (<http://creativecommons.org/licenses/by/2.0>).

² 'WiiMote with MotionPlus' by Asmodai available at https://commons.wikimedia.org/wiki/File:WiiMote_with_MotionPlus.JPG under CC BY SA (<https://creativecommons.org/licenses/by-sa/3.0/>).

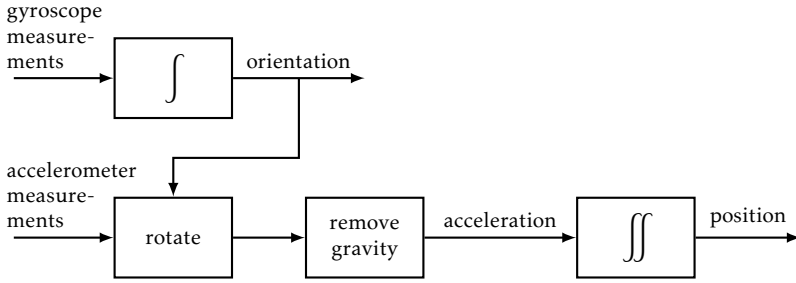
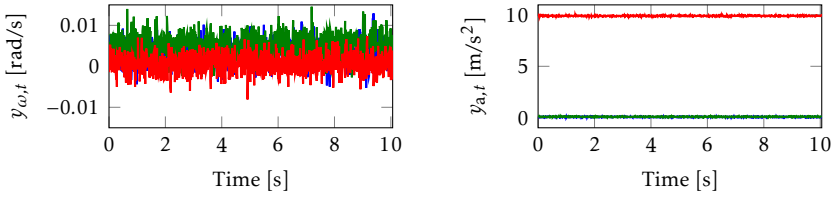


Figure 1.4: Schematic illustration of dead-reckoning, where accelerometer and gyroscope measurements are integrated to position and orientation.

about the orientation of the sensor, because they measure the direction of the earth's gravity with respect to the axes of the sensor. If the sensor is accelerated, the measurements provide information about the change in velocity, which in turn provides information about the change in position. Hence, to obtain position information from the acceleration of the sensor, the signal needs to be integrated twice. To be able to distinguish between the acceleration of the sensor and the earth's gravity, the orientation needs to be known so that the gravity component can be subtracted from the measurements. Because of this, when using inertial sensors, the estimation of the sensor's position is inextricably linked to the estimation of its orientation. The process of integrating the inertial sensor measurements to obtain position and orientation information is often called *dead-reckoning*. This process is summarized in Figure 1.4.

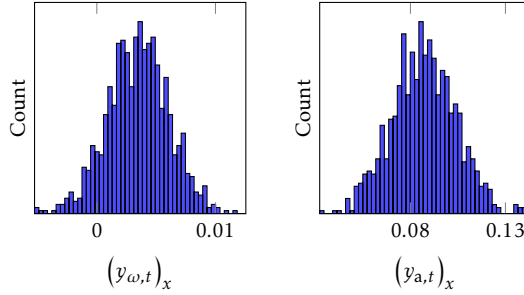
In practice, the position and orientation estimates obtained using dead-reckoning are only accurate for a short time. The reason is that the gyroscope and accelerometer measurements are both biased and noisy, as illustrated in Figure 1.5, where we zoom in on the first 10 seconds of the data shown in Figure 1.2. Because of this, the integration steps from angular velocity to rotation and from acceleration to position introduce *integration drift*. The integration drift in orientation for simulated gyroscope data is illustrated in Figure 1.6. This simulated data has the same bias as the gyroscope measurements in Figure 1.5a, and the same spread in the noise. Because of the constant bias, the orientation error grows linearly with time. The different lines in Figure 1.6 represent the orientation error for different realizations of this noise. The variation in the orientation error for different noise realizations increases over time. The integration drift is more severe for position, which relies both on double integration of the acceleration and on accurate orientation estimates to subtract the earth's gravity.

Because the process of dead-reckoning only gives accurate position and orientation information on a short time scale, inertial sensors are typically combined with additional sensors or additional models. In this thesis, we consider two separate problems related to position and orientation estimation using inertial sensors. The first is concerned only with orientation estimation. The three-dimensional orientation can be described in terms of the roll, pitch and yaw or heading angles. The combination of the roll and pitch angles is often also called inclination. In a



(a) Gyroscope measurements $y_{\omega,t}$ in the x- (blue), y- (green) and z-axis (red) the sensor.

(b) Accelerometer measurements $y_{a,t}$ in the x- (blue), y- (green) and z-axis (red) of the sensor.



(c) Histogram of the gyroscope measurements in the x-axis.

(d) Histogram of the accelerometer measurements in the x-axis.

Figure 1.5: The first 10 seconds of the gyroscope and accelerometer measurements shown in Figure 1.2, during which the smartphone is lying stationary on a table (a,b) and the histograms of one of the axes of the gyroscope and of the accelerometer (c,d).

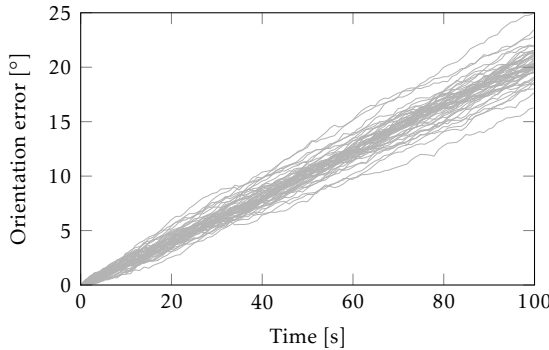


Figure 1.6: Integration of simulated one-dimensional gyroscope measurements to orientation for 50 different noise realizations having the same characteristics as in Figure 1.5a.

second problem, we consider the combined estimation of position and orientation, which is often also called *pose estimation*. In this case, we are interested both in the three-dimensional orientation and in the three-dimensional position.

1.2 Additional sensors and models

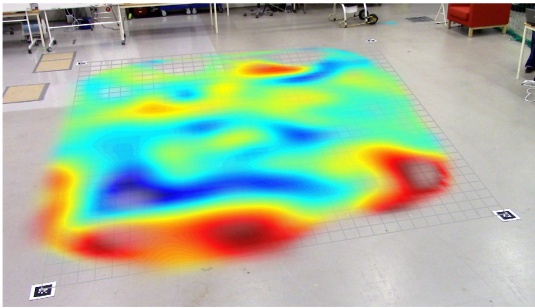
In this section, we will discuss a number of additional sensors and additional models that are used in this thesis to complement the inertial sensors.

1.2.1 Magnetometers

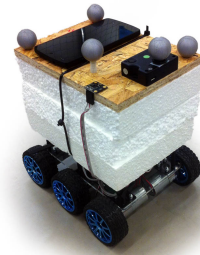
A magnetometer measures the strength and the direction of the magnetic field. The magnetic field consists of contributions both from the local earth magnetic field and from the field due to the presence of magnetic material. The magnitude and the direction of the earth magnetic field depend on the location on the earth. The horizontal component points to the earth magnetic north. The properties of the earth magnetic field are accurately known from geophysical studies, see e.g. National Centers for Environmental Information (2016).

In combination with inertial sensors, magnetometers typically serve the purpose of a compass and are used to provide information about the sensor's heading. This relies on the assumption that the magnetic field is at least locally constant and that it points in the direction of a local magnetic north. There are two reasons why this assumption is frequently violated in practice. Firstly, the sensor can be mounted such that it is rigidly attached to magnetic material. This is for instance the case when the magnetometer is integrated in a smartphone or when it is placed in a car. Secondly, objects containing magnetic material can be present in the vicinity of the sensor, specifically in indoor environments. For instance, there is typically a large amount of magnetic material present in the structures of buildings and in the furniture present in the building.

If the magnetic material is rigidly attached to the sensor, the magnetometer can be calibrated for the presence of this material. Afterwards, the measurements can be used for heading estimation as if the material was not present. The presence of magnetic material in the vicinity of the sensor, however, can not be calibrated for and is typically considered an undesired disturbance. An alternative view is that the presence of magnetic material in indoor environments can be exploited by using it as a source of position information, see e.g. Angermann et al. (2012); Frassl et al. (2013); Solin et al. (2016). This can be done by building a *map* of the magnetic field. Both information about the strength and about the direction of the field can be included in the map. An example of an indoor magnetic field map is shown in Figure 1.7a. It is built from data collected using the mobile robot shown in Figure 1.7b. After the map has been constructed, magnetometer measurements can be compared to it in order to obtain information about possible sensor locations, see e.g. Solin et al. (2016) and Paper G.



(a) Map of the magnitude of the indoor magnetic field.



(b) Mobile robot.

Figure 1.7: Left: Magnitude of an indoor magnetic field estimated using the method presented in Paper F. Right: Mobile robot that was used to collect data.

1.2.2 Ultrawideband

Time of arrival (TOA) measurements from an ultrawideband (UWB) system can be used to provide information about the position of the sensor. UWB is a radio technology which uses a very large frequency band. An example of a UWB system consisting of a number of stationary UWB receivers and a number of small, mobile transmitters is depicted in Figure 1.8a. Each UWB transmitter sends out a UWB pulse as illustrated in Figure 1.8b. The pulse travels with the speed of light towards the receivers, which each measure when the pulse arrives. Combining the measurements from different receivers, it is possible to obtain an estimate of the position of the transmitter. Note that the time when the pulses arrive needs to be measured with very high accuracy. For instance, if the transmitter is 10 meters away from the receiver, it will take the pulse only approximately 33 nanoseconds to reach the receiver.

1.2.3 Biomechanical models

In the examples shown in Figures 1.1a and 1.1b, multiple IMUs are placed on the human body to estimate its movements. More specifically, the IMUs are placed on a large number of body segments and the position and the orientation of each body segment is estimated. This is schematically illustrated in Figure 1.9a. The two body segments can be thought of as the upper and the lower leg, each having an IMU attached to it. The sensors are attached as rigidly as possible to the body segments. This is illustrated in Figure 1.9b, which shows a suit containing 17 IMUs. The suit is meant to be a tight fit such that the sensors move as little as possible with respect to the body. For this application, knowledge about how the human body can move is available to complement the inertial measurements. For instance, the different body segments are known to be connected to each other. This can be captured in biomechanical models.

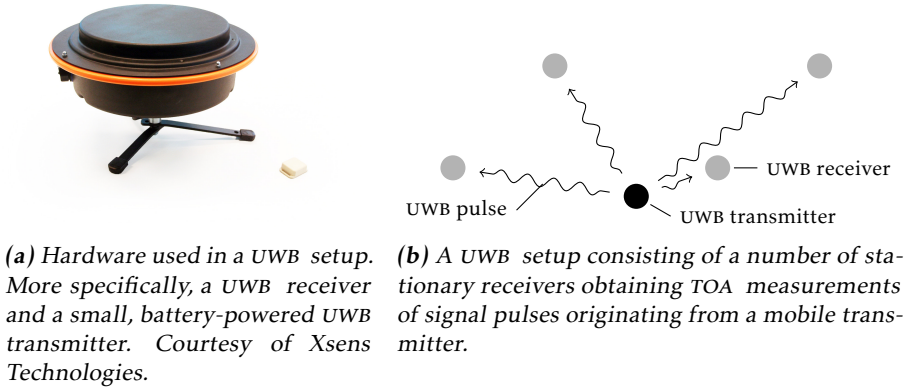


Figure 1.8: Illustration of the TOA measurements and the hardware used in a UWB setup.

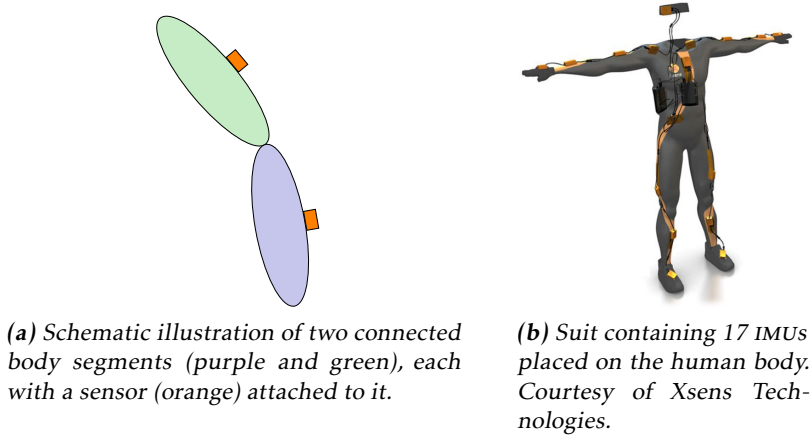


Figure 1.9: Illustration of using IMUs placed on the human body to estimate its movements.

1.3 Main contributions

In this thesis, inertial sensors are combined with additional sensors and additional models for position and orientation estimation. Examples of sensors and models that can be used for this were discussed in Section 1.2. The choice of these examples was highly inspired by the contributions of this thesis. In short, these contributions are:

- A tutorial paper describing the signal processing foundations, i.e. the algorithms and models, underlying position and orientation estimation using inertial sensors [Paper A].
- An approach to estimate the pose of the human body using inertial sensors placed on the body, as illustrated in Figure 1.9 [Paper B]. We also present a method that allows us to solve this problem for large data sets. The same approach can be used to distribute the computations needed to solve the problem over the sensors on the body [Paper C].
- An approach to combine inertial measurements with TOA measurements from a UWB system for indoor positioning. We provide solutions to the pose estimation problem using inertial and UWB measurements, and to the calibration of the UWB setup shown in Figure 1.8 [Paper D].
- We have developed a magnetometer calibration algorithm which uses inertial sensors to calibrate the magnetometer for the presence of magnetic disturbances attached to the sensor. It also calibrates for magnetometer sensor errors and for misalignment between the magnetometer and the inertial sensor axes [Paper E].
- An approach to build maps of the indoor magnetic field, taking into account the well-known physical properties of the magnetic field [Paper F]. An example of a magnetic field map obtained using this method is illustrated in Figure 1.7. We also show that the magnetic field can be used as a source of position information for an experiment where we generate a known magnetic field [Paper G].

1.4 Outline

The thesis consists of two parts. In Part II, seven papers are presented. The contributions of these papers were discussed in Section 1.3. Below we provide a summary of each paper in Part II together with a discussion of the background and of the author's contributions. A background to these papers is provided in Part I. In this introductory chapter, we have briefly introduced the problem at hand, the sensors and models involved and the contributions of the thesis. To combine these different sources of information, also called *sensor fusion*, we make use of *probabilistic models* to take into account each source of information and its accuracy. In Chapters 2 and 3 we discuss the subjects of probabilistic models and inference using these models. Having introduced these topics we revisit the contributions of the thesis in Chapter 4 and discuss them in more technical detail, followed by a discussion of some directions for future work.

Paper A: Using inertial sensors for position and orientation estimation

Paper A is an edited version of

M. Kok, J. D. Hol, and T. B. Schön. Using inertial sensors for position and orientation estimation. Technical Report LiTH-ISY-R-3093, Department of Electrical Engineering, Linköping University, Linköping, Sweden, December 2016a.

Summary: In recent years, micro-machined electromechanical system (MEMS) inertial sensors (3D accelerometers and 3D gyroscopes) have become widely available due to their small size and low cost. Inertial sensor measurements are obtained at high sampling rates and can be integrated to obtain position and orientation (pose) estimates. These pose estimates are accurate on a short time scale, but suffer from integration drift over longer time scales. To overcome this issue, inertial sensors are typically combined with additional sensors and models. In this tutorial we focus on the signal processing aspects of pose estimation using inertial sensors, discussing different modeling choices and a selected number of important algorithms. These algorithms are meant to provide the reader with a starting point to implement their own pose estimation algorithm. The algorithms include optimization-based smoothing and filtering as well as computationally cheaper extended Kalman filter implementations.

Background and contributions: A couple of years ago, Prof. Thomas Schön came up with the idea of writing a tutorial paper on pose estimation using inertial sensors. Towards the end of the PhD of the author of this thesis, the plans for writing this paper became more concrete since it is a nice way of rounding up the work we have done together in the past years. The paper has been written together with Dr. Jeroen Hol.

Paper B: An optimization-based approach to motion capture using inertial sensors

Paper B is an edited version of

M. Kok, J. D. Hol, and T. B. Schön. An optimization-based approach to human body motion capture using inertial sensors. In *Proceedings of the 19th World Congress of the International Federation of Automatic Control*, pages 79–85, Cape Town, South Africa, August 2014.

Summary: In inertial human motion capture, a multitude of body segments are equipped with inertial measurement units, consisting of 3D accelerometers, 3D gyroscopes and 3D magnetometers. Relative position and orientation estimates can be obtained using the inertial data together with a biomechanical model. In this work we present an optimization-based solution to magnetometer-free inertial motion capture. It allows for natural inclusion of biomechanical constraints, for handling of nonlinearities and for using all data in obtaining an estimate. As a proof-of-concept we apply our algorithm to a lower body configuration, illustrating that the estimates are drift-free and match the joint angles from an optical reference system.

Background and contributions: The co-authors Dr. Jeroen Hol and Prof. Thomas Schön came up with the idea of solving the human body motion capture

problem as an optimization problem. The implementation of the optimization algorithm has been done using a framework developed by Xsens Technologies. With this framework, it is possible to define the optimization problem at a high level. The author of this thesis has been involved in developing and implementing the algorithm, in the data collection and has written a major part of the paper.

Paper C: A scalable and distributed solution to the inertial motion capture problem

Paper C is an edited version of

M. Kok, S. Khoshfetrat Pakazad, T. B. Schön, A. Hansson, and J. D. Hol. A scalable and distributed solution to the inertial motion capture problem. In *Proceedings of the 19th International Conference on Information Fusion*, pages 1348–1355, Heidelberg, Germany, July 2016b.

Summary: In inertial motion capture, a multitude of body segments are equipped with inertial sensors, consisting of 3D accelerometers and 3D gyroscopes. Using an optimization-based approach to solve the motion capture problem allows for natural inclusion of biomechanical constraints and for modeling the connection of the body segments at the joint locations. The computational complexity of solving this problem grows both with the length of the data set and with the number of sensors and body segments considered. In this work, we present a scalable and distributed solution to this problem using tailored message passing, capable of exploiting the structure that is inherent in the problem. As a proof-of-concept we apply our algorithm to data from a lower body configuration.

Background and contributions: This work solves the inertial motion capture problem from Paper B using the message passing algorithm developed by Khoshfetrat Pakazad et al. (2016). After the author of this thesis presented the inertial motion capture problem during an internal group meeting, Dr. Sina Khoshfetrat Pakazad suggested that the structure of the motion capture problem can be exploited using the message passing algorithm. The implementation and the writing of the paper has been done together with Dr. Sina Khoshfetrat Pakazad.

Paper D: Indoor positioning using ultrawideband and inertial measurements

Paper D is an edited version of

M. Kok, J. D. Hol, and T. B. Schön. Indoor positioning using ultrawideband and inertial measurements. *IEEE Transactions on Vehicular Technology*, 64(4):1293–1303, 2015b.

Summary: In this work we present an approach to combine measurements from inertial sensors (accelerometers and gyroscopes) with time of arrival measurements from an ultrawideband system for indoor positioning. Our algorithm

uses a tightly-coupled sensor fusion approach, where we formulate the problem as a maximum a posteriori problem that is solved using an optimization approach. It is shown to lead to accurate 6D position and orientation estimates when compared to reference data from an independent optical tracking system. To be able to obtain position information from the ultrawideband measurements, it is imperative that accurate estimates of the ultrawideband receivers' positions and their clock offsets are available. Hence, we also present an easy-to-use algorithm to calibrate the ultrawideband system using a maximum likelihood formulation. Throughout this work, the ultrawideband measurements are modeled by a tailored heavy-tailed asymmetric distribution to account for measurement outliers. The heavy-tailed asymmetric distribution works well on experimental data, as shown by analyzing the position estimates obtained using the ultrawideband measurements via a novel multilateration approach.

Background and contributions: The co-authors of this paper, Dr. Jeroen Hol and Prof. Thomas Schön, have been working on the subject of indoor positioning using ultrawideband and inertial measurements, resulting in Hol et al. (2009, 2010) and in the results presented in Hol (2011). The author of this thesis has substantially extended and adapted the previously presented algorithms for sensor fusion, calibration and multilateration. The paper has been written together with Dr. Jeroen Hol.

Paper E: Magnetometer calibration using inertial sensors

Paper E is an edited version of

M. Kok and T. B. Schön. Magnetometer calibration using inertial sensors. *IEEE Sensors Journal*, 16(14):5679 – 5689, 2016.

Earlier versions of this work were presented in:

M. Kok and T. B. Schön. Maximum likelihood calibration of a magnetometer using inertial sensors. In *Proceedings of the 19th World Congress of the International Federation of Automatic Control*, pages 92–97, Cape Town, South Africa, August 2014,

M. Kok, J. D. Hol, T. B. Schön, F. Gustafsson, and H. Luinge. Calibration of a magnetometer in combination with inertial sensors. In *Proceedings of the 15th International Conference on Information Fusion*, pages 787–793, Singapore, July 2012.

Summary: In this work we present a practical algorithm for calibrating a magnetometer for the presence of magnetic disturbances and for magnetometer sensor errors. To allow for combining the magnetometer measurements with inertial measurements for orientation estimation, the algorithm also corrects for misalignment between the magnetometer and the inertial sensor axes. The calibration algorithm is formulated as the solution to a maximum likelihood problem and the computations are performed offline. The algorithm is shown to give good

results using data from two different commercially available sensor units. Using the calibrated magnetometer measurements in combination with the inertial sensors to determine the sensor's orientation is shown to lead to significantly improved heading estimates.

Background and contributions: Before the author of this thesis started her work as a PhD student at Linköping University, she worked at Xsens Technologies. During this time she studied the topic of magnetometer calibration. Hence, the magnetometer calibration problem provided a good starting point for research during her PhD. A first paper on this subject has therefore been co-authored by Dr. Jeroen Hol and Dr. Henk Luinge from Xsens Technologies. Later work has mainly been done in cooperation with Prof. Thomas Schön. Dr. Henk Luinge and Laurens Slot from Xsens Technologies and Dr. Gustaf Hendeby from Linköping University have been so kind as to help in collecting the data sets presented in the paper. The author of this thesis has implemented the calibration algorithm and has written a major part of the paper.

Paper F: Modeling and interpolation of the ambient magnetic field by Gaussian Processes

Paper F is an edited version of

A. Solin, M. Kok, N. Wahlström, T. B. Schön, and S. Särkkä. Modeling and interpolation of the ambient magnetic field by Gaussian processes. *ArXiv e-prints*, September 2015. arXiv:1509.04634.

Summary: Anomalies in the ambient magnetic field can be used as features in indoor positioning and navigation. By using Maxwell's equations, we derive and present a Bayesian non-parametric probabilistic modeling approach for interpolation and extrapolation of the magnetic field. We model the magnetic field components jointly by imposing a Gaussian process (GP) prior on the latent scalar potential of the magnetic field. By rewriting the GP model in terms of a Hilbert space representation, we circumvent the computational pitfalls associated with GP modeling and provide a computationally efficient and physically justified modeling tool for the ambient magnetic field. The model allows for sequential updating of the estimate and time-dependent changes in the magnetic field. The model is shown to work well in practice in different applications: we demonstrate mapping of the magnetic field both with an inexpensive Raspberry Pi powered robot and on foot using a standard smartphone.

Background and contributions: This paper has largely been written during the author's PreDoc visit to the Bayesian Methodology Group at Aalto University in January – March 2015. It combines the approaches from Wahlström et al. (2013) and Solin and Särkkä (2014) and builds on the common interest of the authors in localization using magnetic fields as a source of position information. The map of the indoor magnetic field obtained using the method presented in this paper, has been used in Solin et al. (2016) for localization. In the future we hope to find time to combine these ideas into a working simultaneous localization and mapping (SLAM) solution. The work on implementation and writing of the paper

has been split more or less equally between Dr. Arno Solin and the author of this thesis.

Paper G: MEMS-based inertial navigation based on a magnetic field map

Paper G is an edited version of

M. Kok, N. Wahlström, T. B. Schön, and F. Gustafsson. MEMS-based inertial navigation based on a magnetic field map. In *Proceedings of the 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 6466–6470, Vancouver, Canada, May 2013.

Summary: This paper presents an approach for 6D pose estimation where MEMS inertial measurements are complemented with magnetometer measurements assuming that a model (map) of the magnetic field is known. The resulting estimation problem is solved using a Rao-Blackwellized particle filter. In our experimental study the magnetic field is generated by a magnetic coil giving rise to a magnetic field that we can model using analytical expressions. The experimental results show that accurate position estimates can be obtained in the vicinity of the coil, where the magnetic field is strong.

Background and contributions: The idea of looking into pose estimation using magnetometers as a source of position information was started through discussions with Dr. Slawomir Grzonka during the CADICS “Learning World Models” workshop in 2010 in Linköping. The experiments used in the paper were performed while the author of this thesis was working at Xsens Technologies. During this time, a first implementation of the pose estimation algorithm was made, using an extended Kalman filter. During the author’s time at Linköping University, the work has been extended with an implementation using a Rao-Blackwellized particle filter. The author of this thesis wrote a major part of this paper. This paper was the start of our work towards SLAM using magnetic measurements.

Publications of related interest, but not included in this thesis

F. Olsson, M. Kok, K. Halvorsen, and T. B. Schön. Accelerometer calibration using sensor fusion with a gyroscope. In *Proceedings of the IEEE Workshop on Statistical Signal Processing*, pages 660–664, Palma de Mallorca, Spain, June 2016.

M. Kok, J. Dahlin, T. B. Schön, and A. Wills. Newton-based maximum likelihood estimation in nonlinear state space models. In *Proceedings of the 17th IFAC Symposium on System Identification*, pages 398–403, Beijing, China, October 2015a.

A. Svensson, T. B. Schön, and M. Kok. Nonlinear state space smoothing using the conditional particle filter. In *Proceedings of the 17th IFAC Symposium on System Identification*, pages 975–980, Beijing, China, October 2015.

A. J. Isaksson, J. Sjöberg, D. Törnqvist, L. Ljung, and M. Kok. Using horizon estimation and nonlinear optimization for grey-box identification. *Journal of Process Control*, 30:69–79, June 2015.

J. Kronander, J. Dahlin, D. Jönsson, M. Kok, T. B. Schön, and J. Unger. Real-time video based lighting using GPU raytracing. In *Proceedings of the 2014 European Signal Processing Conference (EUSIPCO)*, pages 1627–1631, Lisbon, Portugal, September 2014.

M. Kok. *Probabilistic modeling for positioning applications using inertial sensors*. Licentiate’s thesis no. 1656, Linköping University, Linköping, Sweden, June 2014.

N. Wahlström, M. Kok, T. B. Schön, and F. Gustafsson. Modeling magnetic fields using Gaussian processes. In *Proceedings of the 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 3522 – 3526, Vancouver, Canada, May 2013.

2

Probabilistic models

As discussed in Chapter 1, our interest lies in position and orientation estimation using inertial sensors. For general estimation problems, two key questions need to be answered to set up a description of the problem:

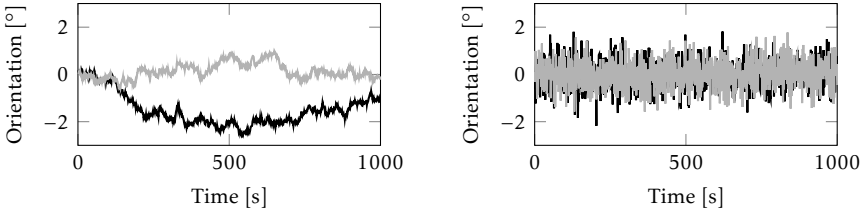
What are we interested in? And which information is available?

For the inertial motion capture problem illustrated in Figure 1.9 for instance, we are interested in estimating the relative position and orientation of each of the body segments. The information that is available are the inertial measurements from each of the 17 IMUs. Furthermore, knowledge is available from biomechanical models. For instance, the body segments are known to be connected to each other.

Our answers to these two key questions will guide us when we *model* the relation between the quantities that we are interested in and the information that is available. It is important to realize that models are *simplifications* of reality, which implies that they are never completely true. Since our sensors are not perfect (see Figure 1.5) and since our models are not perfect descriptions of reality, we typically want to combine multiple sources of information. This is illustrated in Example 2.1.

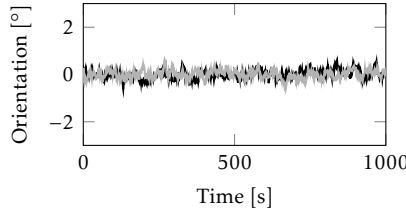
Example 2.1: Estimating orientation using inertial measurements

As described in Chapter 1, the gyroscope measures the angular velocity of the sensor and integration of the measurements provides information about the sensor's orientation. Modeling the accelerometer measurements as measuring only the gravity, its measurements can be used to estimate the inclination of the sensor. In practice, however, the measurements are biased and contain noise, as illustrated in Figure 1.5. We simulate noisy accelerometer and gyroscope measurements, assuming that the sensor is lying still. Note that compared to the data in Figure 1.5,



(a) Inclination estimates using only gyroscope measurements.

(b) Inclination estimates using only accelerometer measurements.



(c) Inclination estimates from combining accelerometer and gyroscope measurements.

Figure 2.1: Estimated inclination using integration of simulated gyroscope measurements (a), by using accelerometer measurements, assuming that the sensor is stationary (b) and by combining the measurements (c). The roll is depicted in black, the pitch in grey.

we have assumed that the measurements do not contain any bias. Furthermore, the noise levels are chosen slightly differently for illustrational purposes.

The inclination estimates obtained by integration of the gyroscope data are shown in Figure 2.1a. Instead of staying around 0° , they drift over time. The inclination estimated from the simulated accelerometer measurements is shown in Figure 2.1b. As can be seen, the orientation estimates are centered around 0° . However, they are quite noisy. We would like to combine the accelerometer and gyroscope measurements to estimate the inclination such that our estimates look as smooth as the ones using the gyroscope data but at the same time do not exhibit any integration drift. An example of our desired outcome is shown in Figure 2.1c.

To effectively combine multiple sources of information, it is beneficial to take the *uncertainty* of the different sources into account. For instance, to obtain Figure 2.1c, we explicitly made use of the knowledge of the noise levels of the (simulated) measurements. This is an important reason for why we are interested in using *probabilistic* models.

We express our models in terms of mathematical relations. For this, we denote all the quantities that we are interested in the *states* x_t or the *parameters* θ . The

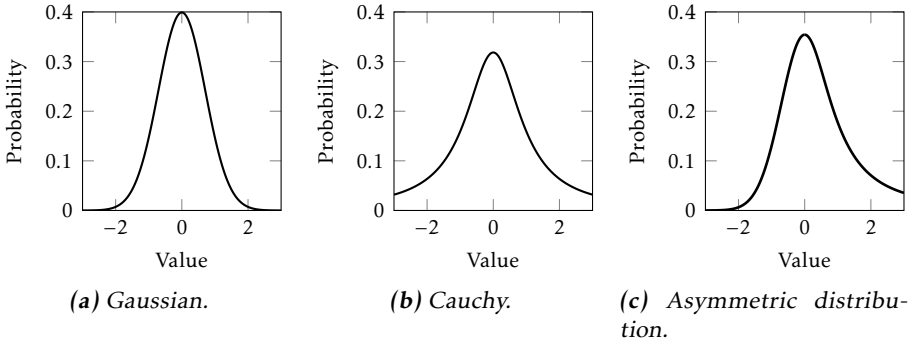


Figure 2.2: A number of probability density functions.

subscript t on x implies that we assume that x *changes over time* and has value x_t at time t . We model the states to be in discrete time from time $t = 1$ to $t = N$. The set of states at all time steps is denoted $x_{1:N}$. In Example 2.1, the state x_t consists of the inclination of the sensor. The parameters θ do not have a subscript t . With this we explicitly indicate that they are *constant*. We will encounter examples of parameters θ in Chapter 3. We denote the measurements at time t by y_t and the set of all measurements from $t = 1, \dots, N$ by $y_{1:N}$. In Example 2.1, the measurements y_t consist of both the gyroscope and the accelerometer measurements.

To take the uncertainty of the states $x_{1:N}$ and the measurements $y_{1:N}$ into account, we represent both the states and the measurements as *random variables* distributed according to some *probability distribution*. Examples of probability distributions that we encounter throughout this thesis are given in Figure 2.2. The Gaussian distribution shown in Figure 2.2a has a *mean* of zero and a *covariance* of one. This implies that the variable is most likely to have a value around 0. In fact, there is a 68% chance that the random variable is between -1 and $+1$ and a 99.7% chance that it is within -3 and $+3$. A general Gaussian distribution with mean μ and covariance Σ is denoted $\mathcal{N}(\mu, \Sigma)$.

For the Gaussian distribution in Figure 2.2a, the probability of the variable to have a value smaller than -3 or larger than $+3$ is very small. The Cauchy distribution shown in Figure 2.2b on the other hand, assigns a larger probability to values deviating more from zero. A distribution that models the probability of large positive values to be higher than the probability of large negative values is shown in Figure 2.2c.

2.1 Models for position and orientation estimation

In this section we discuss a number of probabilistic models to illustrate the types of models that we use for position and orientation estimation in the papers in Part II. We start with an example of a *dynamic model* in Example 2.2. Dynamic

models are used to describe the relation between the state x_{t+1} and x_t as

$$x_{t+1} \mid x_t \sim p(x_{t+1} \mid x_t), \quad (2.1)$$

where $p(\cdot)$ denotes a probability density function. The dynamic model describes the *conditional distribution* of the state x_{t+1} given the state x_t .

Example 2.2: Dynamic model

For almost all applications, we have some knowledge about the *dynamics*. For instance, when estimating the position of a person, it is very unlikely, if not impossible, for the person to be in Linköping, Sweden at a specific time instance, and in Amsterdam, the Netherlands half an hour later. In other words, *conditioned* on the fact that we know that the person is in Linköping at time t , we know something about where the person can be at time $t + 1$.

Since inertial sensors measure the acceleration and the angular velocity of the sensor, they can be used to provide information about the *change* in position and orientation from time t to time $t + 1$. This can be used in a dynamic model. The inertial measurements are both noisy and biased as illustrated in Figure 1.5. Comparing the histograms in Figures 1.5c and 1.5d to the distributions in Figure 2.2, it can be seen that the inertial sensor measurement noise is quite Gaussian with a non-zero mean value (bias) and a covariance that is significantly smaller than one. The presence of Gaussian noise and of a sensor bias can be represented by the probabilistic dynamic model (2.1).

The model discussed in Example 2.2 is used in Papers A – E and Paper G. In some applications, additional knowledge is available about the relation between different parts of the state vector x_t . This can explicitly be modeled in terms of the conditional distribution

$$x_t^a \mid x_t^b \sim p(x_t^a \mid x_t^b), \quad (2.2)$$

where x_t^a and x_t^b are subsets of the states x_t . Two examples related to Papers B and C are discussed in Examples 2.3 and 2.4.

Example 2.3: Sensors placed on body segments

To estimate the pose of the human body, sensors can be placed on different body segments, as discussed in Section 1.2.3. It is not possible to place the sensors directly on the bone. Instead, they are placed on the skin and because of the presence of soft tissue, they will move slightly with respect to the bone. It is difficult to model this movement exactly. Instead, we assume that the position and orientation of the sensors on the body segments are constant up to some Gaussian noise.

Example 2.4: Connection of body segments at joints

When sensors are placed on a human body, it is possible to make use of the knowledge that the body segments are connected to each other at the joints. This assumption is actually *exactly* true. Hence, we would like to model this as a *deterministic constraint* instead of using a probability distribution.

Finally, we can model the knowledge provided by the sensor measurements about the states. This can be represented as

$$y_t \mid x_t \sim p(y_t \mid x_t), \quad (2.3)$$

i.e. in terms of the conditional distribution of the measurements y_t given the state x_t . Examples 2.5 and 2.6 discuss the inclusion of UWB and magnetometer measurements. UWB measurements are used in Paper D, while magnetometers are used in Papers E – G.

Example 2.5: Ultrawideband measurements

In Section 1.2.2, we discussed the use of TOA measurements from a UWB system in combination with inertial sensors. In practice, a small number of pulses sent by the transmitter to the receivers can be *delayed*. This can be because the pulse did not take the shortest path to the receiver, but instead traveled via for instance the floor or a wall in the building. This is called *multipath*. It can also be because the pulse had to travel through some material other than air to reach the receiver. This is called *non-line-of-sight (NLOS)* and causes a delayed pulse since the speed of light in material is lower than the speed of light in air. The presence of a small number of delayed measurements can be modeled by assuming that the TOA measurements y_t given the state x_t are distributed according to an asymmetric distribution such as the one shown in Figure 2.2c. This distribution allows for measurements to be delayed while not allowing for the possibility of measurements arriving *earlier*, i.e. traveling faster than the speed of light.

Example 2.6: Magnetometer measurements

Magnetometers measure the local magnetic field. This field consists of contributions both from the local earth magnetic field and from the magnetic field due to magnetic material such as metallic structures of buildings and furniture. Because of this, especially in indoor environments, it can vary significantly over different locations in the building. Let us define a function $f(p_t^n)$ that gives the magnetic field at each position p_t^n . The magnetometer measurements $y_{m,t}$ can then be modeled as

$$y_{m,t} = R_t^{\text{bn}} f(p_t^n) + e_{m,t}, \quad (2.4)$$

where $e_{m,t}$ is Gaussian measurement noise. The *rotation matrix* R_t^{bn} rotates the magnetic field from the coordinate frame in which the sensor is localized to the coordinate frame in which the sensor obtains its measurements. Note that we

use slightly different notation here compared to (2.3). A subscript m is added to the measurements y_t to explicitly indicate that we consider magnetometer measurements. Furthermore, the state x_t in this case consists of both the position of the sensor p_t^n and the orientation R_t^{bn} .

When the magnetic field is used for heading information, it is typically assumed that the magnetic field is constant, i.e. that $f(p_t^n)$ is a constant three-dimensional vector. Because of this, local variations of the magnetic field are considered undesired disturbances. On the other hand, it is also possible to make use of the changes in the magnetic field to provide position and orientation information. For this we would like to know the function $f(p_t^n)$. In practice, it is typically hard to obtain $f(p_t^n)$ because a large number of magnetic field sources contribute to the magnetic field, severely complicating the modeling process. However, it is possible to estimate the function $f(p_t^n)$ by *learning a map of the magnetic field*. This can be done by collecting *training data*, which can be used to *predict* the magnetic field at previously unknown locations.

The models discussed in Examples 2.2 – 2.6 can be combined and used for position and orientation estimation, which is the topic of Section 3.2. In Section 2.2 we will first discuss a method to build maps of the magnetic field.

2.2 Maps of the magnetic field

In Example 2.6, we introduced the problem of building maps of the magnetic field. An example of a map of the magnetic field is shown in Figure 1.7. The map is obtained by interpolation and extrapolation of magnetic field measurements at different locations, collected by a small robot. Hence, based on a number of measurements, so-called training data, we learn the local magnetic field. This allows us to predict the magnetic field in previously unobserved locations. In Paper F, we build these maps by assuming that the magnetic field can be modeled as a *Gaussian process (GP)*. GPs are defined by Rasmussen and Williams (2006) as:

Definition 2.7. “A GP is a collection of random variables, any finite number of which have a joint Gaussian distribution.”

Consider the slightly more general notation as compared to Example 2.6 and model the measurements y_t as

$$\begin{aligned} y_t &= f(x_t) + e_t, \\ f(x) &\sim \mathcal{GP}(\mu(x), k(x, x')), \end{aligned} \tag{2.5}$$

where $e_t \sim \mathcal{N}(0, \sigma_n^2)$ and $\mathcal{GP}(\mu(x), k(x, x'))$ denotes a GP with mean $\mu(x)$ and covariance $k(x, x')$. Hence, the magnetic field at different locations x_t is jointly

Gaussian distributed as

$$\begin{pmatrix} f(x_1) \\ \vdots \\ f(x_N) \end{pmatrix} \sim \mathcal{N}(\mu, K), \quad (2.6a)$$

with

$$\mu = \begin{pmatrix} \mu(x_1) \\ \vdots \\ \mu(x_N) \end{pmatrix}, \quad K = \begin{pmatrix} K(x_1, x_1) & \cdots & K(x_1, x_N) \\ \vdots & \ddots & \vdots \\ K(x_N, x_1) & \cdots & K(x_N, x_N) \end{pmatrix}. \quad (2.6b)$$

Using GPs, it is possible to incorporate prior information about the physical properties of the magnetic field. This is illustrated in Example 2.8.

Example 2.8: Encoding prior knowledge

The choice of covariance function $k(x, x')$ can encode prior knowledge about the function $f(x)$. Two well-known covariance functions are the squared exponential covariance function $k_{\text{SE}}(x, x')$ and the exponential covariance function $k_{\text{E}}(x, x')$, defined as

$$k_{\text{SE}}(x, x') = \sigma_f^2 \exp\left(-\frac{\|x-x'\|_2^2}{2\ell^2}\right), \quad (2.7a)$$

$$k_{\text{E}}(x, x') = \sigma_f^2 \exp\left(-\frac{\|x-x'\|_2}{\ell}\right). \quad (2.7b)$$

They model the mutual dependence of $f(x)$ on $f(x')$ in terms of the *hyperparameters* σ_f and ℓ . Figures 2.3a – 2.3c show samples drawn from a GP prior using a squared exponential covariance function with different hyperparameters. As can be seen, the parameter σ_f determines the magnitude and the parameter ℓ the length scale. The hyperparameters σ and ℓ therefore influence the shapes of the functions. They can be learned from data. As shown in Figure 2.3d, the choice of covariance function can incorporate prior knowledge for instance about the smoothness of the function.

In Example 2.8, we assumed that x and y are one-dimensional. For the case of modeling the magnetic field, however, the magnetometer measurements $y_{m,t}$ and the position p_t^n are both three-dimensional vectors. This opens up for additional modeling choices, for instance on how these three components are related. Physical knowledge of magnetic fields is available through Maxwell's equations, see also Griffiths (1999); Jackson (1999). We incorporate this into the GP prior in Wahlström et al. (2013) and in Paper F.

2.3 Visualizing the resulting model structures

In this chapter, we have introduced models for two different estimation problems. The first is estimation of position and orientation, for which we have discussed

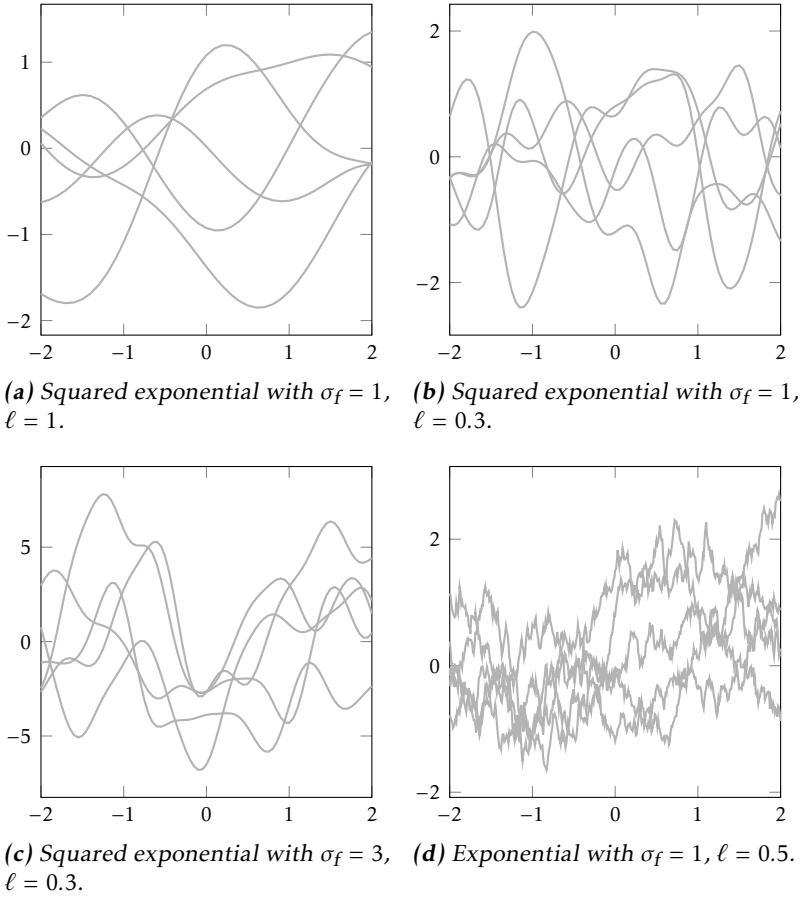


Figure 2.3: Samples from two different GP priors for a number of different hyperparameters.

several models in Section 2.1. The second is estimation of the magnetic field using GPs discussed in Section 2.2. In this section we discuss the structure of these two resulting models and visualize them in terms of their corresponding *graphical models* (Bishop, 2006).

Combining (2.1) and (2.3), the structure of our model for position and orientation estimation is graphically illustrated in Figure 2.4. The state x_{t+1} can be seen to depend on x_t and result in measurements y_{t+1} . Note that x_{t+1} is *conditionally independent* of $x_{1:t-1}$ given the state x_t . This implies that if the sensor's current position, velocity and acceleration are known, it is possible to predict the position and velocity at the next time instance. It is not necessary to know where the sensor has been or how fast it has traveled before arriving in this state. This property of the model is called the *Markov property*. The algorithms used to compute posi-

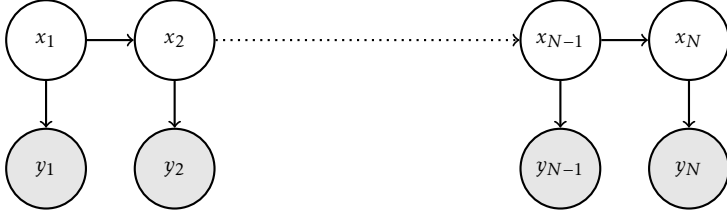


Figure 2.4: Structure of pose estimation problem where x_{t+1} depends on x_t according to (2.1) and results in measurements y_{t+1} according to (2.3).

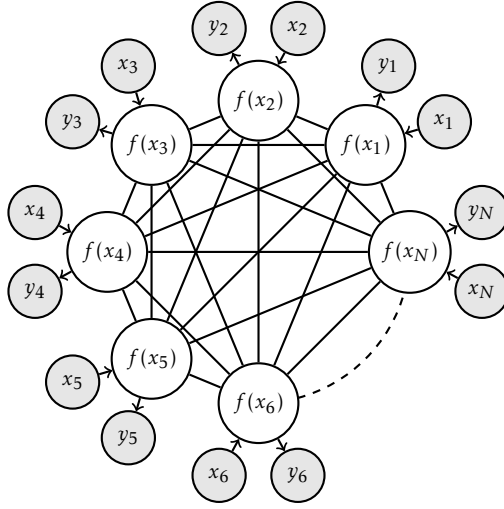


Figure 2.5: Structure of the GP model where noisy measurements y_t of the magnetic field $f(x_t)$ at locations x_t are available.

tion and orientation estimates which will be introduced in Chapter 3 rely on this property.

In the GP model (2.5), noisy measurements y_t of the magnetic field $f(x_t)$ at different locations x_t are available. Having a closer look at the covariance matrix in (2.6), it can be seen that the magnetic field at each location x_t depends on the magnetic field at all other locations, i.e. that all components $K(x, x')$ are non-zero. This is graphically illustrated in Figure 2.5. The fact that $f(x_{1:N})$ are all connected to each other, results in a high computational complexity to build a map of the magnetic field if N is large. Because of this, approaches that use GP models to learn function values $f(x)$ often approximate the model in Figure 2.5, see e.g. Quiñonero-Candela and Rasmussen (2005) for an early survey and Chapter 4 of Bijl (2016) for a more recent overview. The simplest approximation would be to simply discard some of the measurements. An alternative approach is used in Paper F.

3

Inference

In Chapter 2, we focused on modeling the quantities that we are interested in and the information that is available. In this chapter, we will focus on the question:

How can we use the available models and measurements to infer knowledge about the quantities we are interested in?

We will use the models on the forms introduced in Chapter 2 to obtain information about the sensor's position and orientation and about the magnetic field. More formally, our aim is to *infer* information about the states $x_{1:N}$ and the parameters θ using the available models and the measurements $y_{1:N}$. For this, we make extensive use of the basic relations of probabilities, (see e.g. Gut (1995); Bishop (2006)),

$$p(a) = \int p(a, b)db, \quad (3.1a)$$

$$p(a, b) = p(a | b)p(b). \quad (3.1b)$$

where $p(a | b)$ denotes the *conditional probability* of a given b and $p(a, b)$ denotes the *joint probability* of a and b .

3.1 Building maps of the magnetic field

Given a data set with measurements $y = \{y_t\}_{t=1}^N$ at locations $x = \{x_t\}_{t=1}^N$, it is possible to infer knowledge about the magnetic field $f(x_*)$ at some new location x_* using *GP regression*. In the remainder, we will use the short-hand notation f_* to denote $f(x_*)$. Using the fact that y and f_* are jointly Gaussian as

$$p(f_*, y | x, x_*) = \mathcal{N} \left(\begin{pmatrix} y \\ f_* \end{pmatrix}; \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} K(x, x) + \sigma_n^2 & K(x, x_*) \\ K(x_*, x) & K(x_*, x_*) \end{pmatrix} \right), \quad (3.2)$$

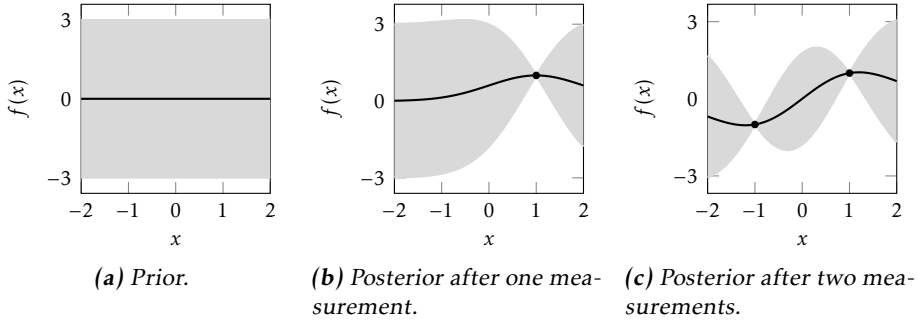


Figure 3.1: Illustration of GP regression where the mean is represented by a black line and the grey area represents the uncertainty.

the conditional distribution $p(f_* | x, y, x_*)$ can be computed. For Gaussian distributions, this conditional distribution can be computed exactly using (3.1b), see e.g. Schön and Lindsten (2011). It results in the Gaussian distribution

$$p(f_* | x, y, x_*) = \mathcal{N}(\mathbb{E}[f_*], \text{cov}[f_*]), \quad (3.3)$$

with

$$\mathbb{E}[f_*] = K(x_*, x) \left(K(x, x) + \sigma_n^2 \mathcal{I}_N \right)^{-1} y, \quad (3.4a)$$

$$\text{cov}[f_*] = K(x_*, x_*) - K(x_*, x) \left(K(x, x) + \sigma_n^2 \mathcal{I}_N \right)^{-1} K(x, x_*). \quad (3.4b)$$

An example of GP regression can be found in Figure 3.1. Before any measurements are observed, the mean of the GP prior is zero and the uncertainty is the same for each position x as illustrated in Figure 3.1a. Observing a measurement y_t at position x_t provides knowledge about the function value $f(x_t)$. This is illustrated in Figures 3.1b and 3.1c. Note that for illustrational purposes we assume that the measurements are noiseless, i.e. $\sigma_n = 0$. Since the function values $f(x)$ at the different locations x depend on each other, the measurements also provide information about the function values at surrounding locations.

3.2 Estimating position and orientation

In this section, we are concerned with inferring knowledge about the states $x_{1:N}$ (containing the position and orientation) from the measurements $y_{1:N}$. This can be expressed in terms of the *joint smoothing distribution*

$$p(x_{1:N} | y_{1:N}). \quad (3.5)$$

In the pose estimation problem, we are typically interested in obtaining *point estimates*, denoted by $\hat{x}_{1:N}$. One way of obtaining these is by solving

$$\begin{aligned}\hat{x}_{1:N} &= \arg \max_{x_{1:N}} p(x_{1:N} \mid y_{1:N}) \\ &= \arg \max_{x_{1:N}} \frac{p(x_{1:N}, y_{1:N})}{p(y_{1:N})} \\ &= \arg \max_{x_{1:N}} p(x_{1:N}, y_{1:N}),\end{aligned}\tag{3.6}$$

where $\arg \max$ denotes the maximizing argument. The first equality sign in (3.6) follows from (3.1) and the second from the fact that $p(y_{1:N})$ in the denominator does not depend on $x_{1:N}$. The estimate $\hat{x}_{1:N}$ is called the *maximum a posteriori* (MAP) estimate. Hence, the MAP estimate $\hat{x}_{1:N}$ is the pose for which the probability of the states given the measurements is maximized.

In (3.6) we assume that all measurements $y_{1:N}$ are used to obtain the posterior distribution of $x_{1:N}$. Although it makes sense to use all available information to obtain the best estimates, a downside of smoothing is that one needs to wait until all measurements are collected before the pose can be computed. Because of this, in many applications, we are also interested in *filtering*, where the aim is to compute the *filtering distribution*

$$p(x_t \mid y_{1:t}).\tag{3.7}$$

The MAP estimate can in this case be computed as

$$\hat{x}_t = \arg \max_{x_t} p(x_t \mid y_{1:t}).\tag{3.8}$$

Using the relations (3.1) in combination with the Markov property discussed in Section 2.3, the full probabilistic model $p(x_{1:N}, y_{1:N})$ in (3.6) can be decomposed as

$$p(x_{1:N}, y_{1:N}) = \underbrace{p(x_1)}_{\text{Prior}} \underbrace{\prod_{t=2}^N p(x_t \mid x_{t-1})}_{\text{Dynamics}} \underbrace{\prod_{t=1}^N p(y_t \mid x_t)}_{\text{Measurements}}.\tag{3.9a}$$

Here, $p(x_1)$ is a *prior distribution* over x_1 . The *dynamics* is modeled in terms of $p(x_{t+1} \mid x_t)$. The distribution $p(y_t \mid x_t)$ models the information given by the measurements about the state. Note that we have encountered explicit examples of these kinds of models in Chapter 2. Similarly, the filtering distribution can be decomposed as

$$p(x_t \mid y_{1:t}) = \frac{\overbrace{p(y_t \mid x_t)}^{\text{Measurements}} \overbrace{p(x_t \mid y_{1:t-1})}^{\text{Prediction}}}{p(y_t \mid y_{1:t-1})},\tag{3.9b}$$

where

$$p(x_t | y_{1:t-1}) = \int \underbrace{p(x_t | x_{t-1})}_{\text{Dynamics}} \underbrace{p(x_{t-1} | y_{1:t-1})}_{\text{Filtering distribution}} dx_{t-1}.$$

If our models in Section 2.1 would be linear and Gaussian, closed form expressions for the MAP estimates of the position and orientation of the sensor could be obtained. For filtering, this results in the well-known *Kalman filter* (Kalman, 1960). For smoothing, this results in the linear smoothing equations, see e.g. Särkkä (2013). Estimation of position and orientation, however, is inherently a nonlinear problem due to the nonlinear nature of orientations. In Examples 3.1 – 3.3 we give a number of examples illustrating how the MAP estimates (3.6) and (3.8) can be obtained for this nonlinear case. The choice of which algorithm to use is highly application-dependent. It depends on the properties of the problem, as well as on the computational resources and the desired accuracy.

Example 3.1: Optimization-based smoothing and filtering

The smoothing and filtering problems can be solved using an *optimization-based approach* (Nocedal and Wright, 2006; Boyd and Vandenberghe, 2004). In this approach, the solutions $\hat{x}_{1:N}$ to (3.6) or $\hat{x}_t$ to (3.8) can be found by studying the *shape* of the smoothing or filtering distributions as a function of the states $x_{1:N}$ or x_t . This can be characterized in terms of their *slope* and *curvature*. If n_x is the size of the state vector x_t , the filtering problem solves N problems of size n_x . The smoothing problem instead solves one problem of size $n_x N$. For example, for orientation estimation the size of x_t is 3. Estimating a smoothing solution of the pose of the lower body for 37 seconds as in Paper B, the state vector has 40 284 elements instead. The framework of optimization naturally allows for including equality and inequality constraints.

Example 3.2: Extended Kalman filters

An alternative approach to solving the filtering problem (3.8) is to use an extended Kalman filter (EKF) (Särkkä, 2013; Gustafsson, 2012). EKFs make a linear approximation of the nonlinear models and use the relations (3.1) to approximate the filtering distribution. They are computationally less expensive than the optimization-based methods from Example 3.1. However, they can not straightforwardly include equality and inequality constraints.

Example 3.3: Particle filters

Particle filters (PFs) approximate the distribution (3.7) using a number of samples and their associated weights. For an introduction to particle filtering, see e.g. Doucet and Johansen (2011). PFs are specifically useful in cases when the models are severely nonlinear and a Gaussian assumption on the state x_t is a poor description. An example when this is the case is localization of the robot in the map in Figure 1.7.

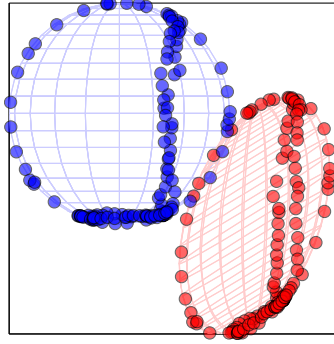


Figure 3.2: Example calibration results with an ellipsoid of magnetometer data before calibration (red) and a sphere of data after calibration (blue).

3.3 Estimating calibration parameters

In Section 3.2, we focused on estimating $x_{1:N}$ from the measurements $y_{1:N}$. We did not consider the presence of unknown parameters θ . Parameter estimation is typically of concern when calibrating the sensors. One approach to estimate constant parameters θ is to straightforwardly include them as additional unknowns in the smoothing optimization problem (3.6), resulting in MAP estimates of the parameters. They can also be included as slowly time-varying parameters in the filtering problem (3.8).

An alternative way of estimating the parameters θ is to solve the *maximum likelihood (ML)* problem defined as

$$\hat{\theta} = \arg \max_{\theta \in \Theta} \mathcal{L}(\theta; y_{1:N}), \quad (3.10)$$

where $\mathcal{L}(\theta; y_{1:N})$ is referred to as the likelihood function, see e.g. Ljung (1999). It is defined as $\mathcal{L}(\theta; y_{1:N}) \triangleq p_{\theta}(Y_{1:N} = y_{1:N})$, where $Y_{1:N}$ are random variables and $y_{1:N}$ are particular realizations of these variables. Hence, $\mathcal{L}(\theta; y_{1:N})$ is a *deterministic* function of a deterministic unknown parameter vector θ . Here, θ is an n_{θ} -dimensional vector which can be limited to a subset Θ of $\mathbb{R}^{n_{\theta}}$. An example of a calibration problem that we encounter in this thesis is calibration of a magnetometer for the presence of magnetic material attached to the sensor, as illustrated in Example 3.4.

Example 3.4: Magnetometer calibration

In Example 2.6, we discussed the use of magnetometers to provide heading information. In that case, the magnetic field at the different locations is assumed to be constant. We will denote this constant magnetic field as m^n . Furthermore, we extend the measurement model (2.4) with calibration parameters D and o as

$$y_{m,t} = DR_t^{\text{bn}} m^n + o + e_{m,t}, \quad (3.11)$$

where $D \in \mathbb{R}^{3 \times 3}$ and $o \in \mathbb{R}^3$. If the sensor would be properly calibrated, rotation of the sensor would lead to a *sphere* of magnetometer data. When magnetic material

is rigidly attached to the sensor, the magnetometer data obtained by rotating the sensor lies on an *ellipsoid* instead. An example based on experimental data is shown in Figure 3.2. Estimating D and o can be interpreted as fitting the red ellipsoid of data to a sphere. Magnetometer calibration is the topic of Paper E.

4

Conclusions and future work

In Chapters 1 – 3 of Part I of this thesis, a background was given to the seven papers that will be presented in Part II. In Chapter 1, an introduction to the sensors and models that we use was given and the contributions of the papers were summarized. Chapter 2 focused on describing the concept of a probabilistic model, illustrated with a few examples of models used in Part II. In Chapter 3, we subsequently discussed how to infer knowledge from these models and the sensor measurements. In the present chapter, we revisit the contributions of this thesis and summarize them in more technical detail. We also discuss possible directions for future work.

4.1 Position and orientation estimation using inertial sensors

Contributions The tutorial in Paper A describes the topic of position and orientation estimation using inertial sensors. Different modeling choices for the dynamics, the measurements and the priors are presented. We introduce smoothing and filtering algorithms solved as optimization problems as well as computationally attractive EKF implementations. The estimates from the different algorithms are compared. Furthermore, some general characteristics are discussed both for the problem of orientation estimation and for the problem of combined position and orientation estimation. Finally, the topic of sensor calibration is discussed and illustrated in terms of the estimation of an unknown gyroscope bias.

Future work In Paper A we discuss four different algorithms for pose estimation. An interesting direction for future work would be to consider also other popular algorithms such as the complementary filter, see e.g. Mahony et al. (2008); Bald-

win et al. (2007), and the filter presented by Madgwick et al. (2011). Furthermore, it would be interesting to have a closer look at using spherical distributions to represent the orientation. These distributions explicitly model the orientation to lie on a manifold. In recent years, a number of approaches have been proposed to estimate the orientation using these kinds of distributions. For instance, in Kurz et al. (2013); Gilitschenski et al. (2016); Glover and Kaelbling (2013), algorithms are presented to estimate orientation by using Bingham distributions.

4.2 Inertial sensor motion capture

Contributions Paper B presents an approach for inertial sensor motion capture to estimate the pose of the human body. We assume that the sensors are more or less rigidly attached to the body and that the body segments are connected to each other. Solving the problem as a smoothing optimization problem allows us to straightforwardly incorporate the connection between the body segments as equality constraints into the problem. The algorithm is applied to experimental data with promising results.

As discussed in Example 3.1, the number of states in the smoothing formulation of the motion capture problem is large. It grows both with time and with the number of sensors and body segments that are considered. However, as illustrated in Figure 2.4, the states x_{t+1} in the motion capture problem are conditionally independent of the states $x_{1:t-1}$ given the states x_t . This structure can be exploited using a technique called message passing as presented in Paper C. This allows for solving the smoothing problem for large data sets. A similar structure can be seen in the human body where the left foot is connected to the left lower leg (and not to any other body segments), the left lower leg is connected to the left upper leg, and so forth. Using this structure, message passing can also be used to solve the problem in a distributed manner using the sensors on the body.

Future work The topic of inertial sensor motion capture is an interesting and large field of study in itself. Because of this, there are many interesting directions of future work for Papers B and C, a few of which will be highlighted here. First of all, Paper B only presents a proof-of-concept, showing that inertial sensor motion capture can indeed be solved using an optimization approach. The algorithm is applied to data from a lower-body configuration with promising results. However, an extensive analysis of the quality of the estimates has not been performed and is an important direction for future work. We also see clear possibilities for future work in the following directions:

- **Relative pose estimation without using magnetometers** Solving the inertial motion capture problem using an optimization-based approach allows us to straightforwardly incorporate equality constraints to model the connections of the different body segments. By incorporating these constraints, the sensors' relative position and orientation become observable as long as the subject is not standing completely still (Hol, 2011). Because of this, it is

not necessary to make use of magnetometer data to provide heading information. The use of magnetometers is known to be problematic specifically in motion capture applications because the magnetic field at the different sensor locations is typically different, see e.g. Ligorio and Sabatini (2016); Seel et al. (2014); El-Gohary and McNames (2015) and the references in Paper B. The claim in Hol (2011) that the relative position and orientation become observable if the subject is not completely standing still, is mainly based on experimental results and on physical insight. It would be an interesting direction for future work to undertake a more rigorous observability study and to derive the necessary and sufficient conditions under which the relative pose is indeed observable.

- **Estimating calibration parameters** In Paper B, we assume that the position and orientation of the sensors on the body segments are known from pre-calibration. It would be an interesting direction of future work to add these calibration parameters as unknowns in the optimization approach. We expect that these parameters will only be identifiable under “sufficient” motion of the body. It would be interesting to derive conditions under which the calibration parameters can be identified from the available data and to study the quality of the parameter estimates as a function of the motion of the body.
- **Online pose estimation** The algorithm derived in Paper B obtains smoothing estimates of the pose. To allow for online estimation, it would be interesting to consider filtering or sliding window approaches. A first step in this direction has been taken in the master thesis project by Lorenz (2016). To run the filtering or sliding window approach in real-time, an efficient implementation is necessary. It would be interesting to see if the message passing algorithm presented in Paper C will be useful for this.
- **Activity recognition** In Paper B we have focused on pose estimation using inertial sensors placed on the human body. A separate but related topic is that of activity recognition using sensors placed on the human body, see e.g. Bulling et al. (2014); Hardegger et al. (2016); Reiss et al. (2010). This field focuses on recognizing the activities that the subject performs. It would be interesting to combine the motion capture approach from Paper B with activity recognition.

A number of these directions of future work have already been addressed in Taetz et al. (2016) and Miezal et al. (2016). In Miezal et al. (2016), an optimization-based solution to the inertial motion capture problem is presented that is inspired by the approach in Paper B. A sliding window approach is used to allow for online estimation. The approach is compared to two different EKF -based approaches, particularly in terms of performance in the presence of calibration errors and dependence on magnetometer usage. The optimization-based approach was shown to result in more accurate pose estimates than the EKF approaches. In Taetz et al. (2016), an optimization-based approach using a sliding window of data was used

to estimate both the human body pose and the calibration parameters, i.e. the position and orientation of the sensors on the body segments. To this end, additional biomechanical models were included in the optimization approach.

4.3 Combining UWB with inertial sensors

Contributions Paper D presents our approach to combine inertial measurements with TOA measurements from a UWB system for indoor positioning. We consider both pose estimation using inertial and UWB measurements, and calibration of the UWB setup as shown in Figure 1.8. By using the asymmetric distribution illustrated in Figure 2.2c, we explicitly model the possibility of delayed measurements due to multipath or NLOS while not allowing for measurements arriving *earlier*, i.e. traveling faster than the speed of light. It is experimentally shown that taking the possibility of delayed measurements into account leads to significantly improved position estimates.

Future work In Paper D we use a tailored asymmetric heavy-tailed distribution to model the possibility of delayed measurements. Related studies have also modeled this presence of delayed measurements using skew- t distributions (Nurminen et al., 2015; Müller et al., 2016) and Gaussian mixture models (Müller et al., 2014). It would be an interesting topic for future work to study how these approaches compare to each other. Another interesting direction of future work is to combine measurements from multiple UWB transmitters and multiple IMUs for human body motion capture. This would be a combination of the work presented in Paper B and that presented in Paper D and would allow us to estimate the absolute pose of the human body.

4.4 Magnetometer calibration

Contributions Paper E presents a magnetometer calibration algorithm which uses inertial sensors to calibrate the magnetometer for the presence of magnetic disturbances attached to the sensor. The algorithm also calibrates for magnetometer sensor errors and for misalignment between the magnetometer and the inertial sensor axes. The calibration algorithm is based on an ML formulation and is shown to give good results using data from two different commercially available IMUs. Using the calibrated magnetometer measurements in combination with the inertial sensors to determine the sensor's orientation is shown to lead to significantly improved heading estimates.

Future work In Paper E we show that our magnetometer calibration algorithm leads to significantly improved heading estimates based on measurements from two different commercially available IMUs. An interesting line of future work is to apply the magnetometer calibration algorithm to inertial and magnetometer

measurements from a smartphone. Smartphones typically use their own magnetometer calibration algorithm, thereby complicating the testing of other calibration algorithms. However, as of Android API level 18 (Jelly Bean MR2), it is possible to log uncalibrated magnetometer data. Hence, it is possible to apply our calibration algorithm to measurements from a smartphone.

The calibration algorithm is formulated as a batch, offline, method. It would be interesting to extend it to an online approach. Using this approach, it might be possible to automatically recalibrate the sensor once it enters a different magnetic environment.

4.5 Mapping and localization using magnetic fields

Contributions Paper F presents an approach to build maps of the indoor magnetic field. Physical knowledge is incorporated by modeling the magnetic field as the gradient of a scalar potential. The magnetic field map is built using GP regression. Since the magnetometers typically sample at 50 or 100 Hz, the amount of data that can be used for building the map of the magnetic field grows quickly over time. As illustrated in Figure 2.5, the GP model assumes that the magnetic field at each location depends on the magnetic field at all other locations. Because of this, GP regression becomes intractable for large amounts of data. To circumvent this issue, we use a computationally efficient implementation using the approach introduced in Solin and Särkkä (2014). Using this approach in combination with the sequential approach introduced in Särkkä et al. (2013) allows for online updating of the magnetic field estimate. It also opens up the possibility to focus on the spatio-temporal problem in which the magnetic field can change over time, for instance due to furniture being moved around.

In Paper G we use the magnetic field as a source of position information. We make use of a magnetic coil which generates a magnetic field that we can model using analytical expressions. Combining magnetometer and inertial measurements, it is possible to estimate the pose of the sensor. Based on experimental results we show that accurate pose estimates can be obtained in the vicinity of the coil, where the magnetic field is strong.

Future work In Paper F we discussed a method to build maps of the magnetic field. Paper G focused on localization in a known map. A natural direction of future work is to consider the problem of simultaneous localization and mapping (SLAM). A first step in this direction has been set by Solin et al. (2016), where a smartphone is localized in a map which is built using the approach presented in Paper F. Using such a SLAM approach, it would be very interesting to analyze what localization accuracy we can achieve. Furthermore, it would be interesting to analyze if this accuracy varies significantly over different buildings because of differences in the amount of magnetic material that is present.

4.6 Concluding remarks

In general, these are interesting times to work with inertial sensors. The quality of their measurements has improved while their cost has decreased, leading to an increase in availability. The fact that they have become so widely available opens up for many exciting possibilities. For instance, a lab for master students has been developed at Linköping University in which students implement their own orientation estimation algorithm using data from a smartphone (Hendeby et al., 2014). Other sensors have also undergone significant developments. For instance, very small devices have been developed which can both act as UWB receiver and transmitter (DecaWave, 2016; BeSpoon, 2016). Most experiments in this thesis have been done with standalone IMUs. In Paper A, however, some experiments have been included using measurements collected with a smartphone. It will be very interesting to see how the measurements from smartphones and from small UWB devices can be used and what the quality of the resulting estimates will be. It will also be interesting to see what new and exciting applications for position and orientation estimation using inertial sensors will open up in the future.

Bibliography

- M. Angermann, M. Frassl, M. Doniec, B. J. Julian, and P. Robertson. Characterization of the indoor magnetic field for applications in localization and mapping. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–9, Sydney, Australia, November 2012.
- G. Baldwin, R. Mahony, J. Trumpf, T. Hamel, and T. Cheviron. Complementary filter design on the special Euclidean group $SE(3)$. In *Proceedings of the IEEE European Control Conference (ECC)*, pages 3763–3770, Kos, Greece, July 2007.
- BeSpoon. <http://spoonphone.com/en/>, 2016. Accessed on November 30.
- H. Bijl. *Gaussian process regression techniques – With applications to wind turbines*. PhD thesis, Delft University of Technology, Delft, the Netherlands, 2016.
- C. M. Bishop. *Pattern recognition and machine learning*. Springer Science + Business Media, 2006.
- S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- A. Bulling, U. Blanke, and B. Schiele. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys (CSUR)*, 46(3):33, 2014.
- DecaWave. <http://www.decawave.com/>, 2016. Accessed on November 30.
- A. Doucet and A. M. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. In *The Oxford Handbook of Nonlinear Filtering*. Oxford University Press, 2011.
- M. El-Gohary and J. McNames. Human joint angle estimation with inertial sensors and validation with a robot arm. *IEEE Transactions on Biomedical Engineering*, 62(7):1759–1767, 2015.
- M. Frassl, M. Angermann, M. Lichtenstern, P. Robertson, B. J. Julian, and M. Doniec. Magnetic maps of indoor environments for precise localization

- of legged and non-legged locomotion. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 913–920, Tokyo, Japan, November 2013.
- I. Gilitschenski, G. Kurz, S. J. Julier, and U. D. Hanebeck. Unscented orientation estimation based on the Bingham distribution. *IEEE Transactions on Automatic Control*, 61(1):172–177, 2016.
- J. Glover and L. P. Kaelbling. Tracking 3-D rotations with the quaternion Bingham filter. *Computer Science and Artificial Intelligence Laboratory Technical Report*, 2013.
- D. J. Griffiths. *Introduction to electrodynamics*, volume 3. Prentice Hall, New Jersey, 1999.
- F. Gustafsson. *Statistical Sensor Fusion*. Studentlitteratur, 2012.
- A. Gut. *An intermediate course in probability*. Springer-Verlag, New York, 1995.
- M. Hardegger, D. Roggen, A. Calatroni, and G. Tröster. S-SMART: A unified bayesian framework for simultaneous semantic mapping, activity recognition, and tracking. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 7(3):34, 2016.
- G. Hendeby, F. Gustafsson, and N. Wahlström. Teaching sensor fusion and Kalman filtering using a smartphone. In *Proceedings of the 19th World Congress of the International Federation of Automatic Control*, pages 10586–10591, Cape Town, South Africa, August 2014.
- J. D. Hol. *Sensor fusion and calibration of inertial sensors, vision, ultra-wideband and GPS*. Dissertation no. 1368, Linköping University, Linköping, Sweden, June 2011.
- J. D. Hol, F. Dijkstra, H. Luinge, and T. B. Schön. Tightly coupled UWB/IMU pose estimation. In *Proceedings of the IEEE International Conference on Ultra-Wideband (ICUWB)*, pages 688–692, Vancouver, Canada, September 2009.
- J. D. Hol, T. B. Schön, and F. Gustafsson. Ultra-wideband calibration for indoor positioning. In *Proceedings of the IEEE International Conference on Ultra-Wideband (ICUWB)*, pages 1–4, Nanjing, China, September 2010.
- A. J. Isaksson, J. Sjöberg, D. Törnqvist, L. Ljung, and M. Kok. Using horizon estimation and nonlinear optimization for grey-box identification. *Journal of Process Control*, 30:69–79, June 2015.
- J. D. Jackson. *Classical Electrodynamics*. Wiley, New York, 3rd edition, 1999.
- R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of basic engineering*, 82(1):35–45, 1960.

- S. Khoshfetrat Pakazad, A. Hansson, M. S. Andersen, and I. Nielsen. Distributed primal-dual interior-point methods for solving tree-structured coupled convex problems using message-passing. *Optimization Methods and Software*, pages 1–35, 2016.
- M. Kok. *Probabilistic modeling for positioning applications using inertial sensors*. Licentiate’s thesis no. 1656, Linköping University, Linköping, Sweden, June 2014.
- M. Kok and T. B. Schön. Maximum likelihood calibration of a magnetometer using inertial sensors. In *Proceedings of the 19th World Congress of the International Federation of Automatic Control*, pages 92–97, Cape Town, South Africa, August 2014.
- M. Kok and T. B. Schön. Magnetometer calibration using inertial sensors. *IEEE Sensors Journal*, 16(14):5679 – 5689, 2016.
- M. Kok, J. D. Hol, T. B. Schön, F. Gustafsson, and H. Luinge. Calibration of a magnetometer in combination with inertial sensors. In *Proceedings of the 15th International Conference on Information Fusion*, pages 787–793, Singapore, July 2012.
- M. Kok, N. Wahlström, T. B. Schön, and F. Gustafsson. MEMS-based inertial navigation based on a magnetic field map. In *Proceedings of the 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 6466–6470, Vancouver, Canada, May 2013.
- M. Kok, J. D. Hol, and T. B. Schön. An optimization-based approach to human body motion capture using inertial sensors. In *Proceedings of the 19th World Congress of the International Federation of Automatic Control*, pages 79–85, Cape Town, South Africa, August 2014.
- M. Kok, J. Dahlin, T. B. Schön, and A. Wills. Newton-based maximum likelihood estimation in nonlinear state space models. In *Proceedings of the 17th IFAC Symposium on System Identification*, pages 398–403, Beijing, China, October 2015a.
- M. Kok, J. D. Hol, and T. B. Schön. Indoor positioning using ultrawideband and inertial measurements. *IEEE Transactions on Vehicular Technology*, 64(4):1293–1303, 2015b.
- M. Kok, J. D. Hol, and T. B. Schön. Using inertial sensors for position and orientation estimation. Technical Report LiTH-ISY-R-3093, Department of Electrical Engineering, Linköping University, Linköping, Sweden, December 2016a.
- M. Kok, S. Khoshfetrat Pakazad, T. B. Schön, A. Hansson, and J. D. Hol. A scalable and distributed solution to the inertial motion capture problem. In *Proceedings of the 19th International Conference on Information Fusion*, pages 1348–1355, Heidelberg, Germany, July 2016b.

- J. Kronander, J. Dahlin, D. Jönsson, M. Kok, T. B. Schön, and J. Unger. Real-time video based lighting using GPU raytracing. In *Proceedings of the 2014 European Signal Processing Conference (EUSIPCO)*, pages 1627–1631, Lisbon, Portugal, September 2014.
- G. Kurz, I. Gilitschenski, S. Julier, and U. D. Hanebeck. Recursive estimation of orientation based on the Bingham distribution. In *Proceedings of the 16th International Conference on Information Fusion*, pages 1487–1494, Istanbul, Turkey, July 2013.
- G. Ligorio and A. M. Sabatini. Dealing with magnetic disturbances in human motion capture: A survey of techniques. *Micromachines*, 7(3):43, 2016.
- L. Ljung. *System Identification, Theory for the User*. Prentice Hall PTR, 2nd edition, 1999.
- M. Lorenz. Towards realtime estimation of human motion using inertial sensors without magnetometers – an optimization-based filtering approach. Master thesis, TU Berlin, Germany, 2016.
- S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan. Estimation of IMU and MARG orientation using a gradient descent algorithm. In *IEEE International Conference on Rehabilitation Robotics*, pages 1–7, Zürich, Switzerland, Jun – Jul 2011.
- R. Mahony, T. Hamel, and J.-M. Pflimlin. Nonlinear complementary filters on the special orthogonal group. *IEEE Transactions on Automatic Control*, 53(5):1203–1218, 2008.
- M. Miezal, B. Taetz, and G. Bleser. On inertial body tracking in the presence of model calibration errors. *Sensors*, 16(7):1132, 2016.
- P. Müller, H. Wymeersch, and R. Piché. UWB positioning with generalized gaussian mixture filters. *IEEE Transactions on Mobile Computing*, 13(10):2406–2414, 2014.
- P. Müller, J. A. del Peral-Rosado, R. Piché, and G. Seco-Granados. Statistical trilateration with skew-t distributed errors in LTE networks. *IEEE Transactions on Wireless Communications*, 15(10):7114–7127, 2016.
- National Centers for Environmental Information. <https://www.ngdc.noaa.gov/geomag/geomag.shtml>, 2016. Accessed on October 10.
- J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research, 2nd edition, 2006.
- H. Nurminen, T. Ardeschiri, R. Piché, and F. Gustafsson. Robust inference for state-space models with skewed measurement noise. *IEEE Signal Processing Letters*, 22(11):1898–1902, 2015.

- F. Olsson, M. Kok, K. Halvorsen, and T. B. Schön. Accelerometer calibration using sensor fusion with a gyroscope. In *Proceedings of the IEEE Workshop on Statistical Signal Processing*, pages 660–664, Palma de Mallorca, Spain, June 2016.
- J. Quiñonero-Candela and C. E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6: 1939–1959, 2005.
- C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.
- A. Reiss, G. Hendeby, G. Bleser, and D. Stricker. Activity recognition using biomechanical model based pose estimation. In *Proceedings of the 5th European Conference on Smart Sensing and Context*, pages 42–55, Passau, Germany, November 2010.
- S. Särkkä. *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.
- S. Särkkä, A. Solin, and J. Hartikainen. Spatiotemporal learning via infinite-dimensional Bayesian filtering and smoothing. *IEEE Signal Processing Magazine*, 30(4):51–61, 2013.
- T. B. Schön and F. Lindsten. Manipulating the multivariate Gaussian density. *Technical Report, Division of Automatic Control, Linköping University, Sweden*, Jan 2011.
- T. Seel, J. Raisch, and T. Schauer. IMU-based joint angle measurement for gait analysis. *Sensors*, 14(4):6891–6909, 2014.
- A. Solin and S. Särkkä. Hilbert space methods for reduced-rank Gaussian process regression. *ArXiv e-prints*, January 2014. arXiv:1401.5508.
- A. Solin, M. Kok, N. Wahlström, T. B. Schön, and S. Särkkä. Modeling and interpolation of the ambient magnetic field by Gaussian processes. *ArXiv e-prints*, September 2015. arXiv:1509.04634.
- A. Solin, S. Särkkä, J. Kannala, and E. Rahtu. Terrain navigation in the magnetic landscape: Particle filtering for indoor positioning. In *Proceedings of the European Navigation Conference*, Helsinki, Finland, May–June 2016.
- A. Svensson, T. B. Schön, and M. Kok. Nonlinear state space smoothing using the conditional particle filter. In *Proceedings of the 17th IFAC Symposium on System Identification*, pages 975–980, Beijing, China, October 2015.
- B. Taetz, G. Bleser, and M. Miezal. Towards self-calibrating inertial body motion capture. In *19th International Conference on Information Fusion*, pages 1751–1759, Heidelberg, Germany, July 2016.
- Trivisio Prototyping GmbH. <http://www.trivisio.com>, 2016. Accessed on November 14.

N. Wahlström, M. Kok, T. B. Schön, and F. Gustafsson. Modeling magnetic fields using Gaussian processes. In *Proceedings of the 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 3522 – 3526, Vancouver, Canada, May 2013.

Xsens Technologies B.V. <http://www.xsens.com>, 2016. Accessed on October 4.

Part II

Publications

Paper A

Using inertial sensors for position and orientation estimation

Authors: Manon Kok, Jeroen D. Hol and Thomas B. Schön

Edited version of the paper:

M. Kok, J. D. Hol, and T. B. Schön. Using inertial sensors for position and orientation estimation. Technical Report LiTH-ISY-R-3093, Department of Electrical Engineering, Linköping University, Linköping, Sweden, December 2016a.

Using inertial sensors for position and orientation estimation

Manon Kok^{*}, Jeroen D. Hol[†] and Thomas B. Schön[‡]

^{*}Dept. of Electrical Engineering,
Linköping University,
SE-581 83 Linköping, Sweden
manko@isy.liu.se

[†]Xsens Technologies B. V.
P.O. Box 559,
7500 AN Enschede, the Netherlands
jeroen.hol@xsens.com

[‡]Dept. of Information Technology
Uppsala University,
SE-751 05 Uppsala, Sweden
thomas.schon@it.uu.se

Abstract

In recent years, micro-machined electromechanical system (MEMS) inertial sensors (3D accelerometers and 3D gyroscopes) have become widely available due to their small size and low cost. Inertial sensor measurements are obtained at high sampling rates and can be integrated to obtain position and orientation information. These estimates are accurate on a short time scale, but suffer from integration drift over longer time scales. To overcome this issue, inertial sensors are typically combined with additional sensors and models. In this tutorial we focus on the signal processing aspects of position and orientation estimation using inertial sensors, discussing different modeling choices and a selected number of important algorithms. The algorithms include optimization-based smoothing and filtering as well as computationally cheaper extended Kalman filter implementations.

1 Introduction

In this section, we introduce the problem of position and orientation estimation using inertial sensors. The combined estimation of both position and orientation is sometimes called *pose estimation*. We will start by providing a brief background and motivation in Section 1.1 by explaining what inertial sensors are, how they can be used for pose estimation and give a few concrete examples of relevant application areas. This will give the basis for the problem of pose estimation using inertial sensors discussed in Section 1.2. Finally, in Section 1.3 we will give an overview of the contents and an outline of the rest of this tutorial.

1.1 Background and motivation

The term *inertial sensor* is used to denote the combination of a three-axis accelerometer and a three-axis gyroscope. Devices containing these sensors are commonly referred to as inertial measurement units (IMUs), but inertial sensors are nowadays also present in any modern smartphone, and in devices such as Wii controllers and virtual reality (VR) headsets, as shown in Figure 1.

A gyroscope measures the sensor's *angular velocity*, i.e. the rate of change of the sensor's orientation. An accelerometer measures the *external specific force* acting on the sensor. The specific force consists of both the sensor's acceleration and the earth's gravity. Nowadays, many gyroscopes and accelerometers are based on micro-machined electromechanical system (MEMS) technology. MEMS components are small, light, inexpensive, have low power consumption and short start-up times. Their accuracy has significantly increased over the years.

There is a large and ever-growing number of application areas for inertial sensors, see e.g. Barbour and Schmidt (2001); Hol (2011); Perlmutter and Robin (2012); Xsens Technologies B.V. (2016). Generally speaking, inertial sensors can be used to provide information about the pose of any object that they are rigidly attached to. It is also possible to combine multiple inertial sensors to obtain information about the pose of separate connected objects. A major application area is to use inertial sensors to track human motion. Several illustrations of this are shown in Figure 2. As can be seen, the application areas are as diverse as robotics, biomechanical analysis and motion capture for the movie and gaming industries. In fact, the use of inertial sensors for pose estimation is now common practice in for instance robotics and human motion tracking, see e.g. Luinge and Veltink (2005); Harle (2013); Raibert et al. (2008). For example, a recent survey (Adler et al., 2015) shows that 28% of the contributions to the IEEE International Conference on Indoor Positioning and Indoor Navigation (IPIN) make use of inertial sensors. Inertial sensors are also frequently used for pose estimation of cars, boats, trains and aerial vehicles, see e.g. Skog and Händel (2009); Chao et al. (2010). Examples of this are shown in Figure 3.

There exists a large amount of literature on the use of inertial sensors for pose estimation. The reason for this is not only the large number of application areas. Important reasons are also that the pose estimation problem is nonlinear and that different parametrizations of the orientation need to be considered (Grisetti et al., 2010a; Kurz et al., 2013), each with its own specific properties. Interestingly, approximative and relatively simple pose estimation algorithms work quite well in practice. However, careful modeling and a careful choice of algorithms do improve the accuracy of the estimates.

In this tutorial we focus on the signal processing aspects of pose estimation using inertial sensors, discussing different modeling choices and a number of important algorithms. These algorithms will provide the reader with a starting point to implement their own pose estimation algorithm. The algorithms will include a relatively simple and computationally cheap implementation of an extended Kalman filter and a more complex algorithm for obtaining smoothed pose estimates by post-processing the data.



(a) Left bottom: an Xsens MTx IMU (Xsens Technologies B.V., 2016). Left top: a Trivisio Colibri Wireless IMU (Trivisio Prototyping GmbH, 2016). Right: a Samsung Galaxy S4 mini smartphone.



(b) A Samsung gear VR.¹



(c) A Wii controller containing an accelerometer and a Motion-Plus expansion device containing a gyroscope.²

Figure 1: Examples of devices containing inertial sensors.

¹ ‘Samsung Gear VR’ available at flic.kr/photos/pestoverde/15247458515 under CC BY 2.0 (<http://creativecommons.org/licenses/by/2.0>).

² ‘WiiMote with MotionPlus’ by Asmodai available at https://commons.wikimedia.org/wiki/File:WiiMote_with_MotionPlus.JPG under CC BY SA (<https://creativecommons.org/licenses/by-sa/3.0/>).



(a) Back pain therapy using serious gaming. IMUs are placed on the chest-bone and on the pelvis to estimate the movement of the upper body and pelvis. This movement is used to control a robot in the game and promotes movements to reduce back pain.



(b) Actor Seth MacFarlane wearing 17 IMUs to capture his motion and animate the bear Ted. The IMUs are placed on different body segments and provide information about the relative position and orientation of each of these segments.

Figure 2: Examples illustrating the use of multiple IMUs placed on the human body to estimate its pose. Courtesy of Xsens Technologies.



(a) Inertial sensors are used in combination with the global positioning system (GPS) to estimate the position of the cars in a challenge on cooperative and autonomous driving.



(b) Due to their small size and low weight, IMUs can be used to estimate the orientation for control of an unmanned helicopter.

Figure 3: Examples illustrating the use of a single IMU placed on a moving object to estimate its pose. Courtesy of Xsens Technologies.

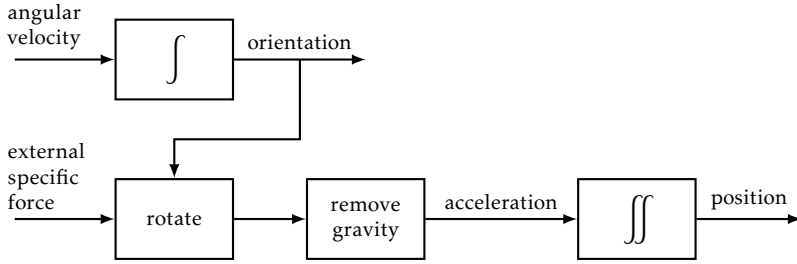


Figure 4: Schematic illustration of dead-reckoning, where the accelerometer measurements (external specific force) and the gyroscope measurements (angular velocity) are integrated to position and orientation.

1.2 Using inertial sensors for pose estimation

IMUs are frequently used for navigation purposes where the position and the orientation of a device is of interest. Integration of the gyroscope measurements provides information about the orientation of the sensor. After subtraction of the earth’s gravity, double integration of the accelerometer measurements provides information about the sensor position. To be able to subtract the earth’s gravity, the orientation of the sensor needs to be known. Hence, estimation of the sensor’s position and orientation are inherently linked when it comes to inertial sensors. The process of integrating the inertial sensors to obtain pose estimates, often called *dead-reckoning*, is summarized in Figure 4.

If the initial pose would be known, and if perfect models for the inertial sensor measurements would exist, the process illustrated in Figure 4 would lead to perfect pose estimates. In practice, however, the inertial measurements are noisy and biased as will be discussed in more detail in Section 2.4. Because of this, the integration steps from angular velocity to rotation and acceleration to position introduce *integration drift*. This is illustrated in Example 1.

Example 1: Integration drift

Let us assume that we have a sensor which measures a non-zero, constant bias. The integrated measurements will grow linearly with time, while the double integration will grow quadratically with time. If the sensor instead measured a zero-mean white noise signal, the expected value of the integrated measurements would be zero, but the variance would grow with time. This is illustrated in Figure 5 for the integration of a signal $y_t = e_t$ with $e_t \sim \mathcal{N}(0, 1)$. Hence, integration drift is both due to integration of a constant bias and due to integration of noise.

To illustrate integration drift using experimental data, a stationary data set is collected with a Sony Xperia Z5 Compact smartphone using the app described in Hendeby et al. (2014). The smartphone contains accelerometers and gyroscopes produced by Invensense (InvenSense, 2016). We integrate the inertial measurements to obtain position and orientation estimates. Since the smartphone is kept stationary during the data collection, we expect the position and orientation to remain the same. However, the orientation estimates drift a few degrees over 10 s

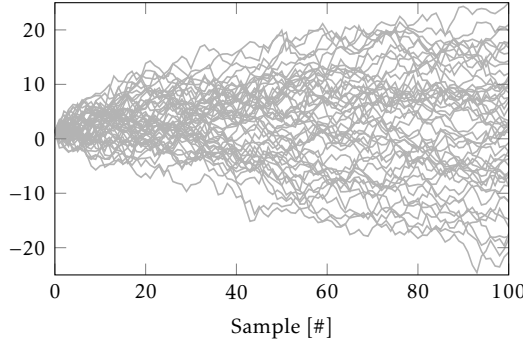
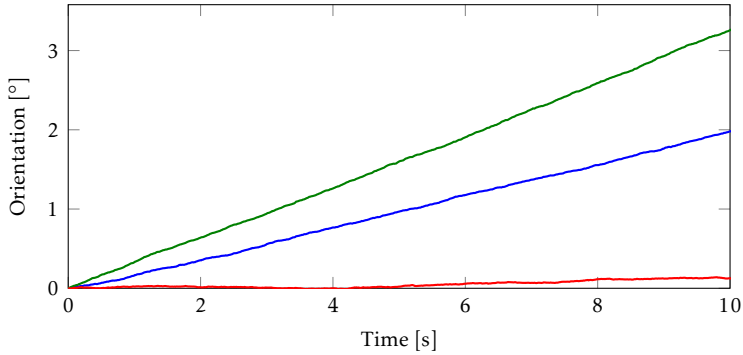


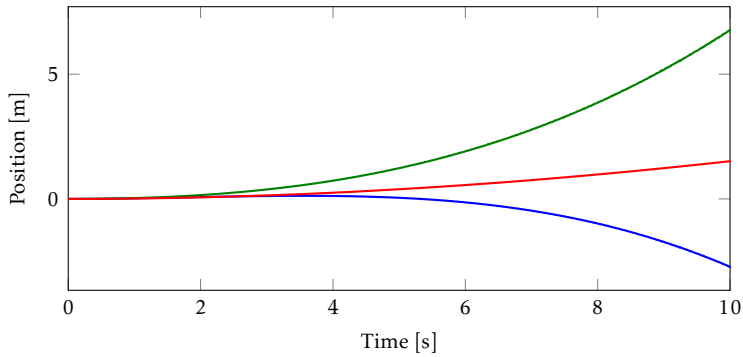
Figure 5: Integration of a white noise signal $y_t \sim \mathcal{N}(0, 1)$ for 50 noise realizations.

as shown in Figure 6a. Note that the integration drift is not the same for all axes. This is mainly due to a different sensor bias in the different axes. This will be studied further in Example 4, where the same data set is used to study the sensor characteristics. As shown in Figure 6b, the position drifts several meters over 10 s. The reason for this is two-fold. First, the accelerometer measurements need to be integrated twice. Second, the orientation estimates need to be used to subtract the gravity and any errors in this will lead to *leakage of gravity* into the other components.

From the example above, it can be concluded that errors in the measurements have a large impact on the quality of the estimated position and orientation using inertial sensors only. This is specifically the case for position, which relies both on double integration of the acceleration and on accurate orientation estimates to subtract the earth's gravity. Because of this, inertial sensors need to be supplemented with other sensors or other models to lead to accurate position and orientation estimates. Inertial sensors provide pose estimates at high sampling rates which are accurate on a short time scale but drift over longer time scales. Because of this, they are very suitable for being used in combination with sensors with a lower sampling rate but with information that does not drift over time. For pose estimation, they are often combined with sensors such as GPS (Kaplan and Hegarty, 1996; Titterton and Weston, 1997; Hol, 2011), ultrawideband (UWB) (Kok et al., 2015; Sczyslo et al., 2008; Pittet et al., 2008; Corrales et al., 2008; De Angelis et al., 2010) and vision (Corke et al., 2007; Hol et al., 2007; Li and Mourikis, 2013; Martinelli, 2012). For orientation estimation, they are often used in combination with magnetometers (Sabatini, 2006; Roetenberg et al., 2005). The reason for this is that the accelerometer can provide information about the vertical direction by measuring the direction of the gravity. The angle of deviation from the vertical is called the *inclination*. However, it can not provide information about the orientation around the vertical, which is also called *heading* or *yaw*. The heading can be



(a) Integrated orientation for the position in x - (blue), y - (green) and z -direction (red).



(b) Integrated position for rotation around the x -axis (blue), the y -axis (green) and the z -axis (red).

Figure 6: Position and orientation estimates based on dead-reckoning of the inertial sensors only. The data is collected with a Sony Xperia Z5 Compact smartphone that is lying stationary on a table.

measured using a magnetometer, which measures the direction of the magnetic field, pointing to the local magnetic north.

This tutorial aims at giving an introduction on how to use inertial sensors for position and orientation estimation, but also on how to combine them with additional information. These additional sensors are not the focus of this paper but simple models will be used for magnetometers and sensors providing position information to illustrate the combined use of these sensors.

1.3 Tutorial content and its outline

To obtain accurate position and orientation estimates using inertial sensors in combination with additional measurements or models, a number of important things need to be considered. First, the quantities measured by the inertial sensors

need to be accurately described and the sources of error need to be characterized. This is the topic of Section 2. Note that throughout the tutorial, we will focus on MEMS inertial sensors and consider both data from standalone IMUs and from smartphones. This implies that we do not focus on for instance mechanical or optical gyroscopes and on mechanical or solid-state accelerometers (Titterton and Weston, 1997). These sensors can have characteristics that are quite different from the MEMS inertial sensors considered here.

Based on the analysis of the sensors in Section 2 and on additional analysis of the application at hand, models can be constructed. This is the topic of Section 3, where we will also discuss different parametrizations of orientation. This will highlight the challenges in parametrizing and estimating orientations and show that the orientation estimation problem is inherently nonlinear. Furthermore, we will present two models that can be used for position and orientation estimation. The first is a model for pose estimation using inertial measurements in combination with position measurements. The second is a model for orientation estimation, using inertial and magnetometer measurements.

In Section 4, different algorithms for position and orientation estimation will be introduced. The general structure of the algorithms will be discussed, after which explicit algorithms for orientation estimation using inertial and magnetometer measurements are given. We will also discuss how the algorithms can be extended to pose estimation when position measurements are available. Some general characteristics of the two estimation problems will be given and the quality of the estimates from the different algorithms will be analyzed. Which algorithm is most suitable for which application depends strongly on the computational power that is available, the accuracy that is required and the characteristics of the problem at hand.

In Section 4, we assume that the sensors are properly calibrated. However, calibration of the sensors is important for instance to estimate the inertial sensor biases. Furthermore, calibration is specifically of concern when combining inertial data with other sensors. In these cases, it is important that the inertial sensor axes and the axes of the additional sensors are aligned. Sensor calibration is the topic of Section 5. As an illustrative example, we will consider the estimation of an unknown gyroscope bias. We will end this tutorial with some concluding remarks in Section 6.

2 Inertial sensors

To combine inertial measurements with additional sensors and models for position and orientation estimation, it is important to accurately describe the quantities measured by the inertial sensors as well as to characterize the typical sensor errors. This will be the topic of this section. It will serve as a basis for the probabilistic models discussed in Section 3.

As discussed in Section 1, accelerometers and gyroscopes measure specific force and angular velocity, respectively. In Sections 2.2 and 2.3, we will discuss the quantities that are measured by the gyroscope and accelerometer in more

detail. To enable a discussion about this, in Section 2.1 a number of coordinate frames and the transformations between them will be discussed. We assume that we have 3D accelerometers and 3D gyroscopes, i.e. that the sensors have three sensitive axes along which these physical quantities are measured. They are measured in terms of an output voltage. Based on a factory pre-calibration, these signals are converted to a physical measurement inside the sensor. Even though the sensors are typically calibrated in the factory, (possibly time-varying) errors can still remain. In Section 2.4, the most commonly present sensor errors are discussed.

2.1 Coordinate frames

In order to discuss the quantities measured by the accelerometer and gyroscope in more detail, a number of coordinate frames need to be introduced:

The body frame b 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 is a local geographic frame in which we want to navigate. In other words, we are interested in the position and orientation of the b -frame with respect to this frame. For most applications it is defined stationary with respect to the earth. However, in cases when the sensor is expected to move 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 tutorial, unless mentioned explicitly.

The inertial frame i is a stationary frame. The IMU measures linear acceleration and angular velocity with respect to this frame. Its origin is located at the center of the earth and its axes are aligned with respect to the stars.

The earth frame e 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 with respect to the earth.

These coordinate frames are illustrated in Figure 7. We use a superscript to indicate in which coordinate frame a vector is expressed. Vectors can be rotated from one coordinate frame to another using a rotation matrix. We use a double superscript to indicate from which coordinate frame to which coordinate frame the rotation is defined. An illustration is given in Example 2.

Example 2: Rotation of vectors to different coordinate frames

Consider a vector x expressed in the body frame b . We denote this vector x^b . The rotation matrix R^{nb} rotates a vector from the body frame b to the navigation frame n . Conversely, the rotation from navigation frame n to body frame b is denoted $R^{bn} = (R^{nb})^T$. Hence, the vector x expressed in the body frame (x^b) and expressed in the navigation frame (x^n) are related according to

$$x^n = R^{nb} x^b, \quad x^b = (R^{nb})^T x^n = R^{bn} x^n. \quad (1)$$

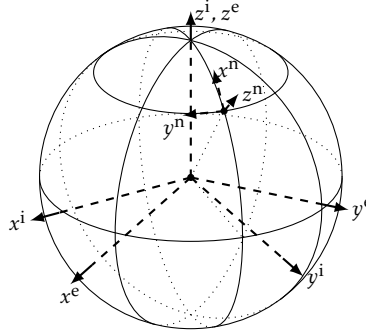


Figure 7: An illustration of the coordinate frames: the n -frame at a certain location on the earth, the e -frame rotating with the earth and the i -frame.

2.2 Angular velocity

The gyroscope measures the angular velocity of the body frame with respect to the inertial frame, expressed in the body frame (Titterton and Weston, 1997), denoted by ω_{ib}^b . This angular velocity can be expressed as

$$\omega_{ib}^b = R^{bn} (\omega_{ie}^n + \omega_{en}^n) + \omega_{nb}^b, \quad (2)$$

where R^{bn} is the rotation matrix from the navigation frame to the body frame. The *earth rate*, i.e. the angular velocity of the earth frame with respect to the inertial frame is denoted by ω_{ie} . The earth rotates around its z -axis in 23.9345 hours with respect to the stars (National Aeronautics and Space Administration, 2016). Hence, the earth rate is approximately $7.29 \cdot 10^{-5}$ rad/s.

In case the navigation frame is not defined stationary with respect to the earth, the angular velocity ω_{en} , i.e. the *transport rate* is non-zero. The angular velocity required for navigation purposes – in which we are interested when determining the orientation of the body frame with respect to the navigation frame – is denoted by ω_{nb} .

2.3 Specific force

The accelerometer measures the specific force f as expressed in the body frame (Titterton and Weston, 1997). This can be expressed as

$$f^b = R^{bn} (a_{ii}^n - g^n), \quad (3)$$

where g denotes the gravity vector and a_{ii}^n denotes the linear acceleration of the sensor expressed in the navigation frame, which is

$$a_{ii}^n = R^{ne} R^{ei} a_{ii}^i. \quad (4)$$

The subscripts are used to indicate in which frame the differentiation is performed. For navigation purposes, we are interested in the position of the sensor in the

navigation frame p^n and its derivatives as performed in the navigation frame

$$\left. \frac{d}{dt} p^n \right|_n = v_n^n, \quad \left. \frac{d}{dt} v_n^n \right|_n = a_{nn}^n. \quad (5)$$

A relation between a_{ii} and a_{nn} can be derived by using the relation between two rotating coordinate frames. Given a vector x in a coordinate frame u ,

$$\left. \frac{d}{dt} x^u \right|_u = \left. \frac{d}{dt} R^{uv} x^v \right|_u = R^{uv} \left. \frac{d}{dt} x^v \right|_v + \omega_{uv}^u \times x^u, \quad (6)$$

where ω_{uv}^u is the angular velocity of the v -frame with respect to the u -frame, expressed in the u -frame. For a derivation of this relation in the context of inertial navigation, see e.g. Hol (2011); Titterton and Weston (1997) or any textbook on dynamics, see e.g. Marion and Thornton (1995); Meriam and Kraige (1998) for a general introduction. Using the fact that

$$p^i = R^{ie} p^e, \quad (7)$$

the velocity v_i and acceleration a_{ii} can be expressed as

$$\begin{aligned} v_i^i &= \left. \frac{d}{dt} p^i \right|_i = \left. \frac{d}{dt} R^{ie} p^e \right|_i = R^{ie} \left. \frac{d}{dt} p^e \right|_e + \omega_{ie}^i \times p^i \\ &= v_e^i + \omega_{ie}^i \times p^i, \end{aligned} \quad (8a)$$

$$\begin{aligned} a_{ii}^i &= \left. \frac{d}{dt} v_i^i \right|_i = \left. \frac{d}{dt} v_e^i \right|_i + \left. \frac{d}{dt} \omega_{ie}^i \right|_i \times p^i + \omega_{ie}^i \times \left. \frac{d}{dt} p^i \right|_i \\ &= a_{ee}^i + 2\omega_{ie}^i \times v_e^i + \dot{\omega}_{ie}^i \times p^i + \omega_{ie}^i \times v_i^i, \end{aligned} \quad (8b)$$

where we have made use of (5), (6) and the fact that the angular velocity of the earth is constant, i.e. $\dot{\omega}_{ie}^i = 0$. Using the relation between the earth and navigation frames,

$$p^e = R^{en} p^n + n_{en}^e, \quad (9)$$

where n_{en} is the distance from the origin of the earth coordinate frame the origin of the navigation coordinate frame, expressions similar to (8) can be derived. Note that in general it can not be assumed that $\dot{\omega}_{en} = 0$. Inserting the obtained expressions into (8), it is possible to derive the relation between a_{ii} and a_{nn} . Instead of deriving these relations, we will assume that the navigation frame is a frame which is fixed to the earth frame and hence R^{en} and n_{en}^e are constant and

$$v_e^e = \left. \frac{d}{dt} p^e \right|_e = \left. \frac{d}{dt} R^{en} p^n \right|_e = R^{en} \left. \frac{d}{dt} p^n \right|_n = v_n^e, \quad (10a)$$

$$a_{ee}^e = \left. \frac{d}{dt} v_e^e \right|_e = \left. \frac{d}{dt} v_n^e \right|_e = a_{nn}^e. \quad (10b)$$

This is a reasonable assumption as long as the sensor does not travel over significant distances of the earth and it will be one of the modeling assumptions that we will use in this tutorial. More on the modeling choices will be discussed in Section 3.

Inserting (10) into (8) and rotating the result, it is possible to express a_{ii}^n in terms of a_{nn}^n as

$$a_{ii}^n = a_{nn}^n + 2\omega_{ie}^n \times v_n^n + \omega_{ie}^n \times \omega_{ie}^n \times p^n, \quad (11)$$

where a_{nn} is the acceleration required for navigation purposes. The term $\omega_{ie}^n \times \omega_{ie}^n \times p^n$ is known as the *centrifugal acceleration* and $2\omega_{ie}^n \times v^n$ is known as the *Coriolis acceleration*. The centrifugal acceleration is typically absorbed in the (local) gravity vector. In Example 3, we illustrate the magnitude of both the centrifugal and Coriolis acceleration.

Example 3: Magnitude of centrifugal and Coriolis acceleration

The centrifugal acceleration depends on the location on the earth. It is possible to get a feeling for its magnitude by considering the property of the cross product stating that

$$\|\omega_{ie}^n \times \omega_{ie}^n \times p^n\|_2 \leq \|\omega_{ie}^n\|_2 \|\omega_{ie}^n\|_2 \|p^n\|_2. \quad (12)$$

Since the magnitude of ω_{ie} is approximately $7.29 \cdot 10^{-5}$ rad/s and the average radius of the earth is 6371 km (National Aeronautics and Space Administration, 2016), the magnitude of the centrifugal acceleration is less than or equal to $3.39 \cdot 10^{-2}$ m/s².

The Coriolis acceleration depends on the speed of the sensor. Let us consider a person walking at a speed of 5 km/h. In that case the magnitude of the Coriolis acceleration is approximately $2.03 \cdot 10^{-4}$ m/s². For a car traveling at 120 km/h, the magnitude of the Coriolis acceleration is instead $4.86 \cdot 10^{-3}$ m/s².

2.4 Sensor errors

As discussed in Sections 2.2 and 2.3, the gyroscope measures the angular velocity ω_{ib}^b and the accelerometer measures the specific force f^b . However, as already briefly mentioned in Section 1.2, there are several reasons for why this is not exactly the case. Two of these reasons are a slowly time-varying sensor bias and the presence of measurement noise. The sensor errors in the inertial measurements are illustrated in Example 4 using experimental data.

Example 4: Inertial sensor measurements and their errors

In Figures 8 – 10, gyroscope and accelerometer measurements are displayed for around 55 min of stationary data collected with a Sony Xperia Z5 Compact smartphone. Part of this data set was used in Example 1 to illustrate integration drift. Since the smartphone is stationary, the gyroscope is expected to only measure the earth's angular velocity. However, as can be seen in Figure 8a, the gyroscope measurements are corrupted by noise. As shown in Figure 9, this noise can be seen to be quite Gaussian. Furthermore, the measurements can be seen to be biased.

During the stationary period, we would expect the accelerometer to measure the gravity, the centrifugal acceleration and the Coriolis acceleration. Note that again the measurements are corrupted by noise, which can be seen to be quite Gaussian in Figure 10. The x - and y -components of the accelerometer measurements are not zero-mean. This can be due to the fact that the table on which the smartphone lies is not completely flat, implying that part of the gravity vector is measured in these components. It can also reflect a sensor bias. The z -component

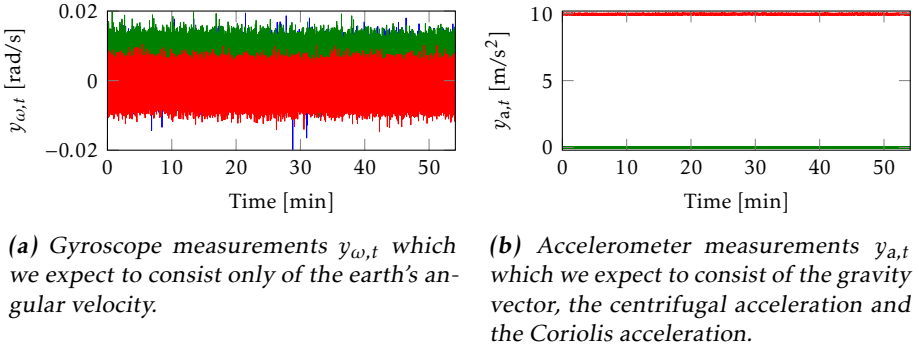


Figure 8: Inertial measurements for 55 min of stationary data. As can be seen, the measurements are corrupted by noise and have a bias.

is actually larger than expected which indicates the presence of an accelerometer bias at least in this axis.

Note that from the above discussion it can be concluded that it is more straightforward to determine the gyroscope bias than it is to determine the accelerometer bias. To be able to estimate the gyroscope bias, it is sufficient to leave the sensor stationary. However, for the accelerometer, the sensor needs to be rotated and its measurements need to be compared to the gravity vector. The gyroscope in the smartphone is actually automatically recalibrated during stationary time periods. What we displayed here is the data that has not been corrected for this (so-called uncalibrated data).

The reason why the gyroscope is calibrated during stationary periods, is because its bias is slowly time-varying. For instance, estimating the gyroscope bias based on the first minute of the data set (approximately 6 000 samples), the bias was $(35.67 \ 56.22 \ 0.30)^T \cdot 10^{-4}$ rad/s. The gyroscope bias estimated using the last minute of data was instead $(37.01 \ 53.17 \ -1.57)^T \cdot 10^{-4}$ rad/s.

The performance of IMUs is often specified in terms of their so-called *Allan variance* (IEEE, 2009; El-Sheimy et al., 2008; Allan, 1966). The Allan variance gives information about the sensor errors for stationary conditions, i.e. in a stable climate without exciting the system. It studies the effect of averaging measurements for different *cluster times* T_c . Typically, the Allan *standard deviation* $\sigma_A(T_c)$ is plotted against the cluster time T_c as illustrated in Figure 11. This figure shows the characteristic behavior for the Allan variance for inertial sensors. To study it more in detail, we will discuss two components of the Allan variance that are typically of concern for inertial sensors: the white noise and the so-called bias instability.

Assume, like in Example 1, that we have a white noise signal with standard deviation σ . A longer averaging time would for this signal lead to values closer

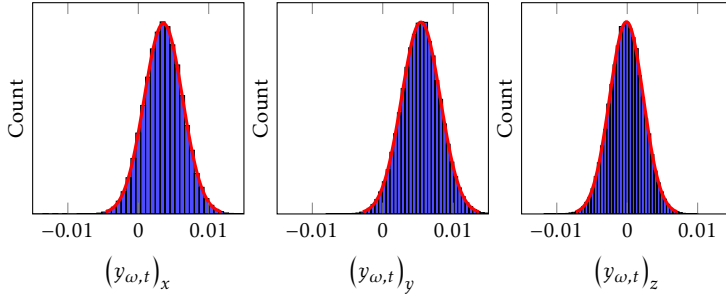


Figure 9: Histogram (blue) of the gyroscope measurements for 55 min of data from a stationary sensor and a Gaussian fit (red) to the data. As can be seen, the measurement noise looks quite Gaussian.

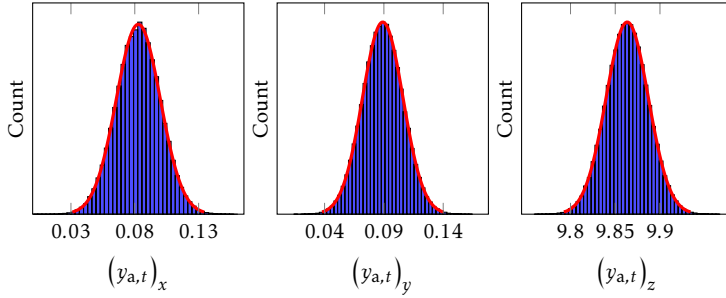


Figure 10: Histogram (blue) of the accelerometer measurements for 55 min of data from a stationary sensor and a Gaussian fit (red) to the data. As can be seen, the measurement noise looks quite Gaussian. Note the different scales on the horizontal axis.

to zero. The contribution to the Allan standard deviation from the white noise component is given by $\sigma_A(T_c) = \frac{\sigma}{\sqrt{n}}$ where n is the number of samples averaged over. This corresponds to a line with slope $-1/2$ in a log-log plot. For instance in the gyroscope Allan deviation in Figure 11, the lines can be seen to have a slope of $-1/2$ until around 10–20 s, which indicates that the white noise is the dominating source of error for these short integration times.

A constant bias does not have any effect on the Allan variance diagram. However, in case the bias changes, longer averaging times will no longer be beneficial. Hence, the Allan variance diagrams in Figure 11 show a deviation from the slope $-1/2$ for longer averaging times.

The Allan variance is a useful tool to study and compare the noise characteristics of inertial sensors. However, it only considers stationary conditions. In dynamic conditions, a large number of other error sources potentially come into play, see e.g. Titterton and Weston (1997); Woodman (2007). These are for instance related to the fact that the sensors sample at discrete times. Hence, to capture

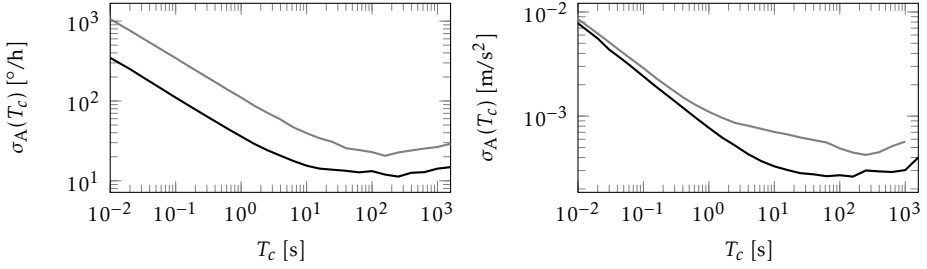


Figure 11: Left: Allan deviation for two gyroscopes. Right: Allan deviation for two accelerometers. Reproduced with permission from Vydhyathan et al. (2015).

high-frequency signals, high sampling frequencies are desired (Savage, 1998a,b). Furthermore, large dynamics can lead to erroneous or saturated measurements. Other errors that are not included are for instance changes in the sensitivity of the axes due to changes in temperature. Therefore, we should never rely just on the Allan variance when deciding which sensor to use in a particular application.

3 Probabilistic models

Pose estimation is about estimating the position and orientation of the body frame b in the navigation frame n . This problem is illustrated in Figure 12, where the position and orientation of the body changes from time t_1 to time t_2 . In this section, we will introduce the concept of probabilistic models and discuss different modeling choices for using inertial sensors for pose estimation.

The subject of probabilistic modeling is introduced in Section 3.1. Most complexity in pose estimation lies in the nonlinear nature of the orientation and the fact that orientation can be parametrized in different ways. How to parametrize the orientation is a crucial modeling choice in any pose estimation algorithm. Because of this, we will discuss different parametrizations for the orientation in Section 3.2 and in Section 3.3 we will discuss how these different parametrizations can be used in probabilistic modeling.

Our probabilistic models consist of three main components. First, in Section 3.4, we introduce models describing the knowledge about the pose that can be inferred from the measurements. Second, in Section 3.5, we model how the sensor pose changes over time. Finally, in Section 3.6, models of the initial pose are introduced.

The section will conclude with a discussion on the resulting probabilistic models in Section 3.7. Here, the models that will be used in the position and orientation estimation algorithms in Section 4 will also be introduced.

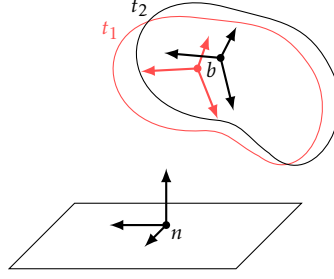


Figure 12: An illustration of the pose estimation problem. We want to express the position and orientation of the moving body frame b at times t_1 and t_2 with respect to the navigation frame n .

3.1 Introduction

Probabilistic models constitute the foundation of the estimation algorithms in Section 4. In this section we will introduce the concept of probabilistic modeling and the notation that is used in building our models. Models are used to describe the information about the *dynamics* and the available *measurements*. These models are subsequently used in combination with the measurements to *infer* some knowledge. The knowledge that we are interested in is the pose of the sensor and we use information about the *sensor dynamics* and the available measurements (amongst others, inertial measurements). A simplified case where probabilistic modeling is used to estimate the position of a sensor is given in Example 5.

Example 5: Probabilistic modeling

Let us estimate the 2D position p_t of a sensor at time t from two position measurements

$$y_t^1 = \begin{pmatrix} 0 & 0 \end{pmatrix}^\top, \quad y_t^2 = \begin{pmatrix} 2 & 0 \end{pmatrix}^\top.$$

A straightforward suggestion for an estimate of the position would be $\hat{p}_t = \begin{pmatrix} 1 & 0 \end{pmatrix}^\top$. Let us now assume that we know the accuracy of the sensors and represent this in terms of the following probabilistic models

$$\begin{aligned} y_t^1 &= p_t + e_t^1, & e_t^1 &\sim \mathcal{N}(0, 0.25 \mathcal{I}_2), \\ y_t^2 &= p_t + e_t^2, & e_t^2 &\sim \mathcal{N}(0, \mathcal{I}_2), \end{aligned}$$

where $\mathcal{I}_2$ denotes a 2×2 identity matrix. Based on this, it is sensible to trust the measurement from the first sensor more than the measurement from the second sensor. A reasonable position estimate would instead be

$$p_t \sim \mathcal{N}\left(\begin{pmatrix} 0.4 \\ 0 \end{pmatrix}, 0.2 \mathcal{I}_2\right).$$

Now consider the case where we are also interested in estimating the position p_{t+1} . Knowledge that the sensor is worn by a human or placed in a car, would give us information about how far the sensor can travel from time t to time $t + 1$. If the sensor would be placed in for instance a train, the motion would even be constrained to be along the tracks. Incorporating this information about the dynamics of the sensor will improve the estimate of p_{t+1} .

We split the knowledge that we want to infer into the unknown *time-varying states* x_t for $t = 1, \dots, N$, or equivalently $x_{1:N}$, and the unknown *constant parameters* θ . We denote the measurements by y_k for $k = 1, \dots, K$. The times k at which these measurements are obtained do not necessarily correspond with the times t at which the states are defined. It is also not necessary for all sensors to sample at the same frequency. As discussed in Section 2.4, the inertial sensors are typically sampled at fairly high rates to capture high-frequency dynamics. In stand-alone, wired IMUs, all sensors typically have the same, constant sampling frequency. Specifically in the case of wireless sensors and smartphones, however, the sampling frequencies can vary both over sensors and over time. In the remainder, we assume that the times t at which the states are defined coincide with the times k at which the gyroscopes sample. Hence, we denote the gyroscope measurements $y_{\omega,t}$ with $t = 1, \dots, N$. For notational convenience, we will also use the subscript t for the measurements from other sensors. Note that these are not required to actually sample at each time t for $t = 1, \dots, N$. For instance, magnetometers in smartphones often sample either at equal or half the sampling frequencies of the inertial sensors, while position aiding sensors like for instance GPS or UWB typically sample at much lower sampling frequencies.

Our aim is now to infer information about the states $x_{1:N}$ and the parameters θ using the measurements $y_{1:N}$ and the probabilistic models. This can be expressed in terms of a *conditional probability distribution*

$$p(x_{1:N}, \theta \mid y_{1:N}), \quad (13)$$

where $p(a \mid b)$ denotes the conditional probability of a given b . In the pose estimation problem, we are interested in obtaining *point estimates* which we denote $\hat{x}_{1:N}$ and $\hat{\theta}$. It is typically also highly relevant to know how *certain* we are about these estimates. This is often expressed in terms of a *covariance*. When the distribution (13) is Gaussian, the distribution is completely described in terms of its mean and covariance.

In (13) we assume that all measurements $y_{1:N}$ are used to obtain the posterior distribution of $x_{1:N}, \theta$. This is referred to as *smoothing*. Although it makes sense to use all available information to obtain the best estimates, a downside of smoothing is that we need to wait until all measurements are collected before the pose can be computed. Because of this, in many applications, we are also interested in *filtering*. In filtering we estimate x_t using all measurements up to and including time t . One way of dealing with constant parameters in filtering is to treat them as slowly time-varying. In this case, they can be considered to be included in the time-varying states x_t . The filtering problem can be expressed in terms of the

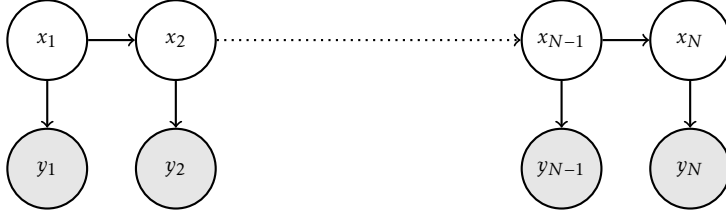


Figure 13: An illustration of the structure of the pose estimation problem.

conditional probability distribution

$$p(x_t | y_{1:t}). \quad (14)$$

We have now introduced smoothing, where the states $x_{1:N}$ are estimated simultaneously, and filtering, where at each time instance the state x_t is estimated. There is a large range of intermediate methods, where a batch of states $x_{t-L_1:t+L_2}$, with L_1 and L_2 being positive integers, is estimated using the measurements $y_{1:t}$. This is related to fixed-lag smoothing and moving horizon estimation (Johansen, 2011; Rao et al., 2001).

The topic of how to estimate the conditional probability distributions for position and orientation estimation will be introduced in Section 4. We will now instead take a closer look at the distributions and their different components. A fundamental assumption here is that we assume that our models possess the *Markov property*, implying that all information up to the current time t is contained in the state x_t . This is illustrated in Figure 13 in terms of a *probabilistic graphical model* (Bishop, 2006). The state x_{t+1} can be seen to depend on x_t and to result in the measurements y_{t+1} . It is conditionally independent of $x_{1:t-1}$ given the state x_t . Using Bayes' rule and the Markov property, the conditional distributions (13) and (14) can be decomposed as

$$p(x_{1:N}, \theta | y_{1:N}) \propto p(\theta)p(x_1) \prod_{t=2}^N p(x_t | x_{t-1}, \theta) \prod_{t=1}^N p(y_t | x_t, \theta), \quad (15a)$$

$$p(x_t | y_{1:t}) \propto p(y_t | x_t)p(x_t | y_{1:t-1}). \quad (15b)$$

The predictive distribution $p(x_t | y_{1:t-1})$ can be computed by *marginalizing out* the previous state x_{t-1} as

$$p(x_t | y_{1:t-1}) = \int p(x_t | x_{t-1})p(x_{t-1} | y_{1:t-1})dx_{t-1}. \quad (16)$$

In (15), $p(\theta)$ and $p(x_1)$ are *prior distributions* over θ and x_1 , respectively. The *dynamics* are modeled in terms of $p(x_{t+1} | x_t, \theta)$ and $p(x_{t+1} | x_t)$. The distributions $p(y_t | x_t, \theta)$ and $p(y_t | x_t)$ model the information given by the measurements about the state and the parameters.

The dynamics of the state can be modeled in terms of a nonlinear function $f_t(\cdot)$ as

$$x_{t+1} = f_t(x_t, w_t). \quad (17)$$

The uncertainty of the dynamic model is modeled in terms of w_t , which is often referred to as the *process noise*. The model (17) provides information about the distribution $p(x_{t+1} | x_t)$. More explicitly, if w_t is Gaussian additive noise with $w_t \sim \mathcal{N}(0, Q)$, then

$$p(x_{t+1} | x_t) \sim \mathcal{N}(x_{t+1}; f_t(x_t), Q), \quad (18)$$

where we use the notation $\mathcal{N}(x_{t+1}; f_t(x_t), Q)$ to explain that the random variable x_{t+1} is normal distributed with mean $f_t(x_t)$ and covariance Q .

The information given by the measurements about the state x_t can be modeled as

$$y_t = h_t(x_t, e_t), \quad (19)$$

where $h_t(\cdot)$ is a possibly nonlinear function and e_t is the measurement noise. The measurement model (19) provides information about the distribution $p(y_t | x_t)$. The combination of (17), (19) and a model of the prior $p(x_1)$ is referred to as a *state space model* (Kailath, 1980) and it is widely used in a large number of fields.

3.2 Parametrizing orientation

Rotating a vector in $\mathbb{R}^3$ changes the *direction* of the vector while retaining its *length*. The group of rotations in $\mathbb{R}^3$ is the special orthogonal group $\text{SO}(3)$. In this section we introduce four different ways of parametrizing orientations. Note that these describe the same quantity and can hence be used interchangeably. The different parametrizations can be converted to one another, see also Appendix A. There are differences in for instance the number of parameters used in the representation, the singularities and the uniqueness.

Rotation matrices

We encountered rotation matrices already in Section 2. Rotation matrices $R \in \mathbb{R}^{3 \times 3}$ have the following properties

$$RR^T = R^T R = \mathcal{I}_3, \quad \det R = 1. \quad (20)$$

The properties (20) provide an interpretation of the name special orthogonal group $\text{SO}(3)$. All orthogonal matrices of dimension 3×3 have the property $RR^T = R^T R = \mathcal{I}_3$ and are part of the orthogonal group $\text{O}(3)$. The notion *special* in $\text{SO}(3)$ specifies that only matrices with $\det R = 1$ are considered rotations.

Consider two coordinate frames denoted u and v . As was illustrated in Example 2, a vector x expressed in the u -frame can be rotated to the v -frame as

$$x^u = R^{uv} x^v, \quad (21a)$$

and conversely we have

$$x^v = (R^{uv})^T x^u = R^{vu} x^u. \quad (21b)$$

A rotation matrix is a unique description of the orientation. It has 9 components which depend on each other through (20).



Figure 14: Left: clockwise rotation α of the vector x to the vector $x_\star$. Right: counterclockwise rotation α of the coordinate frame v to the coordinate frame u .

Rotation vector

As described by Leonhard Euler in Euler (1776), a rotation around a point is always equivalent to a single rotation around some axis through this point, see Palais et al. (2009) for a number of proofs. This is generally referred to as *Euler's rotation theorem*. Hence, it is possible to express the rotation between two coordinate frames in terms of an angle α and a unit axis n around which the rotation takes place. In this section, we will derive a relation between the representation α, n and the rotation matrix parametrization from the previous section. Instead of directly considering the rotation of a coordinate frame, we start by considering the rotation of a vector. Note that a counterclockwise rotation of the coordinate frame is equivalent to a clockwise rotation of a vector, see Example 6.

Example 6: Rotation of a coordinate frame and rotation of a vector

Consider the 2D example in Figure 14, where on the left, a vector x is rotated clockwise by an angle α to $x_\star$. This is equivalent to (on the right) rotating the coordinate frame v counterclockwise by an angle α . Note that $x_\star^v = x^u$.

In Figure 15, a vector x is rotated an angle α around the unit axis n . We denote the rotated vector by $x_\star$. Suppose that x as expressed in the coordinate frame v is known (and denoted x^v) and that we want to express $x_\star^v$ in terms of x^v , α and n . It can first be recognized that the vector x can be decomposed into a component parallel to the axis n , denoted $x_\parallel$, and a component orthogonal to it, denoted $x_\perp$ as

$$x^v = x_\parallel^v + x_\perp^v. \quad (22a)$$

Based on geometric reasoning we can conclude that

$$x_\parallel^v = (x^v \cdot n^v) n^v, \quad (22b)$$

where $\cdot$ denotes the inner product. Similarly, $x_\star^v$ can be decomposed as

$$x_\star^v = (x_\star^v)_\parallel + (x_\star^v)_\perp, \quad (23a)$$

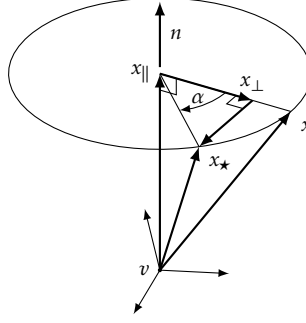


Figure 15: Clockwise rotation of a vector x by an angle α around the unit axis n . The rotated vector is denoted by $x_\star$. The vector x is decomposed in a component $x_\parallel$ that is parallel to the axis n , and a component $x_\perp$ that is orthogonal to it.

where

$$(x_\star^v)_\parallel = x_\parallel^v, \quad (23b)$$

$$(x_\star^v)_\perp = x_\perp^v \cos \alpha + (x^v \times n^v) \sin \alpha. \quad (23c)$$

Hence, $x_\star^v$ can be expressed in terms of x^v as

$$\begin{aligned} x_\star^v &= (x^v \cdot n^v) n^v + (x^v - (x^v \cdot n^v) n^v) \cos \alpha + (x^v \times n^v) \sin \alpha \\ &= x^v \cos \alpha + n^v (x^v \cdot n^v) (1 - \cos \alpha) - (n^v \times x^v) \sin \alpha. \end{aligned} \quad (24)$$

Denoting the rotated coordinate frame the u -frame and using the equivalence between $x_\star^v$ and x^u as shown in Example 6, this implies that

$$x^u = x^v \cos \alpha + n^v (x^v \cdot n^v) (1 - \cos \alpha) - (n^v \times x^v) \sin \alpha. \quad (25)$$

This equation is commonly referred to as the *rotation formula* or *Euler's formula* (Shuster, 1993). Note that the combination of n and α , or equivalently $\eta = n\alpha$, is denoted as the *rotation vector* or the *axis-angle parameterization*.

To show the equivalence between (25) and the rotation matrix parametrization, we will rewrite (25). Here, we make use of the fact that a cross product can equivalently be written as a matrix vector product. Given vectors u and v we have,

$$u \times v = [u \times] v = -[v \times] u, \quad [u \times] \triangleq \begin{pmatrix} 0 & -u_3 & u_2 \\ u_3 & 0 & -u_1 \\ -u_2 & u_1 & 0 \end{pmatrix}, \quad (26)$$

where u_1, u_2, u_3 denote the three components of the vector u . Furthermore, given vectors u, v and w , multiple cross products can be expanded in terms of the inner product as

$$u \times (v \times w) = v(w \cdot u) - w(u \cdot v). \quad (27)$$

Using these relations, (25) can be rewritten as

$$\begin{aligned}
 x^u &= x^v \cos \alpha + n^v (x^v \cdot n^v) (1 - \cos \alpha) - (n^v \times x^v) \sin \alpha \\
 &= x^v \cos \alpha + (n^v \times (n^v \times x^v) + x^v) (1 - \cos \alpha) - (n^v \times x^v) \sin \alpha \\
 &= (\mathcal{I}_3 - \sin \alpha [n^v \times] + (1 - \cos \alpha) [n^v \times]^2) x^v.
 \end{aligned} \tag{28}$$

Comparing (28) and (21a), it can be seen that a rotation matrix can be parametrized in terms of α, n as

$$R^{uv}(n^v, \alpha) = \mathcal{I}_3 - \sin \alpha [n^v \times] + (1 - \cos \alpha) [n^v \times]^2. \tag{29}$$

Note that equivalently, $R^{uv}(n^v, \alpha)$ can also be written as

$$R^{uv}(n^v, \alpha) = \exp(-\alpha [n^v \times]), \tag{30}$$

since

$$\begin{aligned}
 \exp(-\alpha [n^v \times]) &= \sum_{k=0}^{\infty} \frac{1}{k!} (-\alpha [n^v \times])^k \\
 &= \mathcal{I}_3 - \alpha [n^v \times] + \frac{1}{2!} \alpha^2 [n^v \times]^2 + \frac{1}{3!} \alpha^3 [n^v \times] - \frac{1}{4!} \alpha^4 [n^v \times]^2 - \dots \\
 &= \mathcal{I}_3 - \left(\alpha - \frac{1}{3!} \alpha^3 + \dots \right) [n^v \times] + \left(\frac{1}{2!} \alpha^2 - \frac{1}{4!} \alpha^4 + \dots \right) [n^v \times]^2 \\
 &= \mathcal{I}_3 - \sin \alpha [n^v \times] + (1 - \cos \alpha) [n^v \times]^2.
 \end{aligned} \tag{31}$$

The rotation vector introduced in this section parametrizes the orientation in only three parameters. However, it is not a unique parametrization since adding 2π to any angle α results in the same orientation. As shown in (29) and (30), the rotation matrix can straightforwardly be expressed in terms of the axis-angle representation.

Euler angles

Rotation can also be defined as a consecutive rotation around three axes in terms of so-called *Euler angles*. We use the convention (z, y, x) which first rotates an angle ψ around the z -axis, subsequently an angle θ around the y -axis and finally an angle ϕ around the x -axis. These angles are illustrated in Figure 16. Assuming that the v -frame is rotated by (ψ, θ, ϕ) with respect to the u -frame as illustrated in this figure, the rotation matrix R^{uv} is given by

$$\begin{aligned}
 R^{uv} &= R^{uv}(e_1, \phi) R^{uv}(e_2, \theta) R^{uv}(e_3, \psi) \\
 &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{pmatrix} \begin{pmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{pmatrix} \begin{pmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \\
 &= \begin{pmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \sin \phi \cos \theta \\ \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi & \cos \phi \cos \theta \end{pmatrix}, \tag{32}
 \end{aligned}$$

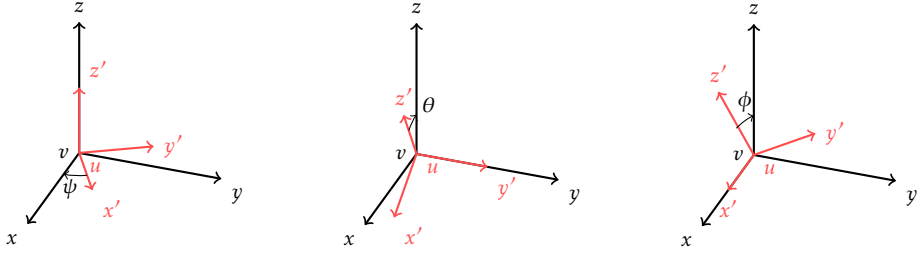


Figure 16: Definition of Euler angles as used in this work with left: rotation ψ around the z -axis, middle: rotation θ around the y -axis and right: rotation ϕ around the x -axis.

where we make use of the notation introduced in (29) and the following definition of the unit vectors

$$e_1 = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}^T, \quad e_2 = \begin{pmatrix} 0 & 1 & 0 \end{pmatrix}^T, \quad e_3 = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}^T. \quad (33)$$

The ψ, θ, ϕ angles are also often referred to as yaw (or heading), pitch and roll, respectively. Furthermore, roll and pitch together are often referred to as inclination.

Similar to the rotation vector, Euler angles parametrize orientation as a three-dimensional vector. Euler angle representations are not unique descriptions of a rotation for two reasons. As can be seen from (32), for instance the rotation $(0, 0, 0)$ is equal to $(0, 0, 2\pi)$. This is called *wrapping*. Furthermore, setting $\theta = \frac{\pi}{2}$ in (32), leads to

$$\begin{aligned} R^{uv} &= \begin{pmatrix} 0 & 0 & -1 \\ \sin \phi \cos \psi - \cos \phi \sin \psi & \sin \phi \sin \psi + \cos \phi \cos \psi & 0 \\ \cos \phi \cos \psi + \sin \phi \sin \psi & \cos \phi \sin \psi - \sin \phi \cos \psi & 0 \end{pmatrix} \\ &= \begin{pmatrix} 0 & 0 & -1 \\ \sin(\phi - \psi) & \cos(\phi - \psi) & 0 \\ \cos(\phi - \psi) & -\sin(\phi - \psi) & 0 \end{pmatrix}. \end{aligned} \quad (34)$$

Hence, only the rotation $\phi - \psi$ can be observed. Because of this, for example the rotations $(\frac{\pi}{2}, \frac{\pi}{2}, 0)$, $(0, \frac{\pi}{2}, -\frac{\pi}{2})$, $(\pi, \frac{\pi}{2}, \frac{\pi}{2})$ are all three equivalent. This is called *gimbal lock* (Diebel, 2006).

Unit quaternions

A commonly used parametrization of orientation is that of unit quaternions. Quaternions were first introduced by Hamilton (1844) and are widely used in orientation estimation algorithms, see e.g. Kuipers (1999); Hol (2011). A unit quaternion use a 4-dimensional representation of the orientation according to

$$q = \begin{pmatrix} q_0 & q_1 & q_2 & q_3 \end{pmatrix}^T = \begin{pmatrix} q_0 \\ q_v \end{pmatrix}, \quad q \in \mathbb{R}^4, \quad \|q\|_2 = 1. \quad (35)$$

A unit quaternion is not a unique description of an orientation. The reason for this is that if q represents a certain orientation, then $-q$ describes the same orientation.

A rotation can be defined using unit quaternions as

$$\bar{x}^u = q^{uv} \odot \bar{x}^v \odot (q^{uv})^c, \quad (36)$$

where $\cdot^c$ denotes the quaternion conjugate, defined as

$$q^c = \begin{pmatrix} q_0 & -q_v^\top \end{pmatrix}^\top, \quad (37)$$

and $\bar{x}^v$ denotes the quaternion representation of x^v as

$$\bar{x}^v = \begin{pmatrix} 0 & (x^v)^\top \end{pmatrix}^\top. \quad (38)$$

Note that (38) is typically not a unit quaternion. The notation $\odot$ denotes the quaternion multiplication given by

$$p \odot q = \begin{pmatrix} p_0 q_0 - p_v \cdot q_v \\ p_0 q_v + q_0 p_v + p_v \times q_v \end{pmatrix} = p^L q = q^R p, \quad (39)$$

where

$$p^L \triangleq \begin{pmatrix} p_0 & -p_v^\top \\ p_v & p_0 \mathcal{I}_3 + [p_v \times] \end{pmatrix}, \quad q^R \triangleq \begin{pmatrix} q_0 & -q_v^\top \\ q_v & q_0 \mathcal{I}_3 - [q_v \times] \end{pmatrix}. \quad (40)$$

Using (37) – (40), (36) can be written as

$$\begin{aligned} \bar{x}^u &= (q^{uv})^L (q^{vu})^R \bar{x}^v \\ &= \begin{pmatrix} q_0 & -q_v^\top \\ q_v & q_0 \mathcal{I}_3 + [q_v \times] \end{pmatrix} \begin{pmatrix} q_0 & q_v^\top \\ -q_v & q_0 \mathcal{I}_3 + [q_v \times] \end{pmatrix} \begin{pmatrix} 0 \\ x^v \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0_{1 \times 3} \\ 0_{3 \times 1} & q_v q_v^\top + q_0^2 \mathcal{I}_3 + 2q_0 [q_v \times] + [q_v \times]^2 \end{pmatrix} \begin{pmatrix} 0 \\ x^v \end{pmatrix}. \end{aligned} \quad (41)$$

Comparing (41) to (29), it can be recognized that if we choose

$$q^{uv}(n^v, \alpha) = \begin{pmatrix} \cos \frac{\alpha}{2} \\ -n^v \sin \frac{\alpha}{2} \end{pmatrix}, \quad (42)$$

the two rotation formulations are equivalent since

$$\begin{aligned} \bar{x}^u &= \begin{pmatrix} 1 & 0_{1 \times 3} \\ 0_{3 \times 1} & \mathcal{I}_3 - 2 \cos \frac{\alpha}{2} \sin \frac{\alpha}{2} [n^v \times] + 2 \sin^2 \frac{\alpha}{2} [n^v \times]^2 \end{pmatrix} \begin{pmatrix} 0 \\ x^v \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0_{1 \times 3} \\ 0_{3 \times 1} & \mathcal{I}_3 - \sin \alpha [n^v \times] + (1 - \cos \alpha) [n^v \times]^2 \end{pmatrix} \begin{pmatrix} 0 \\ x^v \end{pmatrix}. \end{aligned} \quad (43)$$

Here, we made use of standard trigonometric relations and the fact that since $\|n^v\|_2 = 1$, $n^v (n^v)^\top = \mathcal{I}_3 + [n^v \times]^2$. Hence, it can be concluded that q^{uv} can be expressed in terms of α and n^v as in (42).

Equivalently, $q^{uv}(n^v, \alpha)$ can also be written as

$$q^{uv}(n^v, \alpha) = \exp(-\frac{\alpha}{2} \bar{n}^v) = \sum_{k=0}^{\infty} \frac{1}{k!} \left(-\frac{\alpha}{2} \bar{n}^v\right)^k, \quad (44)$$

where

$$(\bar{n}^v)^0 = \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}^T, \quad (45a)$$

$$(\bar{n}^v)^1 = \begin{pmatrix} 0 & (n^v)^T \end{pmatrix}^T, \quad (45b)$$

$$(\bar{n}^v)^2 = \bar{n}^v \odot \bar{n}^v = \begin{pmatrix} -\|n^v\|_2^2 & 0_{3 \times 1} \end{pmatrix}^T = \begin{pmatrix} -1 & 0_{3 \times 1} \end{pmatrix}^T, \quad (45c)$$

$$(\bar{n}^v)^3 = \begin{pmatrix} 0 & -(n^v)^T \end{pmatrix}^T, \quad (45d)$$

This leads to

$$\begin{aligned} q^{uv}(n^v, \alpha) &= \exp(-\frac{\alpha}{2} \bar{n}^v) = \sum_{k=0}^{\infty} \frac{1}{k!} \left(-\frac{\alpha}{2} \bar{n}^v\right)^k \\ &= \begin{pmatrix} 1 - \frac{1}{2!} \frac{\alpha^2}{4} + \frac{1}{4!} \frac{\alpha^4}{16} - \dots \\ -\frac{\alpha}{2} n^v + \frac{1}{3!} \frac{\alpha^3}{8} n^v - \frac{1}{5!} \frac{\alpha^5}{32} n^v + \dots \end{pmatrix} \\ &= \begin{pmatrix} \cos \frac{\alpha}{2} \\ -n^v \sin \frac{\alpha}{2} \end{pmatrix}. \end{aligned} \quad (46)$$

Note the similarity to (30) and (31). The reason for why both rotation matrices and unit quaternions can be described in terms of an exponential of a rotation vector will be discussed in Section 3.3.

3.3 Probabilistic orientation modeling

The four parametrizations of orientation discussed in Section 3.2 can be used interchangeably. However, the choice of which parametrization to use as states x_t in the filtering and smoothing problems introduced in Section 3.1 has significant impact on the workings of the algorithm. An important reason for this is that estimation algorithms typically assume that the unknown states and parameters are represented in Euclidean space. For instance, they assume that the subtraction of two orientations gives information about the difference in orientation and that the addition of two orientations is again a valid orientation. For the four parametrizations discussed in Section 3.2, this is generally not true. For instance, due to wrapping and gimbal lock, subtraction of Euler angles and rotation vectors can result in large numbers even in cases when the rotations are similar. Also, addition and subtraction of unit quaternions and rotation matrices do not in general lead to a valid rotation. The equality constraints on the norm of unit quaternions and on the determinant and the orthogonality of rotation matrices are typically hard to include in the estimation algorithms. In this section, we will discuss a method to represent orientation in estimation algorithms that deals with

the issues described above. It is frequently used in the algorithms that will be described in Section 4. We will also discuss some alternative methods to parametrize orientation for estimation purposes.

Linearization

As mentioned in Section 3.2, the group of rotations in three dimensions is the special orthogonal group $SO(3)$. More specifically, $SO(3)$ is a so-called *matrix Lie group*. For a discussion on the properties of matrix Lie groups and on the reasons why $SO(3)$ is indeed such a group we refer the reader to e.g. Barfoot (2016). Since rotations are a matrix Lie group, there exists an *exponential map* from a corresponding Lie algebra. Using this property, it is possible to represent orientations on $SO(3)$ using unit quaternions or rotation matrices, while orientation deviations are represented using rotation vectors on $\mathbb{R}^3$, see e.g. Bloesch et al. (2016). Hence, we encode an orientation q_t^{nb} in terms of a *linearization point* parametrized either as a unit quaternion $\tilde{q}_t^{\text{nb}}$ or as a rotation matrix $\tilde{R}_t^{\text{nb}}$ and an *orientation deviation* using a rotation vector η_t . Assuming that the orientation deviation is expressed in the body frame b ,³

$$q_t^{\text{nb}} = \tilde{q}_t^{\text{nb}} \odot \exp\left(\frac{\tilde{\eta}_t^{\text{b}}}{2}\right), \quad R_t^{\text{nb}} = \tilde{R}_t^{\text{nb}} \exp\left([\eta_t^{\text{b}} \times]\right), \quad (47)$$

where analogously to (46) and (31),

$$\exp(\tilde{\eta}) = \left(\frac{\cos \|\eta\|_2}{\frac{\eta}{\|\eta\|_2} \sin \|\eta\|_2} \right), \quad (48a)$$

$$\exp([\eta \times]) = \mathcal{I}_3 + \sin(\|\eta\|_2) \left[\frac{\eta}{\|\eta\|_2} \times \right] + (1 - \cos(\|\eta\|_2)) \left[\frac{\eta}{\|\eta\|_2} \times \right]^2. \quad (48b)$$

For notational convenience, in the remainder we will use the mappings

$$q = \exp_{\text{q}}(\eta), \quad \exp_{\text{q}} : \mathbb{R}^3 \rightarrow \{q \in \mathbb{R}^4 : \|q\|_2 = 1\}, \quad (49a)$$

$$R = \exp_{\text{R}}(\eta), \quad \exp_{\text{R}} : \mathbb{R}^3 \rightarrow \{R \in \mathbb{R}^{3 \times 3} : RR^{\text{T}} = \mathcal{I}_3, \det R = 1\}, \quad (49b)$$

which allow us to rewrite (47) as

$$q_t^{\text{nb}} = \tilde{q}_t^{\text{nb}} \odot \exp_{\text{q}}\left(\frac{\tilde{\eta}_t^{\text{b}}}{2}\right), \quad R_t^{\text{nb}} = \tilde{R}_t^{\text{nb}} \exp_{\text{R}}\left(\eta_t^{\text{b}}\right). \quad (50)$$

The reverse mappings are defined as

$$\log_{\text{q}}(q) = \frac{\arccos q_0}{\sin \arccos q_0} q_v = \frac{\arccos q_0}{\|q_v\|_2} q_v, \quad \log_{\text{q}} : \{q \in \mathbb{R}^4 : \|q\|_2 = 1\} \rightarrow \mathbb{R}^3, \quad (51a)$$

$$\log_{\text{R}}(R) = \begin{pmatrix} (\log R)_{32} \\ (\log R)_{13} \\ (\log R)_{21} \end{pmatrix}, \quad \log_{\text{R}} : \mathbb{R}^3 \rightarrow \{R \in \mathbb{R}^{3 \times 3} : RR^{\text{T}} = \mathcal{I}_3, \det R = 1\}, \quad (51b)$$

³A similar derivation can be done by assuming an orientation deviation in the navigation frame n .

where $\log R$ is the standard matrix logarithm. Since we typically assume that η_t^b is small, we will frequently make use of the following approximations

$$\exp_q(\eta) \approx \begin{pmatrix} 1 \\ \eta \end{pmatrix}, \quad \log_q(q) \approx q_v, \quad (52a)$$

$$\exp_R(\eta) \approx \mathcal{I}_3 + [\eta \times], \quad \log_R(R) \approx \begin{pmatrix} R_{32} & R_{13} & R_{21} \end{pmatrix}^T. \quad (52b)$$

The idea briefly outlined in this section is closely related to approaches used to estimate orientation in robotics, see e.g. Grisetti et al. (2010a,b); Bloesch et al. (2016); Barfoot (2016); Forster et al. (2016). It is also related to the so-called multiplicative extended Kalman filter (MEKF) frequently used in aeronautics, see e.g. Markley (2003); Crassidis et al. (2007).

Alternative methods

An alternative method to estimate orientation assumes that the states representing the orientation lie on a manifold. This can be done by modeling the orientation and its uncertainty using a *spherical distribution* which naturally restricts the orientation estimates and their uncertainties to be in $SO(3)$. In recent years, a number of approaches have been proposed to estimate the orientation using these kinds of distributions. For instance, in Kurz et al. (2013); Gilitschenski et al. (2016); Glover and Kaelbling (2013) algorithms are presented to estimate orientation by modeling it using a Bingham distribution.

The difficulties caused by directly using one of the four orientation parametrizations introduced in Section 3.2 in orientation estimation algorithms is widely recognized. Nevertheless, a large number of approaches directly uses these parametrizations in estimation algorithms. For instance, it is common practice to use unit quaternions in estimation algorithms and to normalize the resulting quaternions each time they lose their normalization, see e.g. Sabatini (2006); Marins et al. (2001); Madgwick et al. (2011). Different approaches to handle the normalization of the quaternions in these algorithms are discussed in Julier and LaViola Jr. (2007).

3.4 Measurement models

In the past two sections, we have focused on how orientations can be parametrized. In this section, we will go back to the probabilistic models for the position and orientation estimation problems introduced in Section 3.1 and provide different measurement models $p(y_t \mid x_t, \theta)$.

Gyroscope measurement models

As discussed in Section 2.2, the gyroscope measures the angular velocity ω_{ib}^b at each time instance t . However, as shown in Section 2.4, its measurements are

corrupted by a slowly time-varying bias $\delta_{\omega,t}$ and noise $e_{\omega,t}$. Hence, the gyroscope measurement model is given by

$$y_{\omega,t} = \omega_{ib,t}^b + \delta_{\omega,t}^b + e_{\omega,t}^b. \quad (53)$$

As was shown in Figure 9, the gyroscope measurement noise is typically quite Gaussian. Because of this, it is typically assumed that $e_{\omega,t}^b \sim \mathcal{N}(0, \Sigma_{\omega})$. If the sensor is properly calibrated, the measurements in the three gyroscope axes are independent. In that case, it can be assumed that

$$\Sigma_{\omega} = \begin{pmatrix} \sigma_{\omega,x}^2 & 0 & 0 \\ 0 & \sigma_{\omega,y}^2 & 0 \\ 0 & 0 & \sigma_{\omega,z}^2 \end{pmatrix}. \quad (54)$$

The gyroscope bias $\delta_{\omega,t}^b$ is slowly time-varying, as discussed in Section 2.4. There are two conceptually different ways to treat this slowly time-varying bias. One is to treat the bias as a constant parameter, assuming that it typically changes over a longer time period than the time of the experiment. The bias can then either be pre-calibrated in a separate experiment, or it can be considered to be part of the unknown parameters θ as introduced in Section 3.1. Alternatively, it can be assumed to be slowly time-varying. This can be justified either by longer experiment times or by shorter bias stability. In the latter case, $\delta_{\omega,t}^b$ can instead be considered as part of the state vector x_t and can for instance be modeled as a random walk

$$\delta_{\omega,t+1}^b = \delta_{\omega,t}^b + e_{\delta_{\omega,t}}^b, \quad (55)$$

where $e_{\delta_{\omega,t}}^b \sim \mathcal{N}(0, \Sigma_{\delta_{\omega,t}})$ represents how constant the gyroscope bias actually is.

Modeling the sensor noise and bias is related to the sensor properties. However, there are also modeling choices related to the experiments that can be made. As described in Section 2.2, the angular velocity ω_{ib}^b can be expressed as

$$\omega_{ib,t}^b = R_t^{bn} (\omega_{ie,t}^n + \omega_{en,t}^n) + \omega_{nb,t}^b. \quad (56)$$

If the sensor does not travel over significant distances as compared to the size of the earth – which is often the case for the applications discussed in Section 1 – the navigation frame n can safely be assumed to be stationary. In that case, the transport rate $\omega_{en,t}^n$ is zero. Although the earth rotation ω_{ie} as expressed in the body frame b is not constant, its magnitude as compared to the magnitude of the actual measurements is fairly small (see Section 2.2 and the experimental data presented in Example 4). Assuming that the earth rotation is negligible and the navigation frame is stationary leads to the following simplified measurement model

$$y_{\omega,t} = \omega_{nb,t}^b + \delta_{\omega,t}^b + e_{\omega,t}^b. \quad (57)$$

Accelerometer measurement models

The accelerometer measures the specific force f_t^b at each time instance t , see also Section 2.3. As shown in Section 2.4, the accelerometer measurements are typically assumed to be corrupted by a bias $\delta_{a,t}$ and noise $e_{a,t}$ as

$$y_{a,t} = f_t^b + \delta_{a,t}^b + e_{a,t}^b. \quad (58)$$

The accelerometer noise is typically quite Gaussian as was shown in Figure 10 and can hence be modeled as $e_{a,t}^b \sim \mathcal{N}(0, \Sigma_a)$. For a properly calibrated sensor, the covariance matrix Σ_a can often be assumed to be diagonal.

The accelerometer bias $\delta_{a,t}^b$ is slowly time-varying. Similar to the gyroscope bias, the accelerometer bias can either be modeled as a constant parameter, or as part of the time-varying state, for instance using a random walk model as in (55).

As introduced in Section 2.3, the specific force measured by the accelerometer is given by

$$f^b = R^{bn}(a_{ii}^n - g^n). \quad (59)$$

Assuming that the navigation frame is fixed to the earth frame, we derived a relation for a_{ii}^n as

$$a_{ii}^n = a_{nn}^n + 2\omega_{ie}^n \times v_n^n + \omega_{ie}^n \times \omega_{ie}^n \times p^n. \quad (60)$$

The centrifugal acceleration $\omega_{ie}^n \times \omega_{ie}^n \times p^n$ is typically absorbed in the local gravity vector. The magnitude of the Coriolis acceleration is small compared to the magnitude of the accelerometer measurements (see Example 3 and the experimental data presented in Example 4). Neglecting this term leads to the following simplified measurement model

$$y_{a,t} = R_t^{bn}(a_{nn}^n - g^n) + \delta_{a,t}^b + e_{a,t}^b. \quad (61)$$

Since the accelerometer measures both the local gravity vector and the linear acceleration of the sensor, it provides information both about the change in position and about the inclination of the sensor. For orientation estimation, only the information about the inclination is of concern. Hence, a model for the linear acceleration needs to be made to express the relation between the inclination and the measurements. To model this, it can be recognized that in practice, most accelerometer measurements are dominated by the gravity vector, as illustrated in Example 7.

Example 7: Magnitude of a sensor's linear acceleration

Let us consider a 1D example where a sensor has an initial velocity $v_1 = 0$ m/s and accelerates with $a_{nn}^n = 9.82$ m/s². After 4.51 s, the sensor will have traveled 100 m. This is about twice as fast as the world record currently held by Usain Bolt. In fact, humans can reach fairly high accelerations but can only accelerate for a short time. Naturally, cars can accelerate to higher velocities than humans. The sensor in this example has reached a final velocity of 160 km/h. Even in the case of a car it is therefore unlikely that it can have an acceleration this high for a long time.

Since the accelerometer measurements are typically dominated by the gravity vector, a commonly used model assumes the linear acceleration to be approximately zero

$$y_{a,t} = -R_t^{\text{bn}} g^n + \delta_{a,t}^b + e_{a,t}^b. \quad (62)$$

Naturally, the model (62) is almost never completely true. However, it can often be used as a sufficiently good approximation of reality. Note that the noise term $e_{a,t}^b$ in this case does not only represent the measurement noise, but also the model uncertainty. The model (62) can for instance be used in combination with *outlier rejection* where measurements that clearly violate the assumption that the linear acceleration is zero are disregarded. It is also possible to adapt the noise covariance matrix Σ_a , depending on the sensor's acceleration (Foxlin, 1996; Rehbinder and Hu, 2004). Furthermore, it is possible to model the acceleration based on physical reasoning (Luinge, 2002).

Modeling additional information

In this section we will discuss models for the measurements we use to complement the inertial sensor measurements. For orientation estimation we use magnetometers, while for pose estimation we use position measurements.

Magnetometer models Magnetometers measure the local magnetic field, consisting of both the earth magnetic field and the magnetic field due to the presence of magnetic material. The (local) earth magnetic field is denoted m^n and it is illustrated in Figure 17. Its horizontal component points towards the earth's magnetic north pole. The ratio between the horizontal and vertical component depends on the location on the earth and can be expressed in terms of the so-called dip angle δ . The dip angle and the magnitude of the earth magnetic field are accurately known from geophysical studies, see e.g. National Centers for Environmental Information (2016).

Assuming that the sensor does not travel over significant distances as compared to the size of the earth, the local earth magnetic field can safely be modeled as being constant. In case no magnetic material is present in the vicinity of the sensor, orientation information can be deduced from the magnetometer. More specifically, magnetometers are typically used to complement accelerometers to provide information about the sensor heading, i.e. about the orientation around the gravity vector which can not be determined from the accelerometer measurements. Magnetometers provide information about the heading in all locations on the earth except on the magnetic poles, where the local magnetic field m^n is vertical. Orientation can be estimated based on the *direction* of the magnetic field. The *magnitude* of the field is irrelevant. Because of this, without loss of generality we model the earth magnetic field as

$$m^n = \begin{pmatrix} \cos \delta & 0 & \sin \delta \end{pmatrix}^T, \quad (63)$$

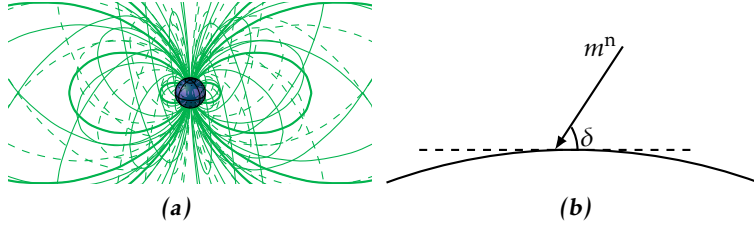


Figure 17: (a) Schematic of the earth magnetic field lines (green) around the earth (blue).⁴ (b) Schematic of a part of the earth where the local earth magnetic field m^n makes an angle δ with the horizontal plane. This angle is called the dip angle.

i.e. we assume that $\|m^n\|_2 = 1$. Assuming that the magnetometer only measures the local magnetic field, its measurements $y_{m,t}$ can be modeled as

$$y_{m,t} = R_t^{\text{bn}} m^n + e_{m,t}, \quad (64)$$

where $e_{m,t} \sim \mathcal{N}(0, \Sigma_m)$. The noise $e_{m,t}$ represents the magnetometer measurement noise as well as the model uncertainty.

In practice, the actual magnetic field can differ significantly from the earth magnetic field. In indoor environments, for instance, presence of magnetic material in the structure of buildings and in furniture influences the magnetic field that is measured by the magnetometer. Furthermore, the magnetic field is affected in applications where the magnetometer is mounted in e.g. a vehicle, train or on a robot. In case the magnetic material is rigidly attached to the sensor, the magnetometer can be calibrated for its presence (Kok and Schön, 2016; Vasconcelos et al., 2011; Renaudin et al., 2010; Salehi et al., 2012). The presence of magnetic material in the vicinity of the sensor that can not be calibrated for is of major concern for practical applications. Because of this, there is a vast amount of literature on the topic, see e.g. Callmer (2013); Ligorio and Sabatini (2016); Roetenberg et al. (2005).

Note that when using magnetometers for orientation estimation, the presence of magnetic material is typically considered to be an undesired disturbance. However, the presence of magnetic material can also be considered to be a property which can be exploited. This is done in approaches which use the magnetic field as a source of position information (Solin et al., 2015; Haverinen and Kempainen, 2009; Robertson et al., 2013).

Position information Position information can be obtained from for instance GPS or UWB measurements. In this tutorial, we will consider a very basic measurement model where the sensors directly measure the position as

$$y_{p,t} = p_t^n + e_{p,t}, \quad (65)$$

⁴Adapted version of ‘Dipolar magnetic field’ by Cyril Langlois available at <http://texample.net> under CC BY 2.5 (<http://creativecommons.org/licenses/by/2.5>).

with $e_{p,t} \sim \mathcal{N}(0, \Sigma_p)$. Many sensors do not measure the position directly. Their measurements can, however, be pre-processed to obtain position estimates and their corresponding covariances (Gustafsson and Gunnarsson, 2005). For example, time of arrival measurements can be pre-processed using multilateration techniques. Measurements of this type often contain a significant amount of outliers. The reason is that the signals can be delayed due to multipath or non-line-of-sight (NLOS) conditions. Possible solutions deal with this by doing outlier rejection, by using robust algorithms, see e.g. Zoubir et al. (2012), or by modeling the noise distribution as a non-Gaussian distribution, see e.g. Kok et al. (2015); Gustafsson and Gunnarsson (2005); Nurminen et al. (2015).

3.5 Choosing the state and modeling its dynamics

The fundamental continuous-time relations that form the basis of our dynamic models are the fact that the position p^n , velocity v_n^n and acceleration a_{nn}^n are related as

$$v_n^n = \left. \frac{dp^n}{dt} \right|_n, \quad a_{nn}^n = \left. \frac{dv^n}{dt} \right|_n, \quad (66)$$

and that the orientation and angular velocity $\omega_{nb,t}^b$ are related as

$$\frac{dq^{nb}}{dt} = q^{nb} \odot \frac{1}{2} \bar{\omega}_{nb}^b, \quad \frac{dR^{nb}}{dt} = R^{nb} [\omega_{nb}^b \times], \quad (67)$$

depending on orientation parametrization. For a derivation of (67), see e.g. Hol (2011). Using an Euler discretization of (66), the dynamics of the position and velocity can be expressed in terms of the acceleration as

$$p_{t+1}^n = p_t^n + T v_{n,t}^n + \frac{T^2}{2} a_{nn,t}^n, \quad (68a)$$

$$v_{n,t+1}^n = v_{n,t}^n + T a_{nn,t}^n, \quad (68b)$$

where T is the time between two samples. Similarly, the dynamics of the orientation can be expressed in terms of unit quaternions or rotation matrices as

$$q_{t+1}^{nb} = q_t^{nb} \odot \exp_q \left(\frac{T}{2} \omega_{nb,t}^b \right), \quad (69a)$$

$$R_{t+1}^{nb} = R_t^{nb} \exp_R \left(T \omega_{nb,t}^b \right). \quad (69b)$$

Dynamic models describe how the state changes over time. For the problem of position and orientation estimation using inertial sensors, there are two commonly used modeling alternatives for the dynamics (Gustafsson, 2012). In the first, we choose the state vector x_t to consists of

$$x_t = \left((p_t^n)^\top \quad (v_{n,t}^n)^\top \quad (a_{nn,t}^n)^\top \quad (q_t^{nb})^\top \quad (\omega_{nb,t}^b)^\top \right)^\top. \quad (70)$$

The change in position, velocity and orientation states can then be described in terms of the velocity, acceleration and angular velocity states, respectively. The

dynamics of the acceleration and the angular velocity can be described in terms of a *motion model*. Examples of motion models that can be used are a constant acceleration model, which assumes that the dynamics of the acceleration can be described as

$$a_{nn,t+1}^n = a_{nn,t}^n + w_{a,t}, \quad (71)$$

with $w_{a,t} \sim \mathcal{N}(0, \Sigma_{w,a})$, and a constant angular velocity model, which describes the dynamics of the angular velocity as

$$\omega_{nb,t+1}^b = \omega_{nb,t}^b + w_{\omega,t}, \quad (72)$$

with $w_{\omega,t} \sim \mathcal{N}(0, \Sigma_{w,\omega})$. The process noise terms $w_{a,t}$ and $w_{\omega,t}$ model the assumptions on how constant the acceleration and angular velocity actually are.

Alternatively, the state vector x_t can be chosen as

$$x_t = \left((p_t^n)^\top \quad (v_{n,t}^n)^\top \quad (q_t^{nb})^\top \right)^\top. \quad (73)$$

To describe the dynamics of the states, the inertial measurements can then be used as an *input* to the dynamic equation (17). Hence, the change in position, velocity and orientation is modeled directly in terms of the inertial measurements. In this case, expressions for $a_{nn,t}^n$ and $\omega_{nb,t}^b$ in (68) and (69) are obtained from the accelerometer measurement model and the gyroscope measurement model, see Section 3.4. The process noise can explicitly be modeled in terms of the accelerometer measurement noise $e_{a,t}$ and the gyroscope measurement noise $e_{\omega,t}$.

The benefit of using a motion model for the state dynamics is that knowledge about the motion of the sensor can be included in this model. However, it comes at the expense of having a larger state vector. The benefit of using the inertial measurements as an input to the dynamics is that the process noise has the intuitive interpretation of representing the inertial measurement noise. Hence, the latter approach is often used for applications where it is difficult to obtain sensible motion models.

3.6 Models for the prior

Looking back at Section 3.1, to solve the smoothing and filtering problems (15a) and (15b), we have now discussed different models for the dynamics $p(x_t | x_{t-1}, \theta)$ in Section 3.5 and for the measurements $p(y_t | x_t, \theta)$ in Section 3.4. The remaining distributions to be defined are the priors $p(x_1)$ and $p(\theta)$, which is the topic of this section.

In many cases, there is fairly little prior information available about the parameters θ . However, it is often possible to indicate what are reasonable values for the parameters. For example, it is reasonable to assume that the gyroscope bias is fairly small but can be both positive and negative. For instance, for the data presented in Example 4, the average bias over the entire data set was $(35.67 \quad 54.13 \quad -1.07)^\top \cdot 10^{-4}$ rad/s. If we would assume that in 68% of the cases,

the bias is within the bounds $-\sigma_{\delta_\omega} \leq \delta_{\omega,t}^b \leq \sigma_{\delta_\omega}$ with $\sigma_{\delta_\omega} = 5 \cdot 10^{-3}$, a reasonable prior would be

$$\delta_{\omega,t}^b \sim \mathcal{N}(0, \sigma_{\delta_\omega}^2 \mathcal{I}_3). \quad (74)$$

For the prior $p(x_1)$, it is typically possible to get a reasonable estimate from data. For the position and velocity, this can be modeled as

$$p_1^n = \check{p}_1^n + e_{p,i}, \quad e_{p,i} \sim \mathcal{N}(0, \sigma_{p,i}^2 \mathcal{I}_3), \quad (75a)$$

$$v_1^n = \check{v}_1^n + e_{v,i}, \quad e_{v,i} \sim \mathcal{N}(0, \sigma_{v,i}^2 \mathcal{I}_3). \quad (75b)$$

Here, the estimate $\check{p}_1^n$ can for instance be determined based on the first position measurement. In that case, the uncertainty $\sigma_{p,i}$ can also be chosen equal to the uncertainty of the position measurements. In case no additional information is available, the estimates $\check{p}_1^n$ and $\check{v}_1^n$ can be set to zero with an appropriate standard deviation instead.

A commonly used method to determine the initial orientation is to use the first accelerometer and magnetometer samples. This method is based on the fact that given two (or more) linearly independent vectors in two coordinate frames, the rotation between the two coordinate frames can be determined. The implicit assumption is that the accelerometer only measures the gravity vector and the magnetometer only measures the local magnetic field. Hence, the four vectors are given by the measurements $y_{a,t}$ and $y_{m,t}$, the local gravity vector g^n and the local magnetic field m^n . These vectors are linearly independent except when the measurements are obtained on the magnetic north or south poles where the dip angle is $\delta = 0$ and the magnetic field does not contain any heading information.

The accelerometer provides information about the sensor's inclination. Heading information is provided by the magnetometer. However, at all locations except on the equator, the magnetometer also provides information about the inclination due to its non-zero vertical component, see (63). In practice, the accelerometer typically provides more accurate inclination information. Hence, we choose to use the magnetometer only to provide heading information by projecting the magnetic field and the magnetometer measurement on the horizontal plane. Furthermore, we normalize the vectors. Because of this, an adapted model uses the four normalized vectors

$$\hat{g}^n = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}^T, \quad \hat{g}^b = \frac{y_{a,1}}{\|y_{a,1}\|_2}, \quad (76a)$$

$$\hat{m}^n = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}^T, \quad \hat{m}^b = \hat{g}^b \times \left(\frac{y_{m,1}}{\|y_{m,1}\|_2} \times \hat{g}^b \right). \quad (76b)$$

A number of algorithms are available to estimate the orientation from these vectors. Well-known examples are the TRIAD algorithm, the QUEST algorithm, see e.g. Shuster and Oh (1981), and the method presented in Horn (1987). For our problem at hand, these methods give equivalent results, even though they use slightly different solution strategies. Generally speaking, they solve the problem of determining the rotation q^{nb} from

$$\arg \min_{q^{nb}} \|\bar{\hat{g}}^n - q^{nb} \odot \bar{\hat{g}}^b \odot q^{bn}\|_2^2 + \|\bar{\hat{m}}^n - q^{nb} \odot \bar{\hat{m}}^b \odot q^{bn}\|_2^2. \quad (77)$$

Recall from (36) that $q^{\text{nb}} \odot \tilde{x}^{\text{b}} \odot q^{\text{bn}}$ is the rotation of the vector x^{b} to the n -frame. The optimization problem (77) therefore determines the orientation q^{nb} that minimizes the distance between the normalized magnetic field and gravity vectors measured in the first sample and the normalized magnetic field and gravity vectors in the navigation frame. These four vectors were defined in (76).

Defining

$$A = -(\tilde{g}^{\text{n}})^{\text{L}} (\tilde{g}^{\text{b}})^{\text{R}} - (\tilde{m}^{\text{n}})^{\text{L}} (\tilde{m}^{\text{b}})^{\text{R}}, \quad (78)$$

where the left and right quaternion multiplications are defined in (40), (77) can equivalently be written as

$$\check{q}_1^{\text{nb}} = \arg \min_{q^{\text{nb}}} (q^{\text{nb}})^{\text{T}} A q^{\text{nb}}. \quad (79)$$

For a derivation, see Hol (2011). The solution to this problem is given by the eigenvector corresponding to the largest eigenvalue. Note that although these methods can be used to compute the orientation from any two linearly independent vectors in two coordinate frames, we only use it to compute a prior on the orientation.

Based on the estimate $\check{q}_1^{\text{nb}}$ from (79), we can model the orientation at time $t = 1$ in terms of an orientation deviation

$$q_1^{\text{nb}} = \check{q}_1^{\text{nb}} \odot \exp_{\text{q}}\left(\frac{e_{\eta,i}}{2}\right), \quad e_{\eta,i} \sim \mathcal{N}(0, \Sigma_{\eta,i}), \quad (80a)$$

or in terms of a quaternion as

$$q_1^{\text{nb}} = \check{q}_1^{\text{nb}} + e_{\text{q},i}, \quad e_{\text{q},i} \sim \mathcal{N}(0, \Sigma_{\text{q},i}). \quad (80b)$$

Explicit formulations for the covariance of the orientation estimates from the TRIAD and QUEST algorithms are discussed by Shuster (2006). In practice, however, the accuracy of the estimates from (79) highly depends on the validity of the model assumptions, i.e. on whether the sensor is indeed far from magnetic material and whether the linear acceleration is indeed zero. Because this has such a significant influence on the quality of the estimates, we choose $\Sigma_{\eta,i}$ and $\Sigma_{\text{q},i}$ somewhat conservatively. Modeling that in 68% of the cases the orientation error is less than 20° ,

$$\Sigma_{\eta,i} = \sigma_{\eta,i}^2 \mathcal{I}_3, \quad \sigma_{\eta,i} = \frac{20}{180} \pi = 0.35, \quad (81a)$$

$$\Sigma_{\text{q},i} = \frac{1}{4} (\check{q}_1^{\text{nb}})^{\text{L}} \frac{\text{d exp}_{\text{q}}(e_{\eta,i})}{\text{d} e_{\eta,i}} \Sigma_{\eta,i} \left(\frac{\text{d exp}_{\text{q}}(e_{\eta,i})}{\text{d} e_{\eta,i}} \right)^{\text{T}} (\check{q}_1^{\text{bn}})^{\text{L}}, \quad (81b)$$

where we use the fact that $(q^{\text{L}})^{\text{T}} = (q^{\text{c}})^{\text{T}}$. Note that the covariance $\Sigma_{\text{q},i}$ is defined in terms of the covariance $\Sigma_{\eta,i}$ to allow for explicit comparison between different algorithms in Section 4.

3.7 Resulting probabilistic models

The information from the previous sections can now be combined into one model which will be used in the algorithms in subsequent sections. In this section, we

describe our modeling choices for the pose estimation problem and for the orientation estimation problem.

We assume that the sensor does not travel over significant distances as compared to the size of the earth and hence keep the navigation frame n fixed with respect to the earth frame e . Furthermore, we assume that the magnitude of the earth rotation and of the Coriolis acceleration are negligible. Our gyroscope and accelerometer models are hence given by

$$y_{a,t} = R_t^{bn}(a_{nn}^n - g^n) + \delta_{a,t}^b + e_{a,t}^b, \quad (82a)$$

$$y_{\omega,t} = \omega_{nb,t}^b + \delta_{\omega,t}^b + e_{\omega,t}^b. \quad (82b)$$

In the remainder, for notational convenience we drop the subscripts n which indicate in which frame the differentiation is performed, see Section 2.3, and use the shorthand notation a^n for a_{nn}^n . Furthermore, we will denote ω_{nb}^b simply by ω and omit the superscript b on the noise terms $e_{a,t}$ and $e_{\omega,t}$ and the bias terms $\delta_{a,t}$ and $\delta_{\omega,t}$. We assume that the inertial measurement noise is given by

$$e_{a,t} \sim \mathcal{N}(0, \sigma_a^2 \mathcal{I}_3), \quad e_{\omega,t} \sim \mathcal{N}(0, \sigma_\omega^2 \mathcal{I}_3), \quad (83)$$

i.e. we assume that the three sensor axes are independent and have the same noise levels.

Pose estimation

For pose estimation, we model the accelerometer and gyroscope measurements as inputs to the dynamics. Hence, the state vector consists of the position p_t^n , the velocity v_t^n and a parametrization of the orientation. We use the inertial measurements in combination with position measurements to estimate the pose.

Using the accelerometer measurement model (82a) in (68), the dynamics of the position and velocity is given by

$$p_{t+1}^n = p_t^n + T v_t^n + \frac{T^2}{2} (R_t^{nb} y_{a,t} + g^n - \delta_{a,t} + e_{a,t}), \quad (84a)$$

$$v_{t+1}^n = v_t^n + T (R_t^{nb} y_{a,t} + g^n - \delta_{a,t} + e_{a,t}), \quad (84b)$$

where without loss of generality, we switch the sign on the noise. Note that the noise term $e_{a,t}$ should be rotated to the navigation frame n by multiplying it with the rotation matrix R_t^{nb} . However, because of the assumption (83), the rotation matrix can be omitted without loss of generality. The dynamics of the orientation parametrized using quaternions is given by

$$q_{t+1}^{nb} = q_t^{nb} \odot \exp_q \left(\frac{T}{2} (y_{\omega,t} - \delta_{\omega,t} - e_{\omega,t}) \right). \quad (85)$$

Equivalent dynamic models can be obtained for the other parametrizations.

The position measurements are modeled as in (65). In summary, this leads to the following state space model for pose estimation

$$\begin{pmatrix} p_{t+1}^n \\ v_{t+1}^n \\ q_{t+1}^{\text{nb}} \end{pmatrix} = \begin{pmatrix} p_t^n + T v_t^n + \frac{T^2}{2} \left(R_t^{\text{nb}} y_{a,t} + g^n - \delta_{a,t} + e_{p,a,t} \right) \\ v_t^n + T \left(R_t^{\text{nb}} y_{a,t} + g^n - \delta_{a,t} + e_{v,a,t} \right) \\ q_t^{\text{nb}} \odot \exp_q \left(\frac{T}{2} (y_{\omega,t} - \delta_{\omega,t} - e_{\omega,t}) \right) \end{pmatrix}, \quad (86a)$$

$$y_{p,t} = p_t^n + e_{p,t}, \quad (86b)$$

where

$$e_{p,a,t} \sim \mathcal{N}(0, \Sigma_a), \quad e_{v,a,t} \sim \mathcal{N}(0, \Sigma_a), \quad (86c)$$

$$e_{p,t} \sim \mathcal{N}(0, \Sigma_p), \quad e_{\omega,t} \sim \mathcal{N}(0, \Sigma_\omega), \quad (86d)$$

with $\Sigma_a = \sigma_a^2 \mathcal{I}_3$ and $\Sigma_\omega = \sigma_\omega^2 \mathcal{I}_3$. Note that we model the process noise on the position and velocity states in terms of the accelerometer noise. However, we do not enforce these to have the same noise realization. Hence, we use the notation $e_{p,a,t}$ and $e_{v,a,t}$ for the two process noise terms. The covariance of both is equal to the covariance of the accelerometer noise. The initial position is assumed to be given by the first position measurement as

$$p_1^n = y_{p,1} + e_{p,1}, \quad e_{p,1} \sim \mathcal{N}(0, \Sigma_{p,1}), \quad (86e)$$

while the initial velocity is assumed to be approximately zero as

$$v_1^n = e_{v,1}, \quad e_{v,1} \sim \mathcal{N}(0, \Sigma_{v,1}). \quad (86f)$$

The orientation at time $t = 1$ is given by the QUEST algorithm described in Section 3.6, parametrized in terms of quaternions or rotation vectors as

$$q_1^{\text{nb}} = \check{q}_1^{\text{nb}} \odot \exp_q \left(\frac{e_{\eta,1}}{2} \right), \quad e_{\eta,1} \sim \mathcal{N}(0, \Sigma_{\eta,1}), \quad (86g)$$

$$q_1^{\text{nb}} = \check{q}_1^{\text{nb}} + e_{q,1}, \quad e_{q,1} \sim \mathcal{N}(0, \Sigma_{q,1}), \quad (86h)$$

where the initial orientation uncertainty is given in terms of a standard deviation of 20° .

In Section 4 we assume that the inertial measurement are properly calibrated. Hence, we assume that their biases $\delta_{a,t}^b$ and $\delta_{\omega,t}^b$ are zero. Calibration is the topic of Section 5 where we will also introduce possible extensions of the state space model in which the bias terms are included either as states or as unknown parameters.

Orientation estimation

For orientation estimation, the state vector only consists of a parametrization of the orientation. We use the inertial sensors in combination with the magnetometer measurements to estimate the orientation. The magnetometer measurements are modeled as in (64). Instead of using the accelerometer measurements model (82a),

we use the model (62) where it is assumed that the linear acceleration is approximately zero. This leads to the following state space model for orientation estimation,

$$q_{t+1}^{\text{nb}} = q_t^{\text{nb}} \odot \exp_{\mathbf{q}} \left(\frac{T}{2} (\gamma_{\omega,t} - \delta_{\omega} - e_{\omega,t}) \right), \quad (87a)$$

$$\gamma_{a,t} = -R_t^{\text{bn}} g^n + e_{a,t}, \quad (87b)$$

$$\gamma_{m,t} = R_t^{\text{bn}} m^n + e_{m,t}, \quad (87c)$$

where (87a) describes the dynamics while (87b) and (87c) describe the measurement models and

$$e_{\omega,t} \sim \mathcal{N}(0, \Sigma_{\omega}), \quad e_{a,t} \sim \mathcal{N}(0, \Sigma_a), \quad e_{m,t} \sim \mathcal{N}(0, \Sigma_m), \quad (87d)$$

with $\Sigma_{\omega} = \sigma_{\omega}^2 \mathcal{I}_3$ and $\Sigma_a = \sigma_a^2 \mathcal{I}_3$. The initial orientation is given by the QUEST algorithm described in Section 3.6 and is modeled as in (86g) or (86h). Also for orientation estimation, in Section 4 we assume that the inertial measurements are properly calibrated. Hence, we assume that the bias $\delta_{\omega,t}^b$ is zero.

4 Estimating position and orientation

In this section we will focus on position and orientation estimation using the models (86) and (87) derived in Section 3. In Section 4.1, we will first describe a method to solve the smoothing problem (15a). Subsequently, in Sections 4.2 and 4.3, we will derive different methods for solving the filtering problem (15b). In each section, after a general introduction of the estimation method, we will illustrate the method by explicitly deriving algorithms to estimate the orientation using the state space model (87). The orientation estimation problem also illustrates the most important parts of the pose estimation problem, since most complexities lie in the parametrization of the orientation and in the nonlinear nature of the orientation. In Section 4.4, we show some characteristics of the different algorithms for the orientation estimation problem. In Section 4.5, we will discuss how the algorithms for orientation estimation can be extended to also estimate the position. Throughout this section, we assume that the sensors are calibrated, i.e. we assume that we do not have any unknown parameters θ in our models. Because of this, the models that we use are the most basic models that can be used for position and orientation estimation using inertial sensors.

4.1 Smoothing in an optimization framework

Perhaps the most intuitive way to solve the smoothing problem is by posing it as an optimization problem, where a *maximum a posteriori* (MAP) estimate is

obtained as

$$\begin{aligned}\hat{x}_{1:N} &= \arg \max_{x_{1:N}} p(x_{1:N} \mid y_{1:N}) \\ &= \arg \max_{x_{1:N}} p(x_1) \prod_{t=2}^N p(x_t \mid x_{t-1}) p(y_t \mid x_t).\end{aligned}\quad (88)$$

Here, we use the notation in terms of probability distributions as introduced in Section 3.1 and model the measurements and the state dynamics as described in Sections 3.4 and 3.5, respectively. Furthermore, we assume that a prior on the initial state is obtained using the measurements at $t = 1$ as described in Section 3.6. Because of this, the measurement model $p(y_1 \mid x_1)$ from (15a) is explicitly omitted in (88). Note that in practice, we typically minimize $-\log p(x_{1:N} \mid y_{1:N})$ instead of maximizing $p(x_{1:N} \mid y_{1:N})$ itself, resulting in the optimization problem

$$\hat{x}_{1:N} = \arg \min_{x_{1:N}} -\log p(x_1) - \sum_{t=2}^N \log p(x_t \mid x_{t-1}) - \sum_{t=2}^N \log p(y_t \mid x_t). \quad (89)$$

There are various ways to solve problems of this kind, for instance particle smoothers (Lindsten and Schön, 2013), an extended Rauch-Tung-Striebel (RTS) smoother (Särkkä, 2013) and optimization methods, see e.g. Nocedal and Wright (2006); Mattingley and Boyd (2010). The latter approach is closely related to iterated Kalman smoothers (Bell, 1994; Jazwinski, 1970). We will solve the problem using an optimization method. Compared to extended RTS smoothers, optimization methods allow for more flexibility in the models that are being used. For instance, additional information outside of the standard state space model can straightforwardly be included. Optimization approaches are typically computationally heavier than extended RTS smoothers but less heavy than particle smoothers. The latter are capable of capturing the whole distribution, which is a clear advantage when the distributions are multi-modal. Optimization instead gives a point estimate and an associated measure of uncertainty. This is typically sufficient for the pose estimation problem using inertial sensors.

Gauss-Newton optimization

To obtain a smoothing estimate of the position and orientation using optimization, we first recognize that based on our models (86) and (87), all probability distributions in (89) are Gaussian. Let us therefore consider a slightly more general problem where the objective function consists of the product of the probability functions $p(e_i(x_{1:N}))$, $i = 1, \dots, M$. Hence, the optimization problem can be written as

$$\hat{x}_{1:N} = \arg \min_{x_{1:N}} - \sum_{i=1}^M \log p(e_i(x_{1:N})). \quad (90)$$

The probability distribution of $e_i(x)$ is given by

$$p(e_i(x_{1:N})) = \frac{1}{\sqrt{(2\pi)^{n_e} \det \Sigma_i}} \exp\left(-\frac{1}{2} e_i^\top(x_{1:N}) \Sigma_i^{-1} e_i(x_{1:N})\right). \quad (91)$$

Omitting the terms independent of $x_{1:N}$, the optimization problem (90) reduces to

$$\hat{x}_{1:N} = \arg \min_{x_{1:N}} \frac{1}{2} \sum_{i=1}^M \|e_i(x_{1:N})\|_{\Sigma_i^{-1}}^2, \quad (92)$$

with $\|e_i(x_{1:N})\|_{\Sigma_i^{-1}}^2 = e_i^\top(x_{1:N})\Sigma_i^{-1}e_i(x_{1:N})$. The function that is being minimized in optimization problems, is often referred to as the *objective function*.

The solution to (92) can be found by studying the shape of the objective function as a function of $x_{1:N}$. This can be characterized in terms of the *gradient* $\mathcal{G}(x_{1:N})$ and *Hessian* $\mathcal{H}(x_{1:N})$, which provide information about the slope and curvature of the function, respectively. Defining

$$e_i^\top(x_{1:N})\Sigma_i^{-1}e_i(x_{1:N}) = \varepsilon_i^\top \varepsilon_i, \quad \varepsilon_i = \Sigma_i^{-1/2}e_i(x_{1:N}),$$

and the stacked variables

$$\varepsilon = \begin{pmatrix} \varepsilon_1^\top & \cdots & \varepsilon_M^\top \end{pmatrix}^\top,$$

the gradient and the Hessian are given by

$$\mathcal{G}(x_{1:N}) = \sum_{i=1}^M \varepsilon_i^\top \frac{d\varepsilon_i}{dx_{1:N}} = \mathcal{J}(x_{1:N})\varepsilon, \quad (93a)$$

$$\begin{aligned} \mathcal{H}(x_{1:N}) &= \sum_{i=1}^M \left(\frac{d\varepsilon_i}{dx_{1:N}} \right)^\top \frac{d\varepsilon_i}{dx_{1:N}} + \varepsilon_i^\top \frac{d^2\varepsilon_i}{dx_{1:N}^2} \\ &= \mathcal{J}(x_{1:N})\mathcal{J}^\top(x_{1:N}) + \sum_{i=1}^M \varepsilon_i^\top \frac{d^2\varepsilon_i}{dx_{1:N}^2}. \end{aligned} \quad (93b)$$

Note that for notational convenience, we have omitted the explicit dependance of ε on $x_{1:N}$. In (93), we introduced the notation $\mathcal{J}(x_{1:N})$, which is the *Jacobian* of the vector ε with respect to $x_{1:N}$ as

$$\mathcal{J}(x_{1:N}) = \begin{pmatrix} \frac{d\varepsilon_1}{dx_1} & \cdots & \frac{d\varepsilon_M}{dx_1} \\ \vdots & & \vdots \\ \frac{d\varepsilon_1}{dx_N} & \cdots & \frac{d\varepsilon_M}{dx_N} \end{pmatrix}. \quad (94)$$

Instead of computing the true Hessian (93b), we compute an approximation of it (Nocedal and Wright, 2006), given by

$$\hat{\mathcal{H}}(x_{1:N}) = \mathcal{J}(x_{1:N})\mathcal{J}^\top(x_{1:N}). \quad (95)$$

This has the benefit of not having to compute second derivatives, at the same time as it guarantees that the Hessian is positive semidefinite. The downside of using (95) is that it introduces an approximation.

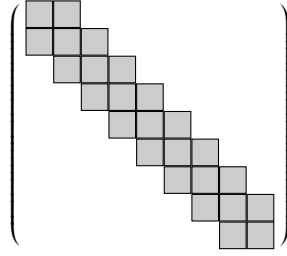


Figure 18: An illustration of the sparsity pattern that is present in the smoothing problem.

The gradient and the (approximate) Hessian can be used to find the minimum of the objective function. For our models (86) and (87), in which the functions $e_i(x_{1:N})$ are nonlinear, an estimate $\hat{x}_{1:N}$ can iteratively be computed as

$$\hat{x}_{1:N}^{(k+1)} = \hat{x}_{1:N}^{(k)} - \beta^{(k)} \left(\hat{\mathcal{H}}(\hat{x}_{1:N}^{(k)}) \right)^{-1} \mathcal{G}(\hat{x}_{1:N}^{(k)}), \quad (96)$$

where k denotes the iteration number. The *step length* $\beta^{(k)}$ is computed for instance using a backtracking line search (Nocedal and Wright, 2006; Boyd and Vandenberghe, 2004). The *search direction* is computed as $\left(\hat{\mathcal{H}}(\hat{x}_{1:N}^{(k)}) \right)^{-1} \mathcal{G}(\hat{x}_{1:N}^{(k)})$.

Note that an initial point $\hat{x}_{1:N}^{(0)}$ needs to be chosen close enough to the desired minimum to ensure convergence to this minimum.

In case the functions $e_i(x_{1:N})$ would be linear, the problem (92) would be a *least squares (LS)* problem for which the update (96) directly leads to the minimum of the objective function, irrespective of the initial point. In our case where the functions $e_i(x_{1:N})$ are nonlinear due to the nonlinear nature of the orientation, the problem (92) is instead a *nonlinear least squares (NLS)* problem. Each iteration in (96) can be interpreted as solving a LS problem around a linearization point. The linearization point is updated after each iteration, bringing it closer to the minimum of the objective function. Computing an estimate $\hat{x}_{1:N}$ by iterating (96) until convergence, making use of the approximate Hessian from (95), is called *Gauss-Newton optimization*.

Note that since the inertial sensors sample at high sampling rates, the length of the vector $x_{1:N}$ quickly becomes fairly large. For instance, for inertial sensors sampling at 100 Hz for 10 s, $N = 1\,000$ and the size of the (approximate) Hessian $\hat{\mathcal{H}}(x)$ is $1\,000n_x \times 1\,000n_x$, where n_x is the length of the vector x_t . However, as can be seen in (89), the components of the objective function only depend on the current and next time steps x_t and x_{t+1} . Hence, the structure of the Hessian (95) is of the form given in Figure 18. There exist efficient algorithms to compute search directions for problems with this sparsity pattern, which can be exploited using sparse matrix packages, see e.g. Davis (2006), or by using tools like dynamic

programming and message passing (Bertsekas, 1995; Golub and van Loan, 2013; Saad, 2003).

Smoothing estimates of the orientation using optimization

In this section, we will illustrate the use of Gauss-Newton optimization to obtain smoothing estimates of the orientation. As discussed in the previous section, the crucial part is to identify the objective function and its Jacobian. From this, the gradient and approximate Hessian can be computed using (93a) and (95), which can be used to iteratively update the estimates using (96).

Combining the general formulation of the smoothing problem (89) and using the model for orientation estimation (87), the orientation smoothing problem is given by

$$\hat{x}_{1:N} = \arg \min_{x_{1:N}} \underbrace{\|e_{\eta,i}\|_{\Sigma_{\eta,i}^{-1}}^2}_{\text{Prior}} + \underbrace{\sum_{t=2}^N \|e_{\omega,t}\|_{\Sigma_{\omega}^{-1}}^2}_{\text{Dynamics}} + \underbrace{\sum_{t=2}^N \left(\|e_{a,t}\|_{\Sigma_a^{-1}}^2 + \|e_{m,t}\|_{\Sigma_m^{-1}}^2 \right)}_{\text{Measurement models}}, \quad (97)$$

with

$$e_{\eta,i} = 2 \log_q \left(\tilde{q}_1^{\text{bn}} \odot q_1^{\text{nb}} \right), \quad e_{\eta,i} \sim \mathcal{N}(0, \Sigma_{\eta,i}), \quad (98a)$$

$$e_{\omega,t} = \frac{2}{T} \log_q \left(q_t^{\text{bn}} \odot q_{t+1}^{\text{nb}} \right) - y_{\omega,t}, \quad e_{\omega,t} \sim \mathcal{N}(0, \Sigma_{\omega}), \quad (98b)$$

$$e_{a,t} = y_{a,t} + R_t^{\text{bn}} g^n, \quad e_{a,t} \sim \mathcal{N}(0, \Sigma_a), \quad (98c)$$

$$e_{m,t} = y_{m,t} - R_t^{\text{bn}} m^n, \quad e_{m,t} \sim \mathcal{N}(0, \Sigma_m). \quad (98d)$$

Convex equality constraints can straightforwardly be incorporated in optimization problems. However, using N norm equality constraints to preserve the unit norm of the quaternions, severely complicates the optimization problem since these norm constraints are non-convex. Instead, we encode an orientation in terms of a linearization point parametrized as a unit quaternion $\tilde{q}_t^{\text{nb}}$ and an orientation deviation parametrized as a rotation vector η_t^{b} as discussed in Section 3.3. Hence, we model the orientation as

$$q_t^{\text{nb}} = \tilde{q}_t^{\text{nb}} \odot \exp_q \left(\frac{\eta_t^{\text{b}}}{2} \right). \quad (99)$$

At each Gauss-Newton iteration (96), we estimate the state vector $\eta_{1:N}^{\text{b}}$. Before starting the next iteration, the linearization points $\tilde{q}_{1:N}^{\text{nb}}$ are updated and the state vector $\eta_{1:N}^{\text{b}}$ is reset to zero.

Using the notation introduced in Section 3.3, the objective function (97) can be expressed in terms of the orientation deviation η_t^b by rewriting (98) as

$$e_{\eta,i} = 2 \log_q \left(\tilde{q}_1^{\text{bn}} \odot \tilde{q}_1^{\text{nb}} \odot \exp_q \left(\frac{\eta_1^b}{2} \right) \right), \quad (100a)$$

$$e_{\omega,t} = \frac{2}{T} \log_q \left((\exp_q \left(\frac{\eta_t^b}{2} \right))^c \odot \tilde{q}_t^{\text{bn}} \odot \tilde{q}_{t+1}^{\text{nb}} \odot \exp_q \left(\frac{\eta_{t+1}^b}{2} \right) \right) - y_{\omega,t}, \quad (100b)$$

$$e_{a,t} = y_{a,t} + \left(\exp_R(\eta_t^b) \right)^\top \tilde{R}_t^{\text{bn}} g^n, \quad (100c)$$

$$e_{m,t} = y_{m,t} - \left(\exp_R(\eta_t^b) \right)^\top \tilde{R}_t^{\text{bn}} m^n. \quad (100d)$$

Here, $\tilde{R}_t^{\text{bn}}$ is the rotation matrix representation of the linearization point $\tilde{q}_t^{\text{bn}}$. This leads to the following derivatives

$$\frac{de_{\eta,i}}{d\eta_1^b} = \frac{d \log_q(q)}{dq} \left(\tilde{q}_1^{\text{bn}} \odot \tilde{q}_1^{\text{nb}} \right)^L \frac{d \exp_q(\eta_1^b)}{d\eta_1^b}, \quad (101a)$$

$$\frac{de_{\omega,t}}{d\eta_{t+1}^b} = \frac{1}{T} \frac{d \log_q(q)}{dq} \left(\tilde{q}_t^{\text{bn}} \odot \tilde{q}_{t+1}^{\text{nb}} \right)^L \frac{d \exp_q(\eta_{t+1}^b)}{d\eta_{t+1}^b}, \quad (101b)$$

$$\frac{de_{\omega,t}}{d\eta_t^b} = \frac{1}{T} \frac{d \log_q(q)}{dq} \left(\tilde{q}_t^{\text{bn}} \odot \tilde{q}_{t+1}^{\text{nb}} \right)^R \frac{d(\exp_q(\eta_t^b))^c}{d \exp_q(\eta_t^b)} \frac{d \exp_q(\eta_t^b)}{d\eta_t^b}, \quad (101c)$$

$$\frac{de_{a,t}}{d\eta_t^b} \approx [\tilde{R}_t^{\text{bn}} g^n \times], \quad (101d)$$

$$\frac{de_{m,t}}{d\eta_t^b} \approx -[\tilde{R}_t^{\text{bn}} m^n \times], \quad (101e)$$

where, using (52) and the definition of the quaternion conjugate (37),

$$\frac{d \log_q(q)}{dq} \approx \begin{pmatrix} 0_{1 \times 3} \\ \mathcal{I}_3 \end{pmatrix}^\top, \quad \frac{d(\exp_q(\eta))^c}{d \exp \eta} = \begin{pmatrix} 1 & 0_{1 \times 3} \\ 0_{3 \times 1} & -\mathcal{I}_3 \end{pmatrix}, \quad \frac{d \exp_q \eta}{d \eta} \approx \begin{pmatrix} 0_{1 \times 3} \\ \mathcal{I}_3 \end{pmatrix}. \quad (102)$$

Using the approximate derivatives (101), the gradient and approximate Hessian can be computed for the Gauss-Newton iterations. The resulting solution is summarized in Algorithm 1. One of the inputs to the algorithm is an initial estimate of the orientation $\tilde{q}_{1:N}^{\text{nb},(0)}$, which will aid the convergence to the desired minimum.

There are (at least) two ways to obtain good initial estimates $\tilde{q}_{1:N}^{\text{nb},(0)}$. First, they can be obtained by direct integration of the gyroscope measurements. As discussed in Section 1.2, these estimates will suffer from integration drift. However, they still form a good initial estimate for the optimization problem. Second, one of the other, computationally cheaper estimation algorithms that will be discussed in the remainder of this section can be used to obtain initial estimates of the orientation.

Computing the uncertainty

As discussed in Section 3.1, we are not only interested in obtaining point estimates of the position and orientation, but also in estimating their uncertainty. In our case of Gaussian noise, this is characterized by the covariance. As shown in e.g.

Algorithm 1 Smoothing estimates of the orientation using optimization

INPUTS: An initial estimate of the orientation $\hat{q}_{1:N}^{\text{nb},(0)}$, inertial data $\{y_{a,t}, y_{\omega,t}\}_{t=1}^N$, magnetometer data $\{y_{m,t}\}_{t=1}^N$ and covariance matrices Σ_{ω} , Σ_a and Σ_m .

OUTPUTS: An estimate of the orientation $\hat{q}_{1:N}^{\text{nb}}$ and optionally its covariance $\text{cov}(\hat{\eta}_{1:N}^{\text{b}})$.

1. Set $\hat{\eta}_t^{\text{b},(0)} = 0_{3 \times 1}$ for $t = 1, \dots, N$, set $k = 0$ and compute $\hat{q}_1^{\text{nb}}$ and $\Sigma_{\eta,i}$ as described in Section 3.6.
2. **while** *termination condition is not satisfied* **do**
 - (a) Compute the gradient (93a) and the approximate Hessian (95) of the orientation smoothing problem (97) using the expressions for the different parts of the cost function and their Jacobians (100) and (101).
 - (b) Apply the update (96) to obtain $\hat{\eta}_{1:N}^{\text{b},(k+1)}$.
 - (c) Update the linearization point as

$$\hat{q}_t^{\text{nb},(k+1)} = \hat{q}_t^{\text{nb},(k)} \odot \exp_{\text{q}}\left(\frac{\hat{\eta}_t^{\text{b},(k+1)}}{2}\right), \quad (103)$$

and set $\hat{\eta}_t^{\text{b},(k+1)} = 0_{3 \times 1}$ for $t = 1, \dots, N$.

- (d) Set $k = k + 1$.

end while

3. Set $\hat{q}_{1:N}^{\text{nb}} = \hat{q}_{1:N}^{\text{nb},(k)}$.
4. Optionally compute

$$\text{cov}(\hat{\eta}_{1:N}^{\text{b}}) = \left(\mathcal{J}(\hat{\eta}_{1:N}^{\text{b}}) \mathcal{J}^{\text{T}}(\hat{\eta}_{1:N}^{\text{b}}) \right)^{-1}. \quad (104)$$

Ljung (1999); Verhaegen and Verdult (2007), if our position and orientation estimation problems would be LS problems, the covariance of the estimates would be given by the inverse of the Hessian of the objective function (93b). Instead, we solve an NLS problem, for which a number of LS problems are solved around linearization points closer and closer to the minimum of the objective function. Hence, when the algorithm has converged, the problem can locally well be described by the quadratic approximation around its resulting linearization point. We can therefore approximate the covariance of our estimates as

$$\text{cov}(\hat{x}_{1:N}) = \left(\mathcal{J}(\hat{x}_{1:N}) \mathcal{J}^{\text{T}}(\hat{x}_{1:N}) \right)^{-1}. \quad (105)$$

An intuition behind this expression is that the accuracy of the estimates is related to the sensitivity of the objective function with respect to the states. The covariance of the estimate will play a crucial role in the filtering approaches discussed in Sections 4.2 and 4.3.

The matrix $\mathcal{J}(\hat{x}_{1:N})\mathcal{J}^\top(\hat{x}_{1:N})$ quickly becomes fairly large due to the high sampling rates of the inertial sensors. Hence, computing its inverse can be computationally costly. We are, however, typically only interested in a subset of the inverse. For instance, we are often only interested in diagonal or block diagonal elements representing $\text{cov}(x_t)$. It is therefore not necessary to explicitly form the complete inverse, which makes the computation tractable also for larger problem sizes.

4.2 Filtering estimation in an optimization framework

One of the benefits of using a smoothing formulation is that all measurements $y_{1:N}$ are used to get the best estimates of the states $x_{1:N}$. However, both the computational cost and the memory requirements grow with the length of the data set. Furthermore, it is a post-processing solution in which we have to wait until all data is available. Alternatively, the filtering problem can be formulated as

$$\begin{aligned}\hat{x}_{t+1} &= \arg \min_{x_{t+1}} -\log p(x_{t+1} | y_{1:t+1}) \\ &= \arg \min_{x_{t+1}} -\log p(y_{t+1} | x_{t+1}) - \log p(x_{t+1} | y_{1:t}).\end{aligned}\quad (106)$$

Note that without loss of generality, we have shifted our time indices as compared to the notation in Section 3.1. The prior $p(x_{t+1} | y_{1:t})$ is obtained by marginalizing out the previous state x_t as

$$p(x_{t+1} | y_{1:t}) = \int p(x_{t+1}, x_t | y_{1:t}) dx_t = \int p(x_{t+1} | x_t) p(x_t | y_{1:t}) dx_t. \quad (107)$$

In this section we will derive an algorithm to obtain filtering estimates of the orientation using optimization. Similar to Algorithm 1, we will iteratively compute estimates using Gauss-Newton optimization. However, instead of optimizing over the whole data set at once, we run an optimization algorithm for each time instance t .

Let us assume that the probability distributions are Gaussian and given by

$$p(x_{t+1} | x_t) \sim \mathcal{N}(x_{t+1}; f(x_t), Q), \quad p(x_t | y_{1:t}) \sim \mathcal{N}(x_t; \hat{x}_t, P_{t|t}). \quad (108)$$

The integral in (107) can then be approximated according to

$$p(x_{t+1} | y_{1:t}) \approx \mathcal{N}(x_{t+1}; f(\hat{x}_t), F_t P_{t|t} F_t^\top + G_t Q G_t^\top), \quad (109)$$

with $F_t = \frac{\partial f(x_t)}{\partial x_t}$ and $G_t = \frac{\partial f(x_t)}{\partial w_t}$, where w_t is the process noise defined in (17).

For the orientation estimation problem, using (99) in combination with (87a), we can express η_{t+1}^b in terms of η_t^b as

$$\begin{aligned}\eta_{t+1}^b &= f(\eta_t^b, y_{\omega,t}, e_{\omega,t}) \\ &= 2 \log_q \left(\hat{q}_{t+1}^{\text{bn}} \odot \hat{q}_t^{\text{nb}} \odot \exp_q \left(\frac{\eta_t^b}{2} \right) \odot \exp_q \left(\frac{T}{2} (y_{\omega,t} + e_{\omega,t}) \right) \right).\end{aligned}\quad (110)$$

To simplify notation, we choose the linearization point $\hat{q}_{t+1}^{\text{nb},(0)}$ at iteration $k = 0$ as

$$\hat{q}_{t+1}^{\text{nb},(0)} = \hat{q}_t^{\text{nb}} \odot \exp_{\mathbf{q}}\left(\frac{T}{2}y_{\omega,t}\right), \quad (111)$$

where $\hat{q}_t^{\text{nb}}$ is the orientation estimate from the previous time step. Around this updated linearization point, the distribution $p(\eta_{t+1}^{\text{b}} | y_{1:t})$ can be written as

$$p(\eta_{t+1}^{\text{b}} | y_{1:t}) \approx \mathcal{N}\left(\eta_{t+1}^{\text{b}}; 0, F_t P_{t|t} F_t^{\top} + G_t Q G_t^{\top}\right), \quad (112)$$

with $Q = \Sigma_{\omega}$ and

$$\begin{aligned} F_t &= \left. \frac{\partial f(\eta_t^{\text{b}}, y_{\omega,t}, e_{\omega,t})}{\partial \eta_t^{\text{b}}} \right|_{\substack{e_{\omega,t}=0 \\ \eta_t^{\text{b}}=0}} \\ &= 2 \left. \frac{\partial}{\partial \eta_t^{\text{b}}} \log_{\mathbf{q}} \left(\exp_{\mathbf{q}}\left(-\frac{T}{2}y_{\omega,t}\right) \odot \exp_{\mathbf{q}}\left(\frac{\eta_t^{\text{b}}}{2}\right) \odot \exp_{\mathbf{q}}\left(\frac{T}{2}y_{\omega,t}\right) \right) \right|_{\eta_t^{\text{b}}=0} \\ &= \exp_{\mathbf{R}}(-T y_{\omega,t}), \end{aligned} \quad (113a)$$

$$\begin{aligned} G_t &= \left. \frac{\partial f(\eta_t^{\text{b}}, y_{\omega,t}, e_{\omega,t})}{\partial e_{\omega,t}} \right|_{\substack{e_{\omega,t}=0 \\ \eta_t^{\text{b}}=0}} \\ &= 2 \left. \frac{\partial}{\partial e_{\omega,t}} \log_{\mathbf{q}} \left(\exp_{\mathbf{q}}\left(-\frac{T}{2}y_{\omega,t}\right) \odot \exp_{\mathbf{q}}\left(\frac{T}{2}(y_{\omega,t} + e_{\omega,t})\right) \right) \right|_{e_{\omega,t}=0} \\ &\approx T \mathcal{I}_3. \end{aligned} \quad (113b)$$

Note that in (113) we assume that $\eta_t^{\text{b}} = 0$ since similar to the approach in Algorithm 1, the linearization point is updated after each iteration of the optimization algorithm.

The covariance $P_{t|t}$ can be approximated as the inverse of the Hessian of the objective function from the previous time step, see also Section 4.1. We make use of the shorthand notation $P_{t+1|t}^{(0)} = F_t P_{t|t} F_t^{\top} + G_t Q G_t^{\top}$ and define

$$e_{f,t} = -2 \log_{\mathbf{q}} \left(\hat{q}_t^{\text{bn}} \odot \hat{q}_{t-1}^{\text{nb}} \odot \exp_{\mathbf{q}}\left(\frac{T}{2}y_{\omega,t-1}\right) \right), \quad \frac{de_{f,t}}{d\eta_t^{\text{b}}} = \mathcal{I}_3. \quad (114)$$

Note that $e_{f,t}$ is equal to zero for iteration $k = 0$ but can be non-zero for subsequent iterations. Using this notation, the filtering problem (106) results in the following optimization problem

$$\begin{aligned} \hat{x}_t &= \arg \min_{x_t} -\log p(x_t | y_{1:t}) \\ &= \arg \min_{x_t} \underbrace{\|e_{f,t}\|_{P_{t|t-1}^{-1}}}_{\text{Dynamics and knowledge about } x_{t-1}} + \underbrace{\|e_{a,t}\|_{\Sigma_a^{-1}} + \|e_{m,t}\|_{\Sigma_m^{-1}}}_{\text{Measurement models}}. \end{aligned} \quad (115)$$

Note the similarity of this optimization problem to the smoothing formulation in (97). The term $\|e_{f,t}\|_{P_{t|t-1}^{-1}}$ takes into account both the knowledge about the previous state x_t and the dynamics. Furthermore, due to the fact that $P_{t|t-1}^{-1}$ is time-varying, the uncertainty and cross-correlation of the states at the previous time

instance is taken into consideration. Including this term is similar to the inclusion of an arrival cost in moving horizon estimation approaches (Rao et al., 2003).

After each Gauss-Newton iteration, we need to recompute the linearization point and compute the covariance around this updated linearization point as

$$\hat{q}_t^{\text{nb},(k+1)} = \hat{q}_t^{\text{nb},(k)} \odot \exp_{\mathbf{q}}\left(\frac{\hat{\eta}_t^{\text{b},(k+1)}}{2}\right), \quad P_{t|t-1}^{(k+1)} = J_t^{(k)} P_{t|t-1}^{(k)} \left(J_t^{(k)}\right)^\top, \quad (116)$$

where $J_t^{(k)} = \exp_{\mathbf{R}}(-\eta_t^{\text{b},(k+1)})$, which can be derived similarly to the derivation of F_t in (113a). The process of estimating orientation using filtering is summarized in Algorithm 2.

4.3 Extended Kalman filtering

Instead of using optimization for position and orientation estimation, it is also possible to use extended Kalman filtering. Extended Kalman filters (EKF) compute filtering estimates in terms of the conditional probability distribution (14). Hence, the approach is similar to the one discussed in Section 4.2. In fact, extended Kalman filtering can be interpreted as Gauss-Newton optimization of the filtering problem using only one iteration (96) with a step length of one, see e.g. Skoglund et al. (2015). In this section we first introduce how an EKF can be used to compute state estimates in a general nonlinear state space model. Subsequently, we illustrate the use of EKFs for position and orientation estimation by focusing on the orientation estimation problem. Two different implementations will be discussed. First, we introduce an EKF implementation that uses unit quaternions as states. Subsequently, we discuss an EKF which parametrizes the orientation in terms of an orientation deviation from a linearization point, similar to the approach used in Sections 4.1 and 4.2.

An EKF makes use of a nonlinear state space model. Assuming that the measurement noise is additive and that both the process and the measurement noise are zero-mean Gaussian with constant covariance, the state space model is given by

$$x_{t+1} = f_t(x_t, u_t, w_t), \quad (120a)$$

$$y_t = h_t(x_t) + e_t, \quad (120b)$$

with $w_t \sim \mathcal{N}(0, Q)$ and $e_t \sim \mathcal{N}(0, R)$.

The state is estimated recursively by performing a *time update* and a *measurement update*. The time update uses the model (120a) to “predict” the state to the next time step according to

$$\hat{x}_{t+1|t} = f_t(\hat{x}_{t|t}, u_t), \quad (121a)$$

$$P_{t+1|t} = F_t P_{t|t} F_t^\top + G_t Q G_t^\top, \quad (121b)$$

with

$$F_t = \left. \frac{\partial f_t(x_t, u_t, w_t)}{\partial x_t} \right|_{\substack{w_t=0 \\ x_t=\hat{x}_{t|t}}}, \quad G_t = \left. \frac{\partial f_t(x_t, u_t, w_t)}{\partial w_t} \right|_{\substack{w_t=0 \\ x_t=\hat{x}_{t|t}}}. \quad (122)$$

Algorithm 2 Filtering estimates of the orientation using optimization

INPUTS: Inertial data $\{y_{a,t}, y_{\omega,t}\}_{t=1}^N$, magnetometer data $\{y_{m,t}\}_{t=1}^N$ and covariance matrices Σ_{ω} , Σ_a and Σ_m .

OUTPUTS: An estimate of the orientation $\hat{q}_t^{\text{nb}}$ and its covariance $P_{t|t}$ for $t = 1, \dots, N$

1. Compute $\hat{q}_1^{\text{nb}}$ and Σ_i as described in Section 3.6 and set $\hat{q}_1^{\text{nb}} = \hat{q}_1^{\text{nb}}$ and $P_{1|1} = \Sigma_{\eta,i}$.

2. **for** $t = 2, \dots, N$ **do**

(a) Set $\hat{q}_t^{\text{b},(0)} = 0_{3 \times 1}$, set $k = 0$, choose the linearization point $\hat{q}_t^{\text{nb},(0)}$ as

$$\hat{q}_t^{\text{nb},(0)} = \hat{q}_{t-1}^{\text{nb}} \odot \exp_{\mathbf{q}}\left(\frac{T}{2} y_{\omega,t-1}\right), \quad (117a)$$

and compute $P_{t|t-1}^{(0)}$ as

$$P_{t|t-1}^{(0)} = F_t P_{t-1|t-1} F_t^{\top} + G_{t-1} \Sigma_{\omega} G_{t-1}^{\top}, \quad (117b)$$

with $F_{t-1} = \exp_{\mathbf{R}}(-T y_{\omega,t-1})$ and $G_t = T \mathcal{I}_3$.

(b) **while** *termination condition is not satisfied* **do**

i. Compute the gradient (93a) and the approximate Hessian (95) of the filtering problem (115) using the expressions for the different parts of the cost function and their Jacobians (114), (100c), (100d), (101d) and (101e).

ii. Apply the update (96) to obtain $\hat{q}_t^{\text{b},(k+1)}$.

iii. Update the linearization point and the covariance $P_{t|t-1}$ as

$$\hat{q}_t^{\text{nb},(k+1)} = \hat{q}_t^{\text{nb},(k)} \odot \exp_{\mathbf{q}}\left(\frac{\hat{q}_t^{\text{b},(k+1)}}{2}\right), \quad (118a)$$

$$P_{t|t-1}^{(k+1)} = J_{t-1}^{(k)} P_{t|t-1}^{(k)} (J_{t-1}^{(k)})^{\top}, \quad (118b)$$

with $J_t^{(k)} = \exp_{\mathbf{R}}\left(-\eta_t^{\text{b},(k+1)}\right)$, and set $\hat{q}_t^{\text{b},(k+1)} = 0_{3 \times 1}$.

iv. Set $k = k + 1$.

end while

(c) Set $\hat{q}_t^{\text{nb}} = \hat{q}_t^{\text{nb},(k)}$ and compute $P_{t|t}$ as

$$P_{t|t} = \left(\mathcal{J}(\hat{q}_t^{\text{b}}) \mathcal{J}^{\top}(\hat{q}_t^{\text{b}})\right)^{-1}. \quad (119)$$

Here, the matrix P denotes the state covariance. The double subscripts on $\hat{x}_{t+1|t}$ and $P_{t+1|t}$ denote the state estimate and the state covariance at time $t + 1$ given measurements up to time t . Similarly, $\hat{x}_{t|t}$ and $P_{t|t}$ denote the state estimate and the state covariance at time t given measurements up to time t .

The measurement update makes use of the measurement model (120b) in combination with the measurements y_t to update the “predicted” state estimate as

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t \varepsilon_t, \quad (123a)$$

$$P_{t|t} = P_{t|t-1} - K_t S_t K_t^T, \quad (123b)$$

with

$$\varepsilon_t \triangleq y_t - \hat{y}_{t|t-1}, \quad S_t \triangleq H_t P_{t|t-1} H_t^T + R, \quad K_t \triangleq P_{t|t-1} H_t^T S_t^{-1}, \quad (124)$$

and

$$\hat{y}_{t|t-1} = h(\hat{x}_{t|t-1}), \quad H_t = \left. \frac{\partial h_t(x_t)}{\partial x_t} \right|_{x_t = \hat{x}_{t|t-1}}. \quad (125)$$

Note that in (123) we have shifted our notation one time step compared to the notation in (121) to avoid cluttering the notation. The EKF iteratively performs a time update and a measurement update to estimate the state and its covariance.

Estimating orientation using quaternions as states

In this section, we will illustrate the use of an EKF to compute filtering estimates of the orientation. As discussed in the previous section, the crucial part is to compute the matrices F_t , G_t and H_t to perform the EKF time and measurement updates. Using the state space model (87) and using unit quaternions as states in the EKF, the dynamic model is given by

$$\begin{aligned} q_{t+1}^{\text{nb}} &= f_t(q_t^{\text{nb}}, y_{\omega,t}, e_{\omega,t}) = q_t^{\text{nb}} \odot \exp_q \left(\frac{T}{2} (y_{\omega,t} - e_{\omega,t}) \right) \\ &= \left(\exp_q \left(\frac{T}{2} (y_{\omega,t} - e_{\omega,t}) \right) \right)^R q_t^{\text{nb}} = \left(q_t^{\text{nb}} \right)^L \exp_q \left(\frac{T}{2} (y_{\omega,t} - e_{\omega,t}) \right). \end{aligned} \quad (126)$$

The following derivatives of the dynamic model (126) can be obtained

$$F_t = \left. \frac{\partial f_t(q_t^{\text{nb}}, y_{\omega,t}, e_{\omega,t})}{\partial q_t^{\text{nb}}} \right|_{\substack{e_{\omega,t}=0 \\ q_t^{\text{nb}} = \hat{q}_{t|t}^{\text{nb}}}} = \left(\exp_q \left(\frac{T}{2} y_{\omega,t} \right) \right)^R, \quad (127a)$$

$$\begin{aligned} G_t &= \left. \frac{\partial f_t(q_t^{\text{nb}}, y_{\omega,t}, e_{\omega,t})}{\partial e_{\omega,t}} \right|_{\substack{e_{\omega,t}=0 \\ q_t^{\text{nb}} = \hat{q}_{t|t}^{\text{nb}}}} \\ &= \left. \frac{\partial}{\partial e_{\omega,t}} \left(q_t^{\text{nb}} \right)^L \exp_q \left(\frac{T}{2} (y_{\omega,t} - e_{\omega,t}) \right) \right|_{\substack{e_{\omega,t}=0 \\ q_t^{\text{nb}} = \hat{q}_{t|t}^{\text{nb}}}} \\ &= -\frac{T}{2} \left(\hat{q}_{t|t}^{\text{nb}} \right)^L \frac{d \exp_q(e_{\omega,t})}{d e_{\omega,t}}. \end{aligned} \quad (127b)$$

In the measurement update of the EKF, the state is updated using the accelerometer and magnetometer measurements. Using the measurement models

$$y_{a,t} = -R_t^{bn} g^n + e_{a,t}, \quad y_{m,t} = R_t^{bn} m^n + e_{m,t}, \quad (128)$$

the matrix H_t in (125) is given by

$$H_t = \frac{\partial}{\partial q_t^{nb}} \left(R_t^{bn} g^n \right) \bigg|_{q_t^{nb} = \hat{q}_{t|t-1}^{nb}} = \begin{pmatrix} -\frac{\partial R_{t|t-1}^{bn}}{\partial q_{t|t-1}^{nb}} \big|_{q_{t|t-1}^{nb} = \hat{q}_{t|t-1}^{nb}} g^n \\ \frac{\partial R_{t|t-1}^{bn}}{\partial q_{t|t-1}^{nb}} \big|_{q_{t|t-1}^{nb} = \hat{q}_{t|t-1}^{nb}} m^n \end{pmatrix}. \quad (129)$$

The derivative can be computed from the definition of the relation between and the rotation matrix and the quaternion representation given in (170).

Note that the quaternion obtained from the measurement update (123) is no longer normalized. We denote this unnormalized quaternion and its covariance by $\tilde{q}_{t|t}^{nb}$ and $\tilde{P}_{t|t}$, respectively. The $\tilde{\cdot}$ instead of the $\hat{\cdot}$ is meant to explicitly indicate that the quaternion still needs to be updated. This is done by an additional renormalization step as compared to a standard EKF implementation. A possible interpretation of the renormalization is as an additional measurement update without measurement noise. Hence, we adjust both the quaternion and its covariance as

$$\hat{q}_{t|t}^{nb} = \frac{\tilde{q}_{t|t}^{nb}}{\|\tilde{q}_{t|t}^{nb}\|_2}, \quad P_{t|t} = J_t \tilde{P}_{t|t} J_t^\top, \quad (130a)$$

with

$$J_t = \frac{1}{\|\tilde{q}_{t|t}^{nb}\|_2^3} \tilde{q}_{t|t}^{nb} \left(\tilde{q}_{t|t}^{nb} \right)^\top. \quad (130b)$$

Here, $q^\top$ is the standard vector transpose. The resulting EKF is summarized in Algorithm 3.

Estimating orientation using orientation deviations as states

An alternative EKF implementation parametrizes the orientation in terms of an orientation deviation around a linearization point. The linearization point is parametrized in terms of quaternions or rotation matrices and denoted $\tilde{q}_t^{nb}$ or equivalently $\tilde{R}_t^{bn}$. The orientation deviation η_t^b is the state vector in the EKF. This EKF implementation is sometimes referred to as a multiplicative EKF (Crassidis et al., 2007; Markley, 2003). One of its advantages is that its implementation is computationally attractive since it only uses a 3-dimensional state compared to the 4-dimensional state in Algorithm 3.

In the time update, we use the dynamic model to directly update the linearization point as

$$\tilde{q}_{t+1}^{nb} = \tilde{q}_t^{nb} \odot \exp_q \left(\frac{T}{2} y_{\omega,t} \right). \quad (134)$$

Algorithm 3 Orientation estimation using an EKF with quaternion states

INPUTS: Inertial data $\{y_{a,t}, y_{\omega,t}\}_{t=1}^N$, magnetometer data $\{y_{m,t}\}_{t=1}^N$ and covariance matrices Σ_{ω} , Σ_a and Σ_m .

OUTPUTS: An estimate of the orientation $\hat{q}_{t|t}^{\text{nb}}$ and its covariance $P_{t|t}$ for $t = 1, \dots, N$.

1. Compute $\check{q}_1^{\text{nb}}$ and Σ_i as described in Section 3.6 and set $\hat{q}_{1|1}^{\text{nb}} = \check{q}_1^{\text{nb}}$ and $P_{1|1} = \Sigma_{q,i}$.

2. **For** $t = 2, \dots, N$ **do**

(a) Time update

$$\hat{q}_{t|t-1}^{\text{nb}} = \hat{q}_{t-1|t-1}^{\text{nb}} \odot \exp_q\left(\frac{T}{2} y_{\omega,t-1}\right), \quad (131a)$$

$$P_{t|t-1} = F_{t-1} P_{t-1|t-1} F_{t-1}^T + G_{t-1} Q G_{t-1}^T, \quad (131b)$$

with $F_{t-1} = \left(\exp_q\left(\frac{T}{2} y_{\omega,t-1}\right)\right)^R$, $G_t = -\frac{T}{2} \left(\hat{q}_{t-1|t-1}^{\text{nb}}\right)^L \frac{d \exp_q(e_{\omega,t-1})}{d e_{\omega,t-1}}$ and $Q = \Sigma_{\omega}$.

(b) Measurement update

$$\tilde{q}_{t|t}^{\text{nb}} = \hat{q}_{t|t-1}^{\text{nb}} + K_t \varepsilon_t, \quad (132a)$$

$$\tilde{P}_{t|t} = P_{t|t-1} - K_t S_t K_t^T, \quad (132b)$$

with ε_t , K_t and S_t defined in (124) and

$$y_t = \begin{pmatrix} y_{a,t} \\ y_{m,t} \end{pmatrix}, \quad \hat{y}_{t|t-1} = \begin{pmatrix} -\hat{R}_{t|t-1}^{\text{bn}} g^n \\ \hat{R}_{t|t-1}^{\text{bn}} m^n \end{pmatrix},$$

$$H_t = \begin{pmatrix} -\frac{\partial R_{t|t-1}^{\text{bn}}}{\partial \hat{q}_{t|t-1}^{\text{nb}}} \Big|_{q_{t|t-1}^{\text{nb}} = \hat{q}_{t|t-1}^{\text{nb}}} g^n \\ \frac{\partial R_{t|t-1}^{\text{bn}}}{\partial \hat{q}_{t|t-1}^{\text{nb}}} \Big|_{q_{t|t-1}^{\text{nb}} = \hat{q}_{t|t-1}^{\text{nb}}} m^n \end{pmatrix}, \quad R = \begin{pmatrix} \Sigma_a & 0 \\ 0 & \Sigma_m \end{pmatrix}.$$

(c) Renormalize the quaternion and its covariance as

$$\hat{q}_{t|t}^{\text{nb}} = \frac{\tilde{q}_{t|t}^{\text{nb}}}{\|\tilde{q}_{t|t}^{\text{nb}}\|_2}, \quad P_{t|t} = J_t \tilde{P}_{t|t} J_t^T, \quad (133)$$

with $J_t = \frac{1}{\|\tilde{q}_{t|t}^{\text{nb}}\|_2^3} \tilde{q}_{t|t}^{\text{nb}} \left(\tilde{q}_{t|t}^{\text{nb}}\right)^T$.

end for

The dynamic equation is therefore given by

$$\begin{aligned}
 \eta_{t+1}^b &= f_t \left(\eta_t^b, y_{\omega,t}, e_{\omega,t} \right) \\
 &= 2 \log \left(\tilde{q}_{t+1}^{bn} \odot \tilde{q}_t^{nb} \odot \exp_q \left(\frac{\eta_t^b}{2} \right) \odot \exp_q \left(\frac{T}{2} (y_{\omega,t} - e_{\omega,t}) \right) \right) \\
 &= 2 \log \left(\exp_q \left(-\frac{T}{2} y_{\omega,t} \right) \odot \exp_q \left(\frac{\eta_t^b}{2} \right) \odot \exp_q \left(\frac{T}{2} (y_{\omega,t} - e_{\omega,t}) \right) \right), \quad (135)
 \end{aligned}$$

where the last equality is based on the update of the linearization point. From (135), the following derivatives can be obtained

$$\begin{aligned}
 F_t &= \left. \frac{\partial f_t(\eta_t^b, y_{\omega,t}, e_{\omega,t})}{\partial \eta_t^b} \right|_{\substack{e_{\omega,t}=0 \\ \eta_t^b=0}} \\
 &= 2 \left. \frac{\partial}{\partial \eta_t^b} \log \left(\exp_q \left(-\frac{T}{2} y_{\omega,t} \right) \odot \exp_q \left(\frac{\eta_t^b}{2} \right) \odot \exp_q \left(\frac{T}{2} y_{\omega,t} \right) \right) \right|_{\eta_t^b=0} \\
 &= \exp_R(-T y_{\omega,t}), \quad (136a)
 \end{aligned}$$

$$\begin{aligned}
 G_t &= \left. \frac{\partial f_t(\eta_t^b, y_{\omega,t}, e_{\omega,t})}{\partial e_{\omega,t}} \right|_{\substack{e_{\omega,t}=0 \\ \eta_t^b=0}} \\
 &= 2 \left. \frac{\partial}{\partial e_{\omega,t}} \log \left(\exp_q \left(-\frac{T}{2} y_{\omega,t} \right) \odot \exp_q \left(\frac{T}{2} (y_{\omega,t} + e_{\omega,t}) \right) \right) \right|_{e_{\omega,t}=0} \\
 &\approx T \mathcal{I}_3. \quad (136b)
 \end{aligned}$$

In the measurement update of the EKF, the state η_t^b is updated using the accelerometer and magnetometer measurements. The accelerometer measurement equation can be written in terms of the state η_t^b as

$$\begin{aligned}
 y_{a,t} &= -R_t^{bn} g^n + e_{a,t} = - \left(\exp([\eta_t^b \times]) \right)^T \tilde{R}_t^{bn} g^n + e_{a,t} \\
 &\approx - \left(\mathcal{I}_3 - [\eta_t^b \times] \right) \tilde{R}_t^{bn} g^n + e_{a,t} = - \tilde{R}_t^{bn} g^n - [\tilde{R}_t^{bn} g^n \times] \eta_t^b + e_{a,t}. \quad (137)
 \end{aligned}$$

Equivalently, the magnetometer measurement equation can be written in terms of the state η_t^b as

$$\begin{aligned}
 y_{m,t} &= R_t^{bn} m^n + e_{m,t} = \left(\exp([\eta_t^b \times]) \right)^T \tilde{R}_t^{bn} m^n + e_{m,t} \\
 &\approx \left(\mathcal{I}_3 - [\eta_t^b \times] \right) \tilde{R}_t^{bn} m^n + e_{m,t} = \tilde{R}_t^{bn} m^n + [\tilde{R}_t^{bn} m^n \times] \eta_t^b + e_{m,t}. \quad (138)
 \end{aligned}$$

From these equations, the derivatives H_t as defined in (125) can straightforwardly be computed.

After the measurement update, the orientation deviation $\hat{\eta}_t^b$ is non-zero. Hence, as an additional step in the EKF, we update the linearization point and reset the state. In our algorithm, we consider the relinearization as the “measurement update” for the linearization point, i.e. the relinearization updates the linearization point $\tilde{q}_{t|t-1}^{nb}$ to $\tilde{q}_{t|t}^{nb}$ as

$$\tilde{q}_{t|t}^{nb} = \tilde{q}_{t|t-1}^{nb} \odot \exp_q \left(\frac{\hat{\eta}_t^b}{2} \right). \quad (139)$$

The covariance around the updated linearization point can be computed as

$$P_{t|t} = J_t \tilde{P}_{t|t} J_t^\top, \quad J_t = \exp_R(-\hat{\eta}_t^b). \quad (140)$$

The resulting EKF is summarized in Algorithm 4. Note the similarities between this algorithm and Algorithm 2.

Algorithm 4 Orientation estimation using an EKF with orientation deviation states

INPUTS: Inertial data $\{y_{a,t}, y_{\omega,t}\}_{t=1}^N$, magnetometer data $\{y_{m,t}\}_{t=1}^N$ and covariance matrices Σ_ω , Σ_a and Σ_m .

OUTPUTS: An estimate of the orientation $\hat{q}_{t|t}^{\text{nb}}$ and the covariance $P_{t|t}$ for $t = 1, \dots, N$.

1. Compute $\hat{q}_1^{\text{nb}}$ and Σ_i as described in Section 3.6 and set $\hat{q}_{1|1}^{\text{nb}} = \hat{q}_1^{\text{nb}}$ and $P_{1|1} = \Sigma_{\eta,i}$.

2. **For** $t = 2, \dots, N$ **do**

(a) Time update

$$\hat{q}_{t|t-1}^{\text{nb}} = \hat{q}_{t-1|t-1}^{\text{nb}} \odot \exp_q\left(\frac{T}{2} y_{\omega,t-1}\right), \quad (141a)$$

$$P_{t|t-1} = F_{t-1} P_{t-1|t-1} F_{t-1}^\top + G Q G^\top, \quad (141b)$$

with $F_{t-1} = \exp_R(-T y_{\omega,t-1})$, $G = T \mathcal{I}_3$ and $Q = \Sigma_\omega$.

(b) Measurement update

$$\hat{\eta}_t^b = K_t \varepsilon_t, \quad (142a)$$

$$\tilde{P}_{t|t} = P_{t|t-1} - K_t S_t K_t^\top, \quad (142b)$$

with ε_t , K_t and S_t defined in (124) and

$$y_t = \begin{pmatrix} y_{a,t} \\ y_{m,t} \end{pmatrix}, \quad \hat{y}_{t|t-1} = \begin{pmatrix} -\tilde{R}_{t|t-1}^{\text{bn}} g^n \\ \tilde{R}_{t|t-1}^{\text{bn}} m^n \end{pmatrix},$$

$$H_t = \begin{pmatrix} -[\tilde{R}_{t|t-1}^{\text{bn}} g^n \times] \\ [\tilde{R}_{t|t-1}^{\text{bn}} m^n \times] \end{pmatrix}, \quad R = \begin{pmatrix} \Sigma_a & 0 \\ 0 & \Sigma_m \end{pmatrix}.$$

(c) Relinearize

$$\hat{q}_{t|t}^{\text{nb}} = \hat{q}_{t|t-1}^{\text{nb}} \odot \exp_q\left(\frac{\hat{\eta}_t^b}{2}\right), \quad P_{t|t} = J_t \tilde{P}_{t|t} J_t^\top, \quad (143)$$

with $J_t = \exp_R(-\hat{\eta}_t^b)$.

end for

4.4 Evaluation based on experimental and simulated data

In this section, we apply the algorithms described in Sections 4.1 – 4.3 to both simulated and experimental data. Some general characteristics of the orientation

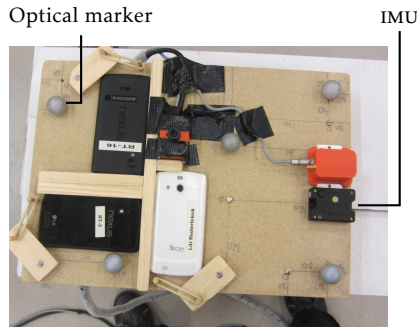


Figure 19: Experimental setup where an IMU is used to collect inertial and magnetometer measurements. Optical markers are tracked using multiple cameras, leading to accurate reference position and orientation estimates. Note that the experimental setup also contains additional IMUs and smartphones. The data from these sensors is not considered in this work.

estimation algorithms will be illustrated and the quality of the different algorithms will be analyzed. The simulated data allows for controlled analysis of the workings of the algorithms. Furthermore, it allows us to compare the different algorithms using Monte Carlo simulations. The experimental data shows the applicability to real-world scenarios. We will start by introducing the data sets.

Experimental data is collected using the setup shown in Figure 19, where data is collected using multiple mobile IMUs and smartphones. The algorithms presented in this section can be applied to measurements from any of these devices. However, we focus our analysis on the data from the Trivisio Colibri Wireless IMU (Trivisio Prototyping GmbH, 2016). In Figure 20, the inertial and magnetometer measurements from this IMU are displayed for around 100 s during which the sensor is rotated around all three axes. The experiments are performed in a lab equipped with multiple cameras (Vicon, 2016), able to track the optical markers shown in Figure 19. This provides accurate reference position and orientation information, against which we can compare our estimates.⁵

In Figure 21, simulated inertial and magnetometer measurements are displayed. The data represents a sensor that is kept stationary for 100 samples, after which it is rotated around all three axes. The sensor is assumed to be rotated around the origin of the accelerometer triad. Hence, during the entire data set, the accelerometer is assumed to only measure the gravity vector. The magnitude of the simulated gravity vector is 9.82 m/s^2 . The magnitude of the simulated local magnetic field is equal to one. Its direction is approximately equal to that in Linköping, Sweden, where a dip angle of 71° leads to a magnetic field

⁵For comparison, the optical and IMU data need to be time-synchronized and aligned. We synchronize the data by correlating the norms of the gyroscope measurements and of the angular velocity estimated by the optical system. Alignment is done using the orientation estimates in combination with Theorem 4.2 from Hol (2011).

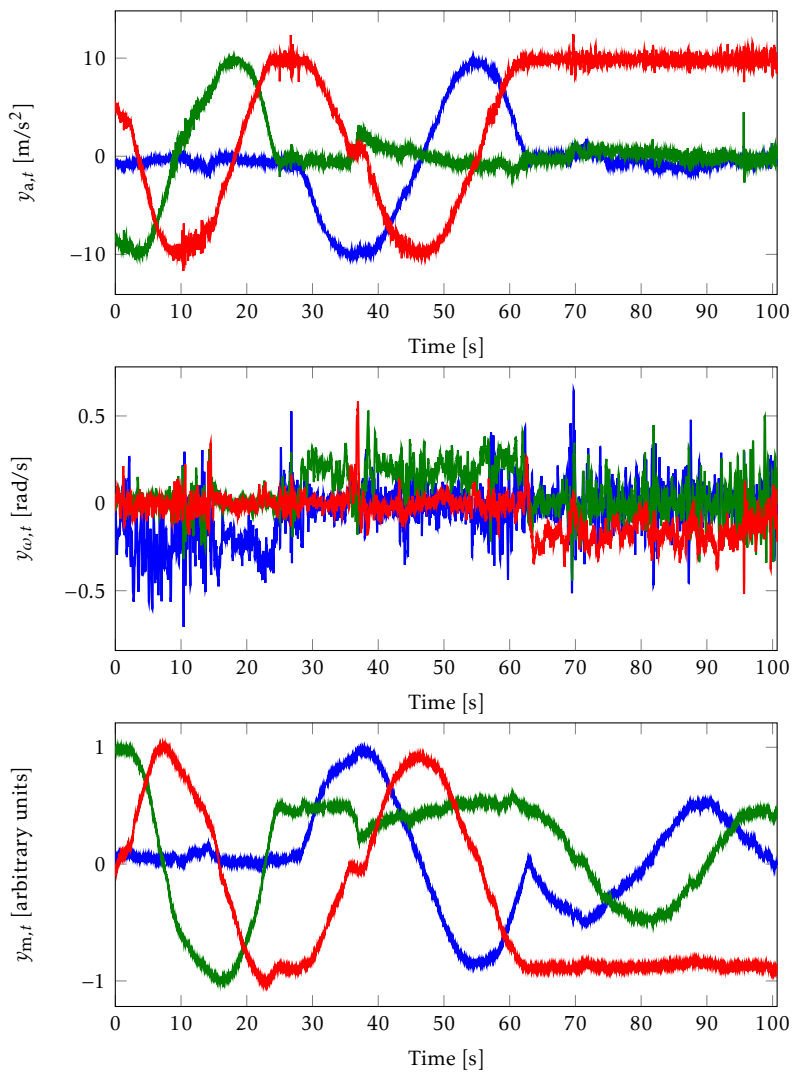


Figure 20: Measurements from an accelerometer ($y_{a,t}$, top), a gyroscope ($y_{\omega,t}$, middle) and a magnetometer ($y_{m,t}$, bottom) for 100 s of data collected with the IMU shown in Figure 19.

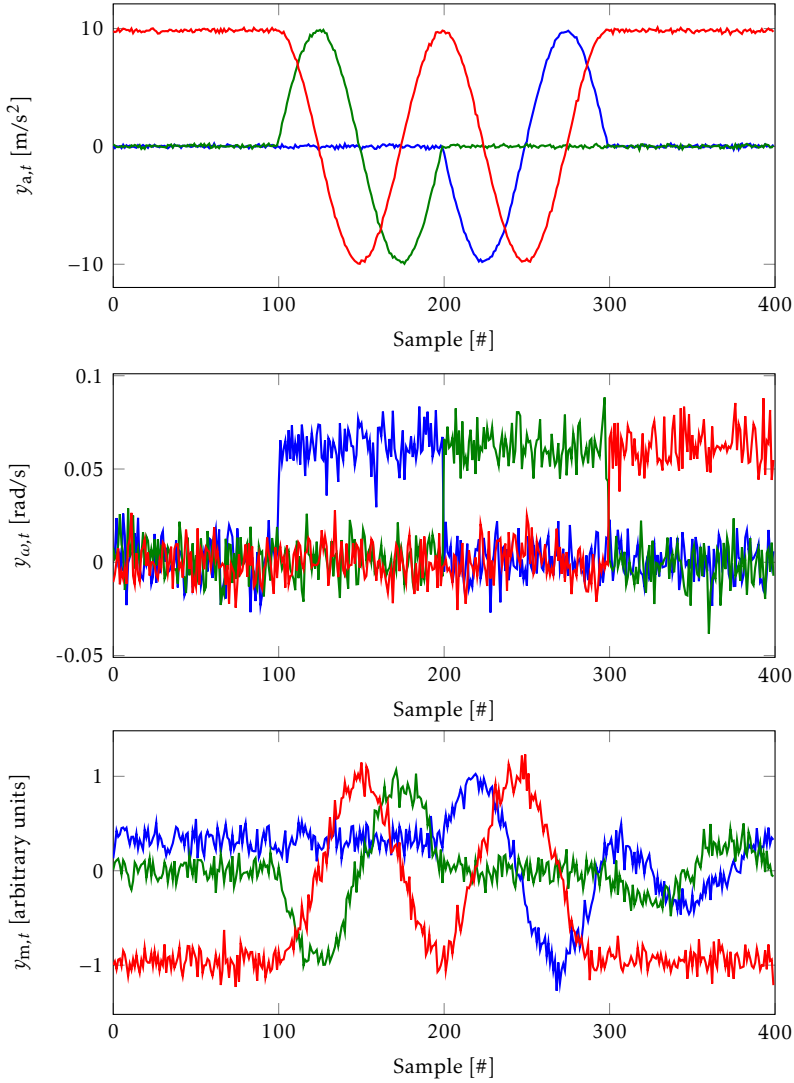


Figure 21: Simulated measurements from an accelerometer ($y_{a,t}$, top), a gyroscope ($y_{\omega,t}$, middle) and a magnetometer ($y_{m,t}$, bottom).

$m^n = (0.33 \quad 0 \quad -0.95)^\top$. The simulated noise levels are

$$\begin{aligned} e_{a,t} &\sim \mathcal{N}(0, \sigma_a^2 \mathcal{I}_3), & \sigma_a &= 1 \cdot 10^{-1}, \\ e_{\omega,t} &\sim \mathcal{N}(0, \sigma_\omega^2 \mathcal{I}_3), & \sigma_\omega &= 1 \cdot 10^{-2}, \\ e_{m,t} &\sim \mathcal{N}(0, \sigma_m^2 \mathcal{I}_3), & \sigma_m &= 1 \cdot 10^{-1}. \end{aligned}$$

Note that we deliberately chose the noise levels to be fairly high, to clearly illustrate the workings of the different algorithms.

Although our algorithms parametrize the orientation as quaternions, it is typically more intuitive to visualize the orientation estimates in Euler angles. Hence, we visualize our results in terms of roll, pitch and heading (yaw) angles. Both for the experimental data and for the simulated data, we are able to compare our estimates $\hat{q}_t^{\text{nb}}$ to reference orientations denoted $q_{\text{ref},t}^{\text{nb}}$. To represent the orientation error, we compute a difference quaternion Δq_t as

$$\Delta q_t = \hat{q}_t^{\text{nb}} \odot (q_{\text{ref},t}^{\text{nb}})^c, \quad (144)$$

which can be converted to Euler angles for visualization. Note that using this definition, the orientation errors in Euler angles can be interpreted as the errors in roll, pitch and heading.

General characteristics

In this section, we will discuss some general characteristics of the orientation estimation problem and illustrate them in three different examples. Our goal is not to compare the different estimation algorithms, but to illustrate some characteristics common to all of them.

In Example 8 we focus on the accuracy of the orientation estimates that can be obtained if the state space model (87) is completely true. We illustrate that it is typically easier to obtain accurate roll and pitch estimates than it is to obtain accurate heading estimates.

Example 8: Orientation estimation using inertial and magnetometer data

The orientation errors from the smoothing optimization approach in Algorithm 1 using simulated inertial and magnetometer measurements as illustrated in Figure 21 are depicted in Figure 22. For comparison we also show the orientation errors from dead-reckoning the gyroscope measurements (see also Section 1.2).

Although the accelerometer and magnetometer measurement noises are of equal magnitude, the heading angle is estimated with less accuracy than the roll and pitch angles. The reason for this is twofold. First, the signal to noise ratio for the magnetometer is worse than that of the accelerometer, since the magnetometer signal has a magnitude of 1 while the accelerometer signal has a magnitude of 9.82 m/s^2 . Second, only the horizontal component of the local magnetic field vector provides heading information. This component is fairly small due to the large dip angle (71°) in Linköping, Sweden.

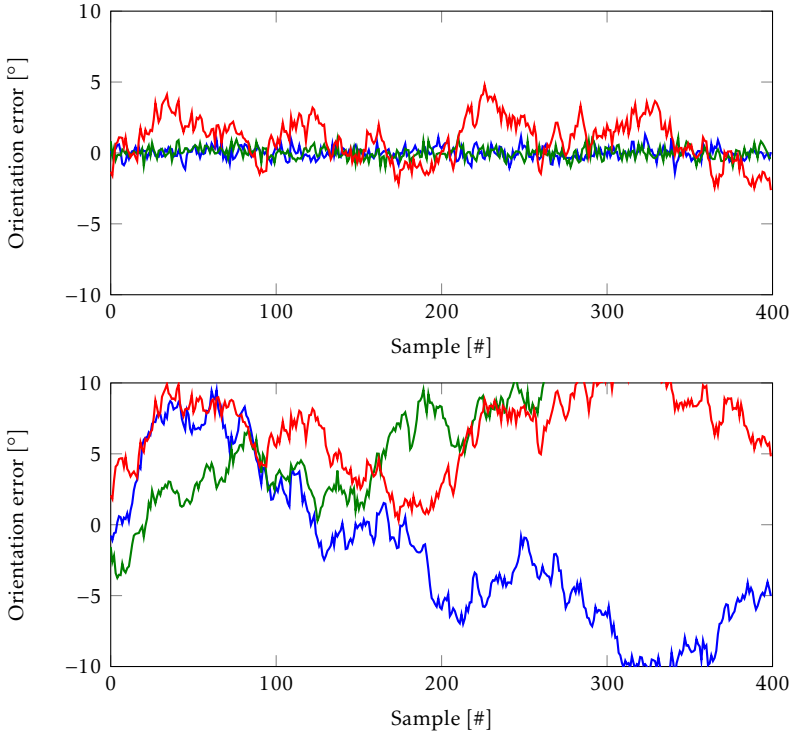


Figure 22: Orientation errors in roll (blue), pitch (green) and heading (red) using simulated measurements for (top) Algorithm 1 using inertial and magnetometer measurements and (bottom) dead-reckoning of the gyroscope measurements.

The accelerometer provides inclination information, while the magnetometer provides heading information, see Section 3.4. In case only inertial measurements and no magnetometer measurements are available, the heading can only be estimated using the gyroscope measurements. As discussed in Section 1.2, the heading estimates will then drift due to the integration of the gyroscope measurements. This is illustrated in Example 9.

Example 9: Orientation estimation using only inertial measurements

The orientation errors from the smoothing optimization approach in Algorithm 1 using simulated inertial measurements as presented in Figure 21 can be found in Figure 23. The roll and pitch angles can be seen to be accurate, while the heading angle drifts. Again, we also show the orientation errors from dead-reckoning the gyroscope measurements for comparison. The drift in the heading angle can be seen to be similar to the drift from dead-reckoning of the gyroscope measurements.

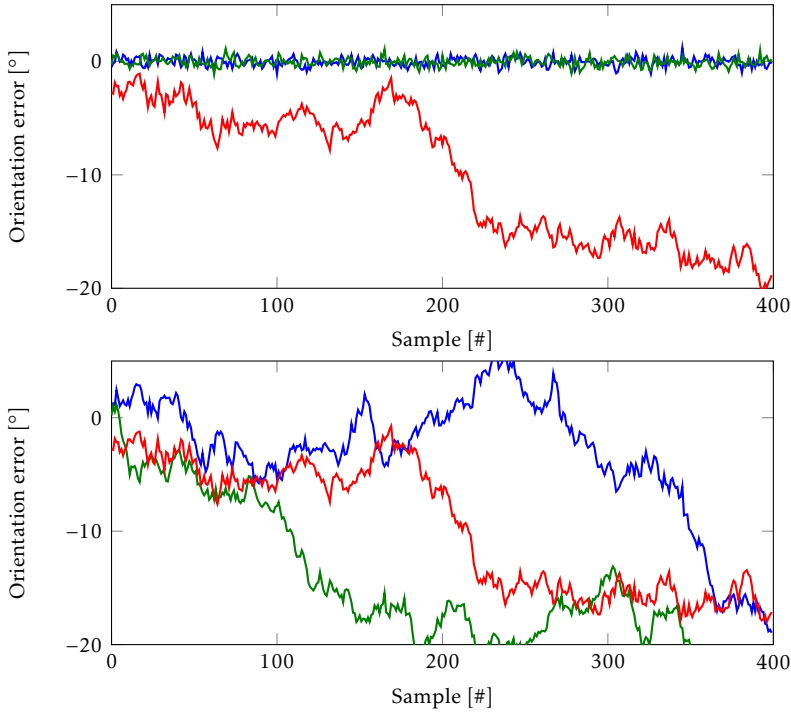


Figure 23: Orientation errors in roll (blue), pitch (green) and heading (red) using simulated measurements for (top) Algorithm 1 using inertial measurements only and (bottom) dead-reckoning of the gyroscope measurements.

The two examples above assume that our state space model (87) is an accurate description of the measurements. In practice, however, this is not always the case, for instance due to the presence of magnetic material in the vicinity of the sensor. In Example 10 we illustrate that if the state space model does not accurately describe the data, it is not possible to obtain accurate orientation estimates.

Example 10: Orientation estimation in the presence of magnetic material

We simulate 400 samples of stationary data. Between samples 150 and 250, we simulate the presence of a magnetic material, causing a change in the magnetic field of $(0.1 \ 0.3 \ 0.5)^\top$. In Figure 24, we show the adapted magnetometer data and the orientation estimates using the smoothing optimization approach from Section 4.1. As can be seen, the orientation estimates show significant errors when the magnetic material is present.

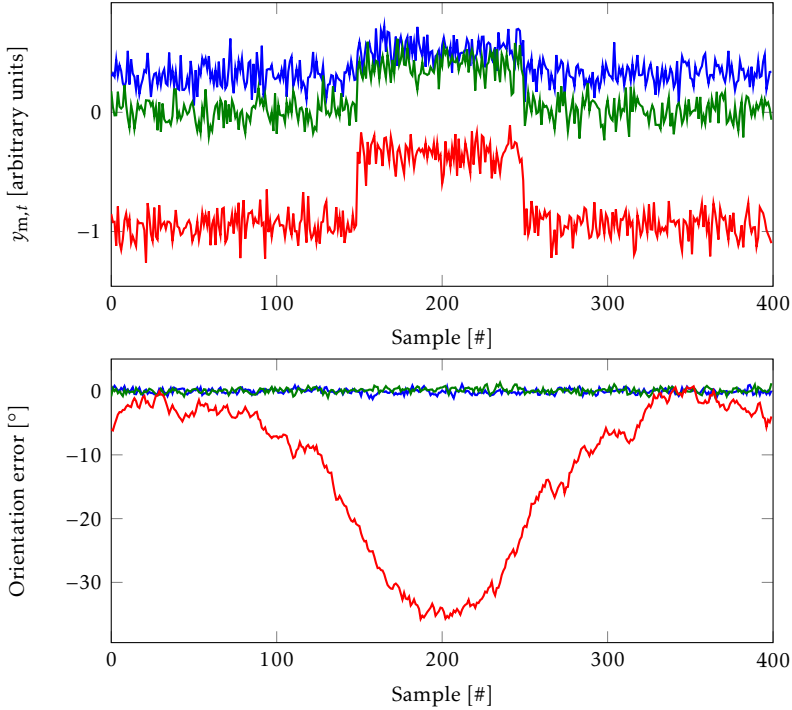


Figure 24: Top: Simulated magnetometer measurements $y_{m,t}$ for 400 samples of stationary data. Between samples 150 and 250 we simulate the presence of a magnetic material in the vicinity of the sensor. Bottom: Orientation estimates in roll (blue), pitch (green) and heading (red) using the simulated inertial and magnetometer measurements.

Representing uncertainty

So far, we have discussed the *quality* of the orientation estimates in three different examples. However, we did not discuss the *uncertainty* of the estimates. We will now discuss how these uncertainties can be displayed and interpreted and highlight some difficulties with this.

Both the optimization and the EKF approaches discussed in Sections 4.1 – 4.3 compute the uncertainty of the estimates in terms of a covariance matrix. Let us use the more general notation $\text{cov}(\hat{\eta}_t^b)$ for the covariance of the orientation deviation states $\hat{\eta}_t^b$, $t = 1, \dots, N$ computed in Algorithms 1, 2 and 4, and $\text{cov}(\hat{q}_t^{\text{nb}})$ for the covariance of the quaternion states $\hat{q}_t^{\text{nb}}$, $t = 1, \dots, N$ computed in Algorithm 3. If the states would be in normal, Euclidean space, the square root of the diagonal of these matrices would represent the standard deviation σ of the estimates in the different directions. These could then be visualized by for instance plotting 3σ confidence bounds around the estimates.

One could imagine that an equivalent way of visualizing the orientation devi-

ation uncertainties, would be to compute the 3σ bounds in terms of orientation deviations in each of the three directions as

$$(\Delta\eta_{i,t}^b)_{+3\sigma} = +3\sqrt{(\text{cov}(\hat{\eta}_t^b))_{ii}}, \quad i = 1, \dots, 3, \quad (145a)$$

$$(\Delta\eta_{i,t}^b)_{-3\sigma} = -3\sqrt{(\text{cov}(\hat{\eta}_t^b))_{ii}}, \quad i = 1, \dots, 3, \quad (145b)$$

after which the bounds can be parametrized in terms of quaternions as

$$(q_t^{\text{nb}})_{+3\sigma} = \hat{q}_t^{\text{nb}} \odot \exp_q(\Delta\eta_t^b)_{+3\sigma}, \quad (146a)$$

$$(q_t^{\text{nb}})_{-3\sigma} = \hat{q}_t^{\text{nb}} \odot \exp_q(\Delta\eta_t^b)_{-3\sigma}. \quad (146b)$$

The resulting estimates and bounds are visualized in terms of Euler angles in Figure 25 for simulated data similar to the data presented in Figure 21. As can be seen, the bounds are difficult to interpret due to the wrapping of the Euler angles.

As argued in Forster et al. (2016), it is more intuitive to directly represent the uncertainty in terms of orientation deviations. Instead of using the covariance estimate $\text{cov}(\eta_t^b)$ directly from our algorithms, we believe that it is more intuitive to represent this covariance in the navigation frame as

$$\text{cov}(\hat{\eta}_t^n) = \hat{R}_t^{\text{nb}} \text{cov}(\hat{\eta}_t^b) \hat{R}_t^{\text{bn}}. \quad (147)$$

The covariance $\text{cov}(\hat{\eta}_t^n)$ can be interpreted as the uncertainty in the roll, pitch and heading angles as illustrated in Example 11.

Example 11: Orientation estimation using only inertial measurements
(continued)

Since the accelerometer provides only inclination information, in the case of Example 9 where magnetometer measurements are unavailable, we expect only the roll and pitch angles to be estimated with small uncertainty. In fact, we expect the uncertainty of the heading at $t = 1$ to be equal to the uncertainty of the initial Σ_i from Section 3.6 and to steadily grow over time, depending on the amount of gyroscope noise. In Figure 26, we plot the standard deviation σ of the orientation estimates computed using the smoothing algorithm from Section 4.1 as the square root of the diagonal elements of $\text{cov}(\hat{\eta}_t^n)$. As can be seen, the standard deviation of the yaw angle at $t = 1$ is indeed 20° as modeled in Section 3.6. The increase in the uncertainty in the yaw angle exactly matches the increase of the uncertainty due to dead-reckoning.

From Example 11 it can be concluded that $\text{cov}(\hat{\eta}_t^n)$ seems to be an intuitive measure of the uncertainty of the orientation estimates. Using (147), the covariances $\text{cov}(\hat{\eta}_t^b)$ computed by Algorithms 1, 2 and 4 can be converted to this measure.

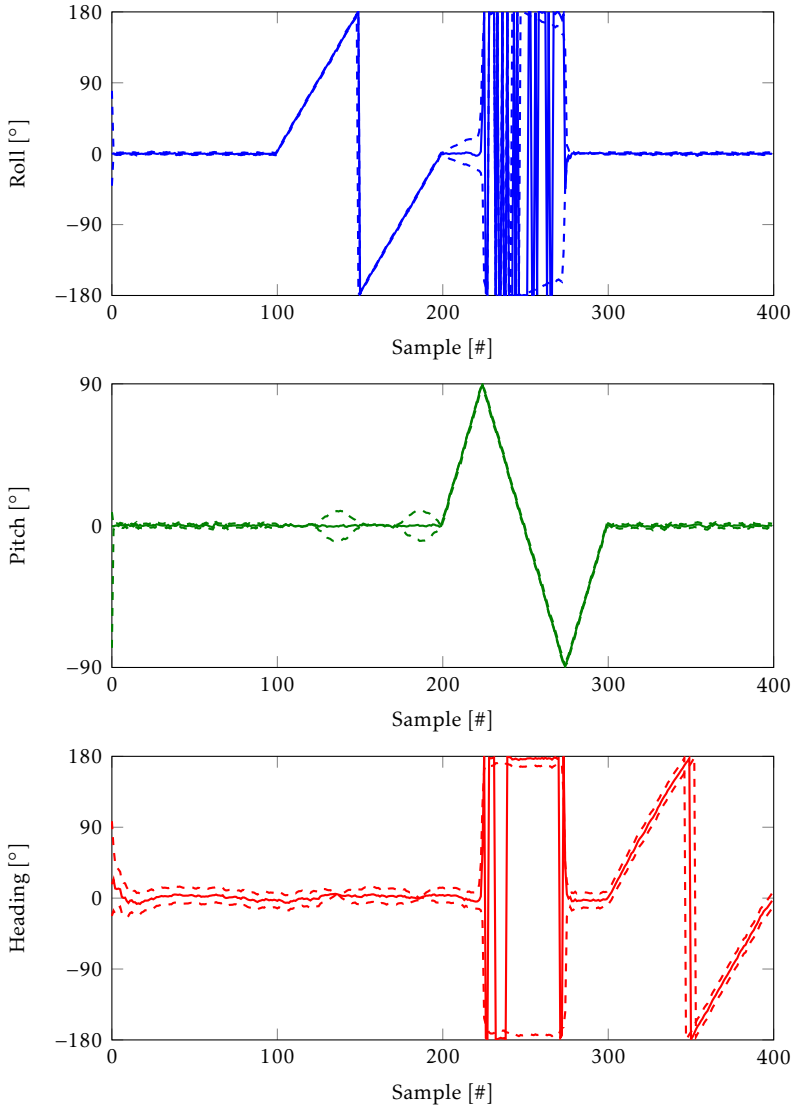


Figure 25: Orientation estimates (solid) and 3σ bounds (dashed) in roll (blue), pitch (green) and heading (red) using inertial and magnetometer measurements.

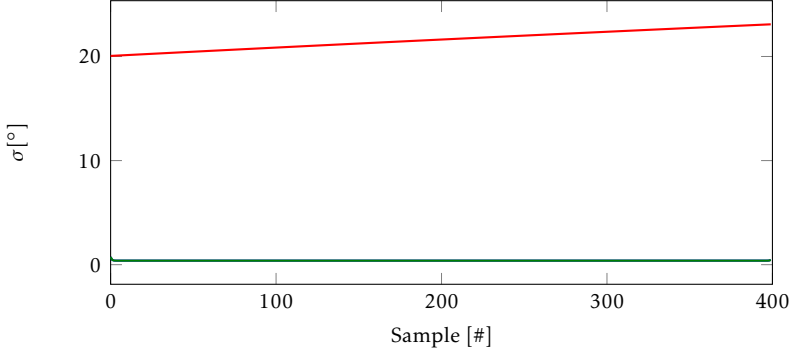


Figure 26: Standard deviation σ in degrees of the orientation estimates in roll (blue), pitch (green) and heading (red) using only inertial measurements.

The covariance $\text{cov}(\hat{q}_t^{\text{nb}})$ computed by Algorithm 3 relates to these covariances as

$$\begin{aligned} \text{cov}(\hat{q}_t^{\text{nb}}) &= \text{cov}(\tilde{q}_t^{\text{nb}} \odot \exp_q(\frac{\hat{\eta}_t^{\text{b}}}{2})) \\ &= \frac{1}{4} (\tilde{q}_t^{\text{nb}})^{\text{L}} \frac{d \exp_q(\hat{\eta}_t^{\text{b}})}{d \hat{\eta}_t^{\text{b}}} \text{cov}(\hat{\eta}_t^{\text{b}}) \left(\frac{d \exp_q(\hat{\eta}_t^{\text{b}})}{d \hat{\eta}_t^{\text{b}}} \right)^{\text{T}} (\tilde{q}_t^{\text{bn}})^{\text{L}}, \end{aligned} \quad (148a)$$

$$\begin{aligned} \text{cov}(\hat{\eta}_t^{\text{b}}) &= \text{cov}(2 \log_q(\tilde{q}_t^{\text{bn}} \odot \hat{q}_t^{\text{nb}})) \\ &= 4 \frac{d \log_q(q)}{dq} (\tilde{q}_t^{\text{bn}})^{\text{L}} \text{cov}(\hat{q}_t^{\text{nb}}) (\tilde{q}_t^{\text{nb}})^{\text{L}} \left(\frac{d \log_q(q)}{dq} \right)^{\text{T}}, \end{aligned} \quad (148b)$$

where we make use of the fact that $(q^{\text{L}})^{\text{T}} = (q^{\text{c}})^{\text{L}}$. The relations (147) and (148) allow us to compare the orientation estimates and covariances from the different algorithms in more detail in Example 12.

Example 12: Orientation estimation using only inertial measurements (continued)

As discussed in Example 11, using only inertial measurements and no magnetometer measurements, we can only expect to be able to accurately estimate the inclination. The uncertainty of the heading estimates grows over time. We will now analyze the behavior of the different algorithms in more detail for this specific example. In Table 1, we show the root mean square error (RMSE) values over 100 Monte Carlo simulations for Algorithms 1 – 4. In Figure 27, we also represent the orientation estimates from the four algorithms for one of these realizations. As can be seen from both Table 1 and Figure 27, as expected, the smoothing algorithm outperforms the other algorithms. However, more surprisingly, the EKF with quaternion states has much larger errors in the heading angle. In Figure 28, we also show the standard deviations of the estimates from all algorithms. As can be seen, the EKF with quaternion states over-estimates its confidence in the estimates of the heading direction. This can most likely be attributed to linearization issues.

Table 1: Mean RMSE values over 100 Monte Carlo simulations estimating orientation using only inertial measurements.

RMSE	Roll [°]	Pitch [°]	Heading [°]
Smoothing optimization (Alg. 1)	0.39	0.39	7.50
Filtering optimization (Alg. 2)	0.46	0.46	7.50
EKF quaternions (Alg. 3)	0.46	0.46	18.41
EKF orientation deviation (Alg. 4)	0.46	0.46	7.50

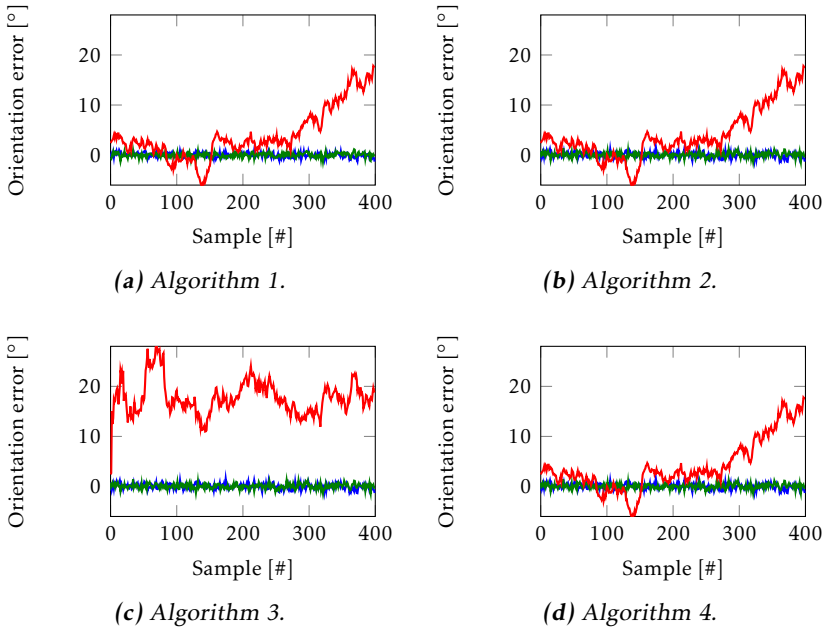


Figure 27: Orientation estimates of Algorithms 1 – 4 in roll (blue), pitch (green) and heading (red) using only inertial measurements.

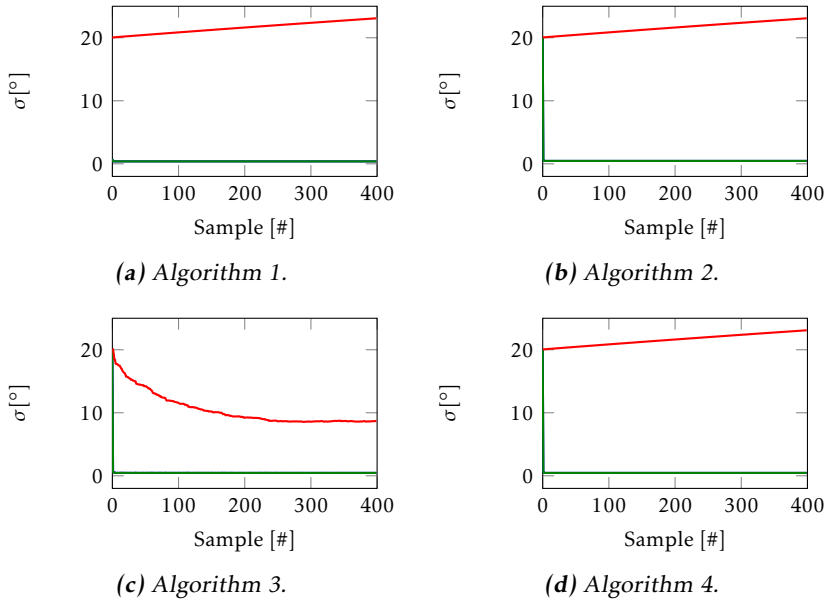


Figure 28: Standard deviation σ in degrees of the orientation estimates of Algorithms 1 – 4 in roll (blue), pitch (green) and heading (red) using only inertial measurements.

From Example 12, it can be concluded that not properly estimating the covariance can have negative effects on the quality of the estimates. It can also be concluded from this section that covariances can best be represented in terms of $\text{cov}(\hat{\eta}_t^n)$ but that they are difficult to visualize in Euler angles. Because of that, in the remainder of this section, we will typically plot the uncertainty in a separate plot as in Figure 28.

Comparing the different algorithms

We use the experimental data presented in Figure 20 to assess the quality of the estimates from the different algorithms. The RMSE values as compared to the optical reference system for the different methods described in this section are summarized in Table 2. As can be seen, the three filtering solutions perform similarly while the smoothing solution outperforms them. As an illustration of the estimates, the orientation estimates as obtained using the smoothing algorithm and the orientations from the optical reference system are shown in Figure 29. Since it is difficult to draw quantitative conclusions based on only one data set, we will also study the accuracy of the different methods using simulated data.

We run 100 Monte Carlo simulations where the simulated data illustrated in Figure 21 is generated with different noise realizations. Table 3 shows the mean RMSE for the four estimation algorithms. The smoothing approach can be seen to

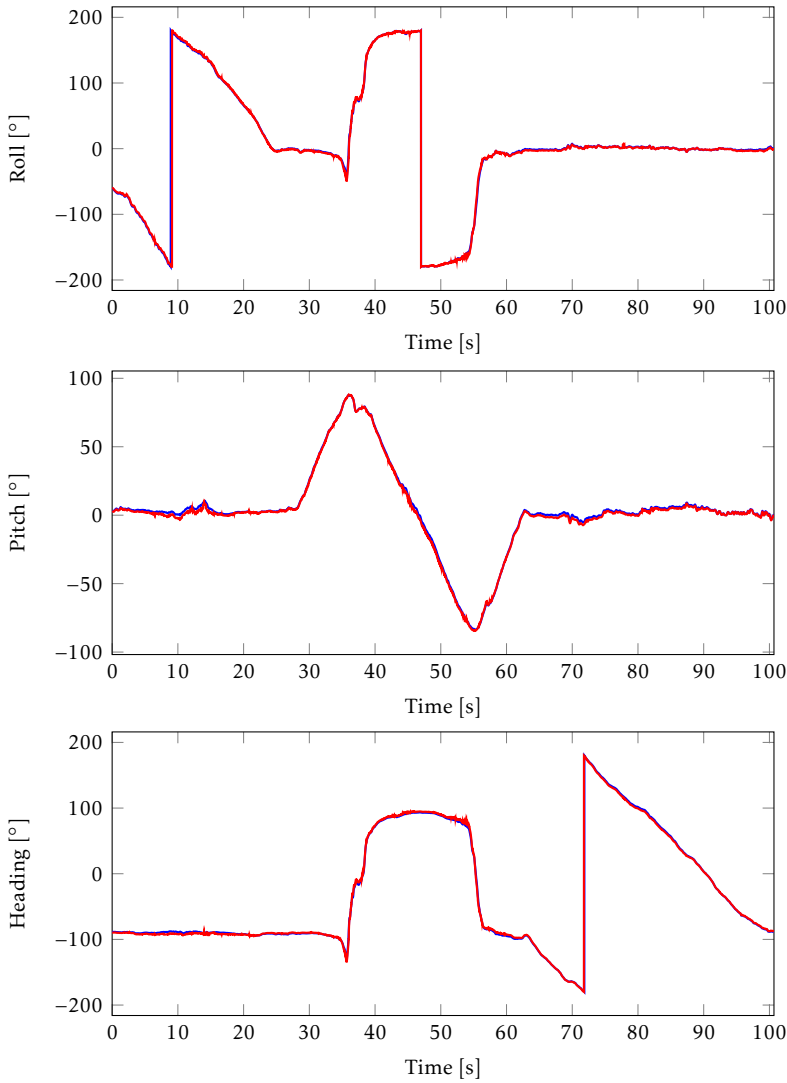


Figure 29: Red: Orientation from the optical reference system. Blue: Orientation estimates obtained using Algorithm 1 for the experimental data from Figure 20.

Table 2: RMSE of the orientation estimates obtained using Algorithms 1 – 4 and the experimental data presented in Figure 20.

RMSE	Roll [°]	Pitch [°]	Heading [°]
Smoothing optimization (Alg. 1)	1.24	1.16	1.65
Filtering optimization (Alg. 2)	1.34	1.12	2.05
EKF quaternions (Alg. 3)	1.34	1.12	2.05
EKF orientation deviation (Alg. 4)	1.34	1.12	2.05

Table 3: Mean RMSE of the orientation estimates from 100 Monte Carlo simulations using Algorithms 1 – 4.

RMSE	Roll [°]	Pitch [°]	Heading [°]
Smoothing optimization (Alg. 1)	0.39	0.39	2.35
Filtering optimization (Alg. 2)	0.45	0.45	3.52
EKF quaternions (Alg. 3)	0.45	0.45	3.54
EKF orientation deviation (Alg. 4)	0.45	0.45	3.52

outperform the filtering approaches. The estimates and their covariances for one of the noise realizations are shown in Figures 30 and 31, respectively. The filtering approaches estimate the standard deviation of the orientation errors at $t = 1$ to be equal to 20° . After this, they can be seen to converge to around 3.15° degrees for the heading angle and 0.46° for roll and pitch angles. The smoothing algorithm estimates an uncertainty in the heading angle of around 3.17° for the first and last sample, while converging to a standard deviation of 2.25° for the middle of the data set. For the roll and pitch angles, the initial and final uncertainties are estimated to be around 0.73° , converging to 0.39° for the middle of the data set. Note that these values correspond fairly well with the RMSE values in Table 3.

For the Monte Carlo simulations described above, the three filtering algorithms perform similarly. However, differences can be seen when an update of the filter needs to correct the orientation estimates significantly. Examples for when this happens are when the initial orientation is not accurately known or when magnetometer measurements are not available for a longer period of time. In these cases, the uncertainty of the state is large and large corrections to the state estimates are needed when measurements become available. To analyze this case in more detail, we assume that the estimate of the initial orientation $\hat{q}_1^{\text{nb}}$ is normal distributed around the true initial orientation with a standard deviation of 20° . Hence, we do not use the first accelerometer and magnetometer data for initialization. Note that the standard deviation of 20° is equal to the uncertainty on the initial state assumed by the algorithms. The results for 100 Monte Carlo simulations are sum-

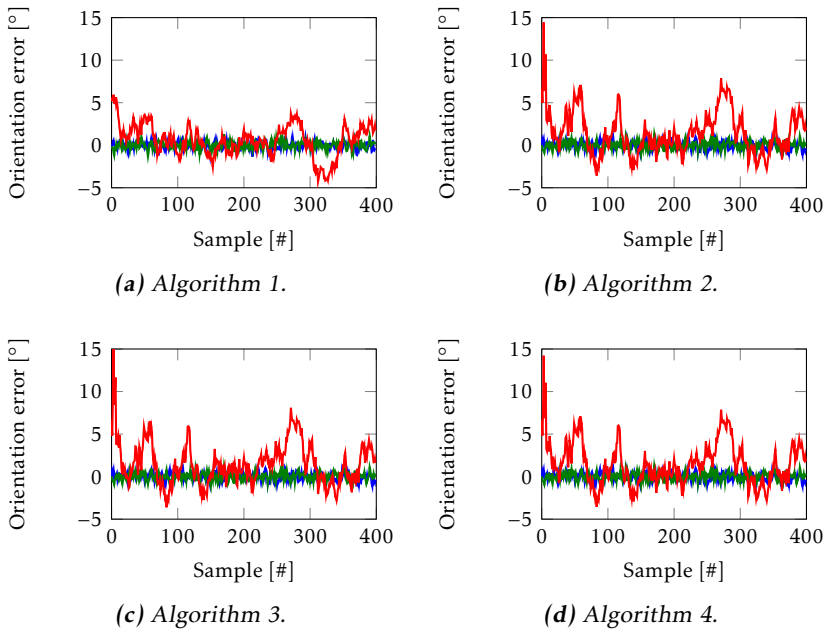


Figure 30: Orientation errors of Algorithms 1 – 4 in roll (blue), pitch (green) and heading (red) using simulated inertial and magnetometer measurements.

Table 4: Mean RMSE of the orientation estimates from 100 Monte Carlo simulations. The estimate of the initial orientation is assumed to be normal distributed around the true initial orientation with a standard deviation of 20° .

RMSE	Roll [°]	Pitch [°]	Heading [°]
Smoothing optimization (Alg. 1)	0.39	0.39	2.21
Filtering optimization (Alg. 2)	1.02	0.94	3.52
EKF quaternions (Alg. 3)	1.03	0.96	4.29
EKF orientation deviation (Alg. 4)	1.03	0.96	3.53

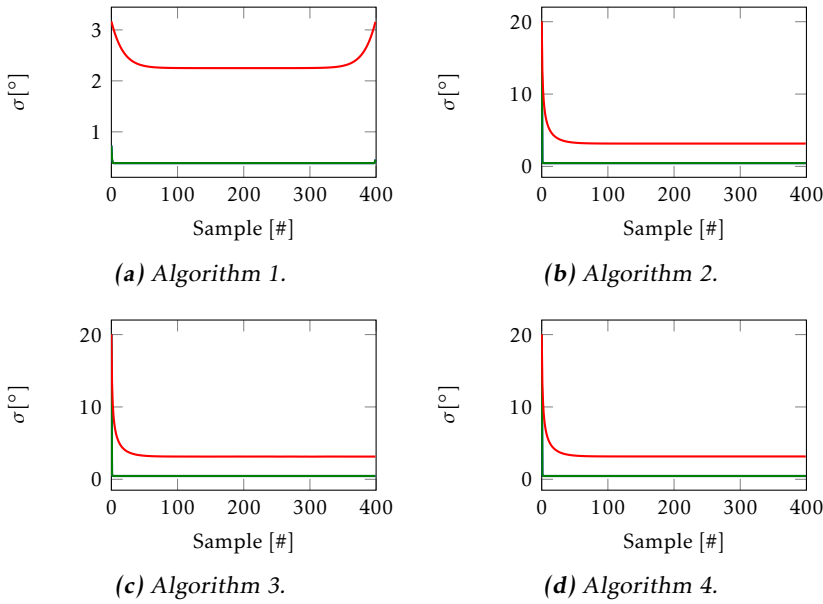


Figure 31: Standard deviation σ in degrees of the orientation estimates of Algorithms 1 – 4 in roll (blue), pitch (green) and heading (red) using simulated inertial and magnetometer measurements. Note the different scale on the vertical axis of (a) as compared to (b) – (d).

marized in Table 4. As can be seen, specifically the EKF with quaternion states performs worse than Algorithm 2 and Algorithm 4 for this data.

Which algorithm should be used is highly application-specific. However, in general it can be concluded that all four algorithms actually produce fairly good orientation estimates, assuming that the models from Section 3.7 are indeed valid. The smoothing algorithm performs better than the filtering approaches but is also the most computationally expensive. The EKF with quaternion states suffers from linearization issues when large orientation corrections need to be made or when magnetometer data is unavailable.

4.5 Extending to pose estimation

In Section 4.4, we have evaluated the workings of Algorithms 1 – 4 for orientation estimation. The estimation methods presented in Sections 4.1 – 4.3 can also be used to estimate the sensor’s pose using the state space model (86). The pose

estimation problem can be written as a smoothing optimization problem as

$$\hat{x}_{1:N} = \arg \min_{x_{1:N}} \underbrace{\|e_{p,i}\|_{\Sigma_{p,i}^{-1}}^2 + \|e_{v,i}\|_{\Sigma_{v,i}^{-1}}^2 + \|e_{\eta,i}\|_{\Sigma_{\eta,i}^{-1}}^2}_{\text{Prior}} + \underbrace{\sum_{t=2}^N \|e_{\omega,t}\|_{\Sigma_{\omega}^{-1}}^2 + \|e_{a,p,t}\|_{\Sigma_{a,p}^{-1}}^2 + \|e_{a,v,t}\|_{\Sigma_{a,v}^{-1}}^2}_{\text{Dynamics}} + \underbrace{\sum_{t=2}^N \|e_{p,t}\|_{\Sigma_p^{-1}}^2}_{\text{Measurement model}}, \quad (149)$$

with $x_t = (p_t^\top \quad v_t^\top \quad (\eta_t^b)^\top)^\top$ and

$$e_{p,i} = p_1^n - y_{p,1}, \quad e_{p,i} \sim \mathcal{N}(0, \Sigma_{p,i}), \quad (150a)$$

$$e_{v,i} = v_1, \quad e_{v,i} \sim \mathcal{N}(0, \Sigma_{v,i}), \quad (150b)$$

$$e_{\eta,i} = 2 \log_q(\hat{q}_1^{bn} \odot q_1^{nb}), \quad e_i \sim \mathcal{N}(0, \Sigma_{\eta,i}), \quad (150c)$$

$$e_{p,a,t} = \frac{2}{T^2} (p_{t+1}^n - p_t^n - T v_t^n) - R_t^{nb} y_{a,t} - g^n, \quad e_{p,a,t} \sim \mathcal{N}(0, \Sigma_a), \quad (150d)$$

$$e_{v,a,t} = \frac{1}{T} (v_{t+1}^n - v_t^n) - R_t^{nb} y_{a,t} - g^n, \quad e_{v,a,t} \sim \mathcal{N}(0, \Sigma_a), \quad (150e)$$

$$e_{\omega,t} = \frac{2}{T} \log_q(\hat{q}_t^{bn} \odot q_{t+1}^{nb}) - y_{\omega,t}, \quad e_{\omega,t} \sim \mathcal{N}(0, \Sigma_{\omega}), \quad (150f)$$

$$e_{p,t} = y_{p,t} - p_t^n, \quad e_{p,t} \sim \mathcal{N}(0, \Sigma_p). \quad (150g)$$

In this section, we will discuss some details about the workings of the pose estimation algorithm using this model. We will not go through a complete derivation of the four algorithms. However, the adaptations that are needed to use Algorithms 1 – 4 for pose estimation can be found in Appendix B.

An important observation is that $e_{a,p,t}$ and $e_{a,v,t}$ in (150d) and (150e) depend on the orientation R_t^{nb} . Because of this, the position, velocity and orientation states are coupled. The position measurements therefore do not only provide information about the position and velocity, but also about the orientation of the sensor. This is the reason why it is no longer essential to include magnetometer data and to assume that the acceleration is approximately zero. However, the accuracy of the orientation estimates depends on the movements of the sensor. This will be illustrated below. For this, we simulate 400 samples of inertial and position measurements for a non-rotating sensor with noise levels

$$\begin{aligned} e_{a,t} &\sim \mathcal{N}(0, \sigma_a^2 \mathcal{I}_3), & \sigma_a &= 1 \cdot 10^{-1}, \\ e_{\omega,t} &\sim \mathcal{N}(0, \sigma_{\omega}^2 \mathcal{I}_3), & \sigma_{\omega} &= 1 \cdot 10^{-2}, \\ e_{p,t} &\sim \mathcal{N}(0, \sigma_p^2 \mathcal{I}_3), & \sigma_p &= 1 \cdot 10^{-2}. \end{aligned}$$

We consider four different cases for the movement of the sensor. First, we simulate data assuming that the sensor is stationary. For this case, the position measurements provide information about the inclination of the sensor but not about the heading. This is illustrated in Example 13.

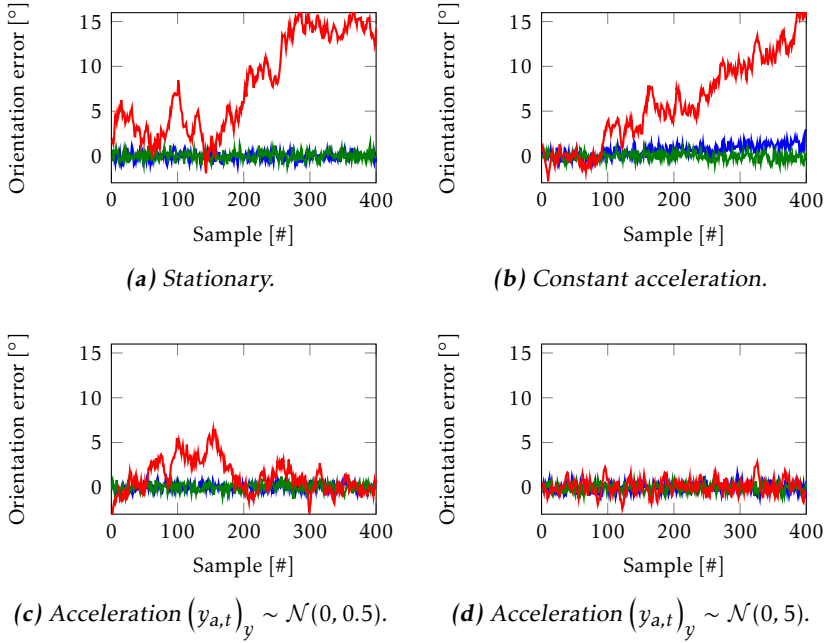


Figure 32: Orientation errors of the roll (blue), pitch (green) and heading (red) using simulated inertial and magnetometer measurements.

Example 13: Pose estimation for a stationary sensor

We estimate the pose of a stationary sensor using simulated data and a smoothing algorithm that solves (149) as described in Section 4.1. The orientation error for a specific noise realization is depicted in Figure 32a. The inclination errors can be seen to be small, while the heading estimates drift.

Next, in Example 14, we consider the case of the sensor having a constant linear acceleration. For this case, a drift in the orientation estimates can be seen in the direction that is orthogonal to the direction of the accelerometer measurements.

Example 14: Pose estimation for a sensor with constant linear acceleration

We estimate the pose of a sensor with an acceleration of 1 m/s^2 in the y -direction using simulated data and obtain smoothing estimates by solving (149). The orientation error for a specific noise realization is depicted in Figure 32b. Again, a drift can be seen in the orientation estimates. This drift is no longer only in the heading direction, but also a small drift in the roll can be observed.

Finally, in Example 15 we consider the case of time-varying linear acceleration. Based on simulated data, we show that accurate heading estimates can be obtained

Table 5: Mean RMSE of the position and orientation estimates from 100 Monte Carlo simulations. Considered are a stationary sensor, a sensor with constant acceleration and two cases of time-varying accelerations with different magnitudes.

RMSE	Roll [°]	Pitch [°]	Heading [°]	Position [cm]
Stationary	0.40	0.40	13.27	0.97
Constant acceleration	1.25	0.47	11.59	0.97
Acceleration $(y_{a,t})_y \sim \mathcal{N}(0, 0.5)$	0.41	0.41	2.53	0.97
Acceleration $(y_{a,t})_y \sim \mathcal{N}(0, 5)$	0.46	0.39	0.88	0.97

for this case. Furthermore, we show that the larger the acceleration, the more accurate the heading estimates will be.

**Example 15: Pose estimation for a sensor with time-varying
linear acceleration**

We estimate the pose of a sensor with an acceleration in the y -direction of $(y_{a,t})_y \sim \mathcal{N}(0, 0.5)$ m/s² using simulated data and obtain smoothing estimates by solving (149). The orientation error for a specific noise realization is depicted in Figure 32c. Furthermore, we simulate data with $(y_{a,t})_y \sim \mathcal{N}(0, 5)$ m/s². The orientation errors based on this data can be found in Figure 32d. As can be seen, for these cases, it is possible to obtain reliable heading estimates using the state space model (86). The larger the acceleration, the more accurate the heading estimates.

In general, it can be concluded that it is possible to estimate orientation from position measurements in the state space model (86). Furthermore, except in the cases of constant or zero acceleration, it is also possible to obtain drift-free orientation estimates. The heading accuracy depends on the amount of acceleration. This is summarized in Table 5 where the mean RMSE of the state estimates over 100 Monte Carlo simulations is shown. Four cases are considered, inspired by Examples 13 – 15.

5 Calibration

In Section 4, we assumed that the sensors were properly calibrated. In practice, however, there are often calibration parameters to be taken into account. Examples of calibration parameters are the inertial sensor biases discussed in Section 2. Furthermore, calibration is specifically of concern when combining the inertial data with other sensors. In these cases, it is important that the inertial sensor axes

and the axes of the additional sensors are aligned. Examples include using inertial sensors in combination with magnetometers (Kok and Schön, 2016; Salehi et al., 2012; Bonnet et al., 2009) and with cameras (Hol et al., 2010; Lobo and Dias, 2007; Mirzaei and Roumeliotis, 2008).

In this section we will introduce several useful calibration methods. In Section 5.1 we explain how calibration parameters can be included as unknowns in the smoothing and filtering algorithms from Section 4. This results in MAP estimates of the parameters. In Section 5.2 we instead focus on obtaining maximum likelihood (ML) estimates of the parameters. In Section 5.3, the workings of the calibration algorithms are illustrated by considering the gyroscope bias to be unknown in the orientation estimation problem. Finally, in Section 5.4, we discuss the topic of *identifiability*. Parameters are said to be identifiable if they can be estimated from the available data.

5.1 Maximum a posteriori calibration

As discussed in Section 3.1, unknown parameters θ can be estimated in the smoothing problem (13) as

$$\{\hat{x}_t, \hat{\theta}\} = \arg \max_{x_t, \theta} p(x_t, \theta \mid y_{1:t}), \quad (151)$$

with

$$p(x_{1:N}, \theta \mid y_{1:N}) \propto p(\theta)p(x_1) \prod_{t=1}^N p(x_t \mid x_{t-1}, \theta)p(y_t \mid x_t, \theta). \quad (152)$$

Recall that a discussion on the choice of the prior of the parameters $p(\theta)$ and the states $p(x_1)$ can be found in Section 3.6.

Within the filtering context we typically model the parameters as slowly time-varying states. These can be estimated by solving

$$\{\hat{x}_t, \hat{\theta}_t\} = \arg \max_{x_t, \theta_t} p(x_t, \theta_t \mid y_{1:t}), \quad (153)$$

where

$$p(x_t, \theta_t \mid y_{1:t}) \propto p(y_t \mid x_t, \theta_t)p(x_t, \theta_t \mid y_{1:t-1}), \quad (154a)$$

and

$$p(x_t, \theta_t \mid y_{1:t-1}) = \iint p(x_t, \theta_t \mid x_{t-1}, \theta_{t-1})p(x_{t-1}, \theta_{t-1} \mid y_{1:t-1})dx_{t-1}d\theta_{t-1}. \quad (154b)$$

Note that compared to Section 3.1, in (154) we do not consider the parameters to be part of x_t but instead represent them explicitly. A prior $p(\theta_1)$ on the parameters at $t = 1$ has to be included as well as a dynamic model of the parameters.

Both the formulations (151) and (153) compute MAP estimates of the parameters. The algorithms presented in Section 4 can straightforwardly be extended to also estimate these unknown parameters θ or $\theta_{1:N}$. This is illustrated in Example 16 for the case of orientation estimation in the presence of an unknown gyroscope bias.

Example 16: MAP estimates of the gyroscope bias

It is possible to estimate an unknown gyroscope bias in the state space model (87). For this, the dynamic model (87a) in the smoothing problem described in Section 4.1 is assumed to include a constant gyroscope bias δ_ω as

$$q_{t+1}^{\text{nb}} = q_t^{\text{nb}} \odot \exp_{\mathbf{q}} \left(\frac{T}{2} (y_{\omega,t} - \delta_\omega - e_{\omega,t}) \right). \quad (155a)$$

In the filtering algorithms in Sections 4.2 and 4.3, the dynamic model is instead assumed to include a slowly time-varying gyroscope bias $\delta_{\omega,t}$ as

$$q_{t+1}^{\text{nb}} = q_t^{\text{nb}} \odot \exp_{\mathbf{q}} \left(\frac{T}{2} (y_{\omega,t} - \delta_{\omega,t} - e_{\omega,t}) \right), \quad (155b)$$

where the dynamics of the gyroscope bias can be described as a random walk (see also Section 3.5)

$$\delta_{\omega,t+1} = \delta_{\omega,t} + e_{\delta_{\omega,t}}, \quad e_{\delta_{\omega,t}} \sim \mathcal{N}(0, \Sigma_{\delta_{\omega,t}}). \quad (155c)$$

The smoothing algorithm presented in Section 4.1 can be extended to also estimate δ_ω . Furthermore, the filtering algorithms presented in Sections 4.2 and 4.3 can be extended to estimate $\delta_{\omega,t}$ for $t = 1, \dots, N$. Only minor changes to the algorithms presented in these sections are needed. These mainly concern including derivatives with respect to the additional unknowns δ_ω or $\delta_{\omega,t}$. Explicit expressions for these can be found in Appendix C.

5.2 Maximum likelihood calibration

Alternatively, it is possible to obtain ML estimates of the parameters θ as

$$\hat{\theta}^{\text{ML}} = \arg \max_{\theta \in \Theta} \mathcal{L}(\theta; y_{1:N}). \quad (156)$$

Here, $\Theta \subseteq \mathbb{R}^{n_\theta}$ and $\mathcal{L}(\theta; y_{1:N})$ is referred to as the likelihood function. It is defined as $\mathcal{L}(\theta; y_{1:N}) \triangleq p_\theta(Y_{1:N} = y_{1:N})$, where $Y_{1:N}$ are random variables and $y_{1:N}$ are a particular realization of $Y_{1:N}$. Using conditional probabilities and the fact that the logarithm is a monotonic function we have the following equivalent formulation of (156),

$$\hat{\theta}^{\text{ML}} = \arg \min_{\theta \in \Theta} - \sum_{t=1}^N \log p_\theta(Y_t = y_t \mid Y_{1:t-1} = y_{1:t-1}), \quad (157)$$

where we use the convention that $y_{1:0} \triangleq \emptyset$. The ML estimator (157) enjoys well-understood theoretical properties including strong consistency, asymptotic normality, and asymptotic efficiency (Ljung, 1999).

Due to the nonlinear nature of the orientation parametrization, our estimation problems are nonlinear, implying that there is no closed form solution available for the one step ahead predictor $p_\theta(Y_t = y_t \mid Y_{1:t-1} = y_{1:t-1})$ in (157). However, similar to the filtering approaches from Section 4, it is possible to approximate the one step ahead predictor according to

$$p_\theta(Y_t = y_t \mid Y_{1:t-1} = y_{1:t-1}) \approx \mathcal{N}(y_t; \hat{y}_{t|t-1}(\theta), S_t(\theta)), \quad (158)$$

where $\hat{y}_{t|t-1}(\theta)$ and $S_t(\theta)$ are defined in (124) and (125), respectively. Inserting (158) into (157) and neglecting all constants not depending on θ results in the following optimization problem,

$$\hat{\theta} = \arg \min_{\theta \in \Theta} \frac{1}{2} \sum_{t=1}^N \|y_t - \hat{y}_{t|t-1}(\theta)\|_{S_t^{-1}(\theta)}^2 + \log \det S_t(\theta). \quad (159)$$

Unlike the optimization problems discussed so far, it is not straightforward to obtain an analytical expression of the gradient of (159). This is because it is defined recursively through the filtering update equations. In Åström (1980) and Segal and Weinstein (1989), different approaches to derive analytical expressions for objective functions of the same type as (159) are provided. They, however, consider the case of a *linear* model. Some methods for obtaining ML estimates of parameters in nonlinear models are explained in the tutorial by Schön et al. (2015).

Instead of deriving analytical expressions for the gradient of (159), it is also possible to compute a numerical approximation of the gradient. Numerical gradients can be used in a number of different optimization algorithms, such as the Broyden-Fletcher-Goldfarb-Shanno (BFGS) method, see e.g. Nocedal and Wright (2006). Similar to the Gauss-Newton method, in the BFGS method, the parameters are iteratively updated until convergence. However, instead of using the Hessian approximation (95), BFGS iteratively estimates the Hessian using information from previous iterations. Hence, solving (159) using BFGS with numerical gradients, requires running at least $n_\theta + 1$ filtering algorithms for each iteration. These are required to evaluate the objective function and to compute the numerical gradients. More evaluations can be necessary to compute a step length, see also Section 4.1.

Example 17: ML estimates of the gyroscope bias

To obtain ML estimates of the gyroscope bias, we run the EKF with orientation deviation states from Algorithm 4 to obtain $\hat{y}_{t|t-1}(\delta_\omega)$ and $S_t(\delta_\omega)$ for a given value of δ_ω . This allows us to evaluate the objective function in (159). To compute $\hat{\delta}_\omega$, the optimization problem (159) is solved iteratively using BFGS.

Table 6: Mean RMSE of the orientation estimates from 100 Monte Carlo simulations in the presence of a gyroscope bias that is being estimated.

RMSE	Roll [°]	Pitch [°]	Heading [°]
Smoothing optimization	0.39	0.39	2.29
EKF orientation deviation	0.46	0.46	4.20

5.3 Orientation estimation with an unknown gyroscope bias

We estimate the gyroscope bias in simulated data as described in Section 4.4 and illustrated in Figure 21. Compared to the data presented in Section 4.4, however, a constant gyroscope bias to be estimated is added. Using Monte Carlo simulations of this data, we illustrate a few specific features of the different ways to estimate the bias.

First, we focus on obtaining MAP estimates of the bias using the smoothing and filtering approaches as described in Section 5.1. We simulate the measurement noise as described in Section 4.4 and simulate the gyroscope bias as

$$\delta_\omega \sim \mathcal{N}(0, \sigma_{\delta_\omega}^2 \mathcal{I}_3), \quad \sigma_{\delta_\omega} = 5 \cdot 10^{-2}. \quad (160)$$

Note that σ_{δ_ω} is a factor 10 larger than the value discussed in Section 3.6 to clearly illustrate the effect of the presence of a gyroscope bias. The priors $p(\theta)$ and $p(\theta_1)$ in the smoothing and filtering algorithms are set equal to the distribution in (160). The covariance of the random walk model (155c) is set as $\Sigma_{\delta_{\omega,t}} = \sigma_{\delta_{\omega,t}}^2 \mathcal{I}_3$ with $\sigma_{\delta_{\omega,t}} = 1 \cdot 10^{-10}$. This small value ensures that after convergence, the bias estimate is quite constant. The resulting mean RMSEs of the orientation over 100 Monte Carlo simulations are summarized in Table 6. Since the filtering algorithms typically have similar characteristics as discussed in Section 4.4, we only consider the EKF with orientation deviation states here. Comparing these results to the ones presented in Table 3, the RMSEs of the smoothing optimization algorithm are almost the same as when there was no gyroscope bias present. However, the filtering results are worse. This is because the bias needs some time to be properly estimated. This is illustrated in Figure 33 where the gyroscope bias estimates and their uncertainties are shown for the filtering algorithm.

A major difference between the MAP and the ML approaches, is that the MAP takes into account a prior on the gyroscope bias. We analyze the effect of this prior using 500 Monte Carlo simulations, simulating the gyroscope bias to be $\delta_\omega = (0.05 \ 0.01 \ -0.04)^\top$ rad/s. We study the estimated gyroscope biases using ML estimation, and using MAP estimation by including the gyroscope bias as an unknown in the smoothing algorithm. The smoothing algorithm assumes two different priors on the gyroscope bias $\delta_\omega \sim \mathcal{N}(0, \sigma_{\delta_\omega}^2 \mathcal{I}_3)$. In the first case, the prior on the gyroscope bias can well describe the data ($\sigma_{\delta_\omega} = 0.05$). In the other case, the prior is too tight ($\sigma_{\delta_\omega} = 1 \cdot 10^{-3}$). The mean and standard deviations for the

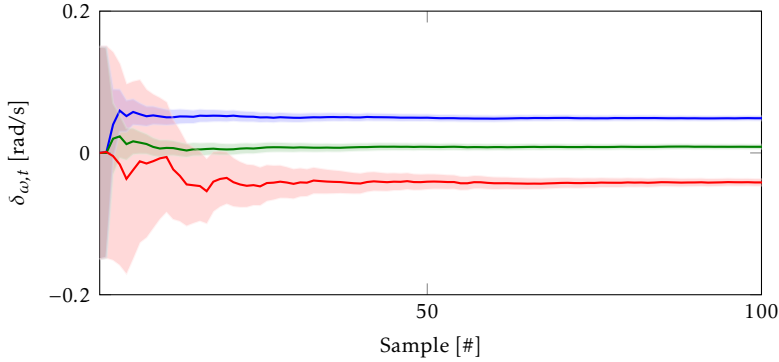


Figure 33: Filtering estimates of the gyroscope bias and their uncertainties with the x -, y - and z -components in blue, green and red, respectively. Note that the estimates for only the first 100 samples are shown, to focus on the period in which the estimates converge.

gyroscope bias estimates are summarized in Table 7. As can be seen, when the prior is chosen appropriately, the ML and MAP estimates are comparable. If the prior is too tight, the MAP estimates can be seen to be biased towards zero.

Table 7: Mean and standard deviation of the gyroscope estimates over 500 Monte Carlo simulations with $(0.05 \ 0.01 \ -0.04)^T$ rad/s. Considered are the cases of ML estimation and MAP estimation by including the gyroscope bias as an unknown in a smoothing algorithm with a prior on the gyroscope bias of $\delta_\omega \sim \mathcal{N}(0, \sigma_{\delta_\omega}^2 \mathcal{I}_3)$.

RMSE	Mean $\hat{\delta}_\omega (\cdot 10^2)$			Standard deviation $\hat{\delta}_\omega (\cdot 10^4)$		
	x	y	z	x	y	z
ML	5.0	1.0	-4.0	5.1	5.3	6.4
MAP $\sigma_{\delta_\omega} = 0.05$	5.0	1.0	-4.0	5.1	5.3	6.4
MAP $\sigma_{\delta_\omega} = 1 \cdot 10^{-3}$	3.9	0.8	-2.8	4.1	4.0	4.7

5.4 Identifiability

Parameters are said to be *identifiable* if it is possible to determine a unique parameter value from the data and if this value is equal to the true value (Ljung, 1999). The concept of identifiability is closely related to the concept of *observability* which is concerned with the question of if the time-varying states can be determined from the available data (Kailath, 1980). The states discussed in Section 4 are typically observable. Identifiability, however, becomes of concern

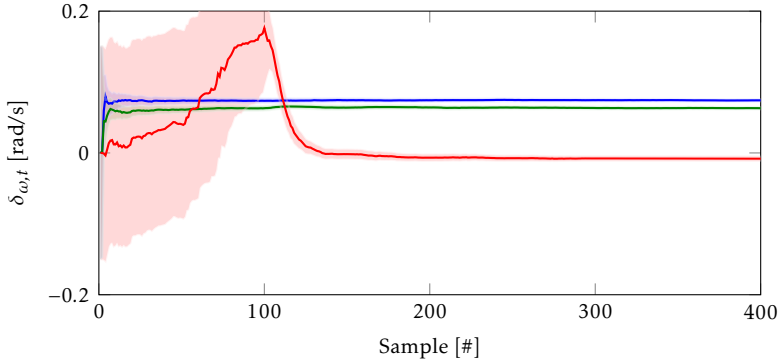


Figure 34: Filtering estimates of the gyroscope bias and their uncertainties with the x -, y - and z -components in blue, green and red, respectively. The estimates are obtained using an EKF and inertial (but no magnetometer) measurements. As can be seen, the estimates of the gyroscope bias in the x - and y -axes converge quickly while the estimates of the gyroscope bias in the z -axis only start to converge after 100 samples, when the sensor starts to rotate around the x -axis.

when estimating calibration parameters. Specifically, in many applications, certain parameters are not identifiable when the sensor is completely stationary and sufficient excitation in terms of change in position and orientation is needed to make the parameters identifiable. This is illustrated in Example 18 for the case of identifiability of the gyroscope bias.

Example 18: Identifiability of the gyroscope bias

We consider the example of orientation estimation using only inertial measurements in the presence of a gyroscope bias. We simulate data as described in Section 4.4. The filtering estimates of the gyroscope bias and their uncertainties from an EKF with orientation deviation states is shown in Figure 34. Using only inertial measurements, the gyroscope bias of the sensor's z -axis is not identifiable when the sensor is placed horizontally. However, when the sensor is rotated, the accelerometer provides information that aids the estimation and the bias can be seen to converge. Note the difference with Figure 33, where only the first 100 samples were displayed and the bias estimates in the z -axis converged significantly faster due to the inclusion of magnetometer measurements.

6 Concluding remarks

The goal of this tutorial was not to give a complete overview of all algorithms that can be used for position and orientation estimation. Instead, our aim was

to give a pedagogical introduction to the topic of position and orientation estimation using inertial sensors. By integrating the inertial sensor measurements (so-called dead-reckoning), it is possible to obtain information about the position and orientation of the sensor. However, errors in the measurements will accumulate and the estimates will drift. Because of this, to obtain accurate position and orientation estimates using inertial measurements, it is necessary to use additional sensors and/or additional models. In this tutorial, we have considered two separate estimation problems. The first is orientation estimation using inertial and magnetometer measurements, assuming that the acceleration of the sensor is approximately zero. Including magnetometer measurements removes the drift in the heading direction (as illustrated in Example 9), while assuming that the acceleration is approximately zero removes the drift in the inclination. The second estimation problem that we have considered is pose estimation using inertial and position measurements. Using inertial measurements, the position and orientation estimates are coupled and the position measurements therefore provide information also about the orientation.

A number of algorithms for position and orientation estimation have been introduced in Section 4. These include smoothing and filtering approaches solved as an optimization problem or via extended Kalman filtering, using either unit quaternion or orientation deviations from a linearization point as states. The filtering approaches use the data up to a certain time t to estimate the position and orientation at this time t . Smoothing instead makes use of all data from time $t = 1, \dots, N$. In general, using all data to obtain the estimates will naturally lead to better estimates. The filtering approaches can be seen to be quite uncertain about their estimates for the first samples and require some time to “converge” to accurate estimates. This is even more pronounced in the presence of calibration parameters as discussed in Section 5. Although smoothing algorithms give better estimates, practical applications might not allow for computing smoothing estimates because of computational limitations or real-time requirements. For the examples discussed in this paper, the optimization-based filtering algorithm and the EKF with orientation deviation states perform very similarly. The EKF with quaternion states, however, was able to handle wrong initial orientations less well as shown in Table 4. Furthermore, it underestimated the uncertainty in the heading direction in the absence of magnetometer measurements, see also Example 12.

Some benefits of the optimization-based approaches which have not directly been considered in this work, are that they can easily include models outside of the standard state space model. An example where we make use of this is our work on inertial sensor motion capture (Kok et al., 2014). Here, multiple IMUs are placed on different body segments of the human body. We assume that the body segments are attached to each other by including a number of equality constraints in the estimation problem. Including these constraints in an optimization framework is straightforward. Furthermore, optimization-based approaches allow for natural inclusion of non-Gaussian noise. In Kok et al. (2015) we combine inertial measurements with time of arrival measurements from a UWB system. To allow for delayed measurements due to multipath and non-line-of-sight, we model the

time of arrival measurements using a tailored heavy-tailed asymmetric distribution. This distribution can straightforwardly be used when solving the estimation problem using an optimization-based approach.

Apart from the differences between the estimation algorithms discussed in this tutorial, it can also be concluded that the position and orientation estimation problems using inertial sensors are actually quite forgiving. Any of the algorithms discussed in this tutorial can give reasonably good estimates with fairly little effort. However, careful *modeling* is important since the quality of the estimates of the estimation algorithms highly depends on the validity of the models, see for instance Example 10. In recent years, inertial sensors have undergone major developments. The quality of their measurements has improved while their cost has decreased, leading to an increase in availability. Furthermore, the computational resources are steadily increasing. Because of these reasons, we believe that inertial sensors can be used for even more diverse applications in the future.

Acknowledgements

This work is supported by CADICS, a Linnaeus Center, and by the project *Probabilistic modeling of dynamical systems* (Contract number: 621-2013-5524), both funded by the Swedish Research Council (VR). It is also supported by the Swedish Foundation for Strategic Research (SSF) via the project *ASSEMBLE*. High accuracy reference measurements are provided through the use of the Vicon real-time tracking system courtesy of the UAS Technologies Lab, Artificial Intelligence and Integrated Computer Systems Division (AIICS) at the Department of Computer and Information Science (IDA), Linköping University, Sweden <http://www.ida.liu.se/divisions/aiics/aiicssite/index.en.shtml>. The authors would also like to thank Fredrik Olsson, Michael Lorenz and Dr. Oliver Woodford for comments and suggestions that greatly improved this tutorial.

Appendix

A Orientation parametrizations

In Section A.1 of this appendix, we will summarize some important results on quaternion algebra that we make frequent use of throughout this tutorial. In Section A.2, we summarize some results on how to convert between the different orientation parametrizations.

A.1 Quaternion algebra

A quaternion is a 4-dimensional vector q ,

$$q = \begin{pmatrix} q_0 & q_v^T \end{pmatrix}^T = \begin{pmatrix} q_0 & q_1 & q_2 & q_3 \end{pmatrix}^T. \quad (161)$$

A special case is a unit quaternion, which has the property that $\|q\|_2 = 1$. We use unit quaternions as a parametrization of orientations. An example of a quaternion that is not a unit quaternion and that we frequently encounter in this tutorial is the quaternion representation of a vector. For a vector v , its quaternion representation is given by

$$\bar{v} = \begin{pmatrix} 0 \\ v \end{pmatrix}. \quad (162)$$

The rotation of a vector v by a unit quaternion q can be written as

$$q \odot \bar{v} \odot q^c. \quad (163)$$

Here, the quaternion multiplication $\odot$ of two quaternions p and q is defined as

$$p \odot q = \begin{pmatrix} p_0 q_0 - p_v \cdot q_v \\ p_0 q_v + q_0 p_v + p_v \times q_v \end{pmatrix}. \quad (164)$$

This can alternatively be defined in terms of the left and right multiplication matrices

$$p \odot q = p^L q = q^R p, \quad (165)$$

with

$$p^L \triangleq \begin{pmatrix} p_0 & -p_v^T \\ p_v & p_0 \mathcal{I}_3 + [p_v \times] \end{pmatrix}, \quad q^R \triangleq \begin{pmatrix} q_0 & -q_v^T \\ q_v & q_0 \mathcal{I}_3 - [q_v \times] \end{pmatrix}, \quad (166)$$

where $[q_v \times]$ denotes the cross product matrix

$$[q_v \times] = \begin{pmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_2 & q_1 & 0 \end{pmatrix}. \quad (167)$$

The quaternion conjugate is given by

$$q^c = \begin{pmatrix} q_0 \\ -q_v \end{pmatrix}. \quad (168)$$

Hence, the rotation of the vector v is given by

$$\begin{aligned} q \odot \bar{v} \odot q^c &= q^L (q^c)^R \bar{v} \\ &= \begin{pmatrix} q_0 & -q_v^T \\ q_v & q_0 \mathcal{I}_3 + [q_v \times] \end{pmatrix} \begin{pmatrix} q_0 & -q_v^T \\ q_v & q_0 \mathcal{I}_3 - [q_v \times] \end{pmatrix} \begin{pmatrix} 0 \\ v \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0_{1 \times 3} \\ 0_{3 \times 1} & q_v q_v^T + q_0^2 \mathcal{I}_3 + 2q_0 [q_v \times] + [q_v \times]^2 \end{pmatrix} \begin{pmatrix} 0 \\ v \end{pmatrix}. \end{aligned} \quad (169)$$

A.2 Conversions between different parametrizations

A quaternion q can be converted into a rotation matrix R as

$$\begin{aligned} R &= q_v q_v^T + q_0^2 \mathcal{I}_3 + 2q_0[q_v \times] + [q_v \times]^2 \\ &= \begin{pmatrix} 2q_0^2 + 2q_1^2 - 1 & 2q_1q_2 - 2q_0q_3 & 2q_1q_3 + 2q_0q_2 \\ 2q_1q_2 + 2q_0q_3 & 2q_0^2 + 2q_2^2 - 1 & 2q_2q_3 - 2q_0q_1 \\ 2q_1q_3 - 2q_0q_2 & 2q_2q_3 + 2q_0q_1 & 2q_0^2 + 2q_3^2 - 1 \end{pmatrix}. \end{aligned} \quad (170)$$

Note the similarity with (169), where the rotation of the vector v can equivalently be expressed as Rv . Conversely, a rotation matrix

$$R = \begin{pmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{pmatrix}, \quad (171)$$

can be converted into a quaternion as

$$q_0 = \frac{\sqrt{1 + \text{Tr } R}}{2}, \quad q_v = \frac{1}{4q_0} \begin{pmatrix} R_{32} - R_{23} \\ R_{13} - R_{31} \\ R_{21} - R_{12} \end{pmatrix}. \quad (172)$$

Note that a practical implementation needs to take care of the fact that the conversion (172) only leads to sensible results if $1 + \text{Tr } R > 0$ and $q_0 \neq 0$. To resolve this issue, the conversion is typically performed in different ways depending on the trace of the matrix R and its diagonal values, see e.g. Baker (2016).

A rotation vector η can be expressed in terms of a unit quaternion q via the quaternion exponential as

$$q = \exp_q \eta = \begin{pmatrix} \cos \|\eta\|_2 \\ \frac{\eta}{\|\eta\|_2} \sin \|\eta\|_2 \end{pmatrix}. \quad (173)$$

Note that any practical implementation needs to take care of the fact that this equation is singular at $\eta = 0$, in which case $\exp_q \eta = \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}^T$. The inverse operation is executed by the quaternion logarithm,

$$\eta = \log_q q = \frac{\arccos q_0}{\sin \arccos q_0} q_v = \frac{\arccos q_0}{\|q_v\|_2} q_v. \quad (174)$$

Note that this equation is singular at $q_v = 0$. In this case, $\log q$ should return $0_{3 \times 1}$.

The rotation vector η can also be converted into a rotation matrix as

$$R = \exp_R \eta = \exp([\eta \times]), \quad \eta = \log_R R = \begin{pmatrix} (\log R)_{32} \\ (\log R)_{13} \\ (\log R)_{21} \end{pmatrix}, \quad (175)$$

where $\log R$ is the matrix logarithm and $\log_R$ and $\exp_R$ are the mappings introduced in (49) and (51).

A rotation in terms of Euler angles can be expressed as a rotation matrix R as

$$R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{pmatrix} \begin{pmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{pmatrix} \begin{pmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \sin \phi \cos \theta \\ \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi & \cos \phi \cos \theta \end{pmatrix}. \quad (176)$$

The rotation matrix R can be converted into Euler angles as

$$\psi = \tan^{-1} \left(\frac{R_{12}}{R_{11}} \right) = \tan^{-1} \left(\frac{2q_1 q_2 - 2q_0 q_3}{2q_0^2 + 2q_1^2 - 1} \right), \quad (177a)$$

$$\phi = -\sin^{-1} (R_{13}) = \sin^{-1} (2q_1 q_3 + 2q_0 q_2), \quad (177b)$$

$$\theta = \tan^{-1} \left(\frac{R_{23}}{R_{33}} \right) = \tan^{-1} \left(\frac{2q_2 q_3 - 2q_0 q_1}{2q_0^2 + 2q_3^2 - 1} \right). \quad (177c)$$

Using the relations presented in this section, it is possible to convert the parametrizations discussed in Section 3.2 into each other.

B Pose estimation

In this appendix, we will introduce the necessary components to extend Algorithms 1 – 4 to pose estimation algorithms using the state space model (86).

B.1 Smoothing in an optimization framework

In Section 4.5, we presented the smoothing optimization problem for pose estimation. To adapt Algorithm 1, the derivatives (101a) – (101c) in combination with the following derivatives are needed

$$\frac{de_{p,i}}{dp_1^n} = \mathcal{I}_3, \quad \frac{de_{v,i}}{dv_1^n} = \mathcal{I}_3, \quad \frac{de_{p,t}}{dp_t^n} = -\mathcal{I}_3, \quad (178a)$$

$$\frac{de_{p,a,t}}{dp_{t+1}^n} = \frac{2}{T^2} \mathcal{I}_3, \quad \frac{de_{p,a,t}}{dp_t^n} = -\frac{2}{T^2} \mathcal{I}_3, \quad \frac{de_{p,a,t}}{dv_t^n} = -\frac{1}{T} \mathcal{I}_3, \quad \frac{de_{p,a,t}}{d\eta_t^b} = \tilde{R}_t^{\text{bn}}[y_{a,t} \times], \quad (178b)$$

$$\frac{de_{v,a,t}}{dv_{t+1}^n} = \frac{1}{T} \mathcal{I}_3, \quad \frac{de_{v,a,t}}{dv_t^n} = -\frac{1}{T} \mathcal{I}_3, \quad \frac{de_{v,a,t}}{d\eta_t^b} = \tilde{R}_t^{\text{bn}}[y_{a,t} \times]. \quad (178c)$$

B.2 Filtering in an optimization framework

To obtain position, velocity and orientation estimates in a filtering framework, the optimization problem (115) is adapted to

$$\hat{x}_t = \arg \min_{x_t} -\log p(x_t | y_{1:t}) = \arg \min_{x_t} \|e_{f,t}\|_{P_{t|t-1}^{-1}} + \|e_{p,t}\|_{\Sigma_p^{-1}}, \quad (179)$$

where $\|e_{p,t}\|_{\Sigma_p^{-1}}$ is the position measurement model also used in the smoothing optimization problem presented in Section 4.5. Furthermore, $e_{f,t}$ is extended

from (114) to be

$$e_{f,t} = \begin{pmatrix} p_t^n - p_{t-1}^n - T v_{t-1}^n - \frac{T^2}{2} (R_t^{\text{nb}} y_{a,t-1} + g^n) \\ v_t^n - v_{t-1}^n - T (R_t^{\text{nb}} y_{a,t-1} + g^n) \\ -2 \log_q(\tilde{q}_t^{\text{bn}} \odot \tilde{q}_{t-1}^{\text{nb}} \odot \exp_q(\frac{T}{2}(y_{\omega,t-1}))) \end{pmatrix}, \quad \frac{de_{f,t}}{d\eta_t^b} = \mathcal{I}_9. \quad (180)$$

The covariance matrix $P_{t|t-1}^{-1}$ is again defined as $P_{t+1|t}^{(0)} = F_t P_{t|t} F_t^\top + G_t Q G_t^\top$ with

$$F_t = \begin{pmatrix} \mathcal{I}_3 & T\mathcal{I}_3 & -\frac{T^2}{2} \tilde{R}_t^{\text{nb}}[y_{a,t} \times] \\ 0 & \mathcal{I}_3 & -T \tilde{R}_t^{\text{nb}}[y_{a,t} \times] \\ 0 & 0 & \exp_R(-T y_{\omega,t}) \end{pmatrix}, \quad G_t = \begin{pmatrix} \mathcal{I}_6 & 0 \\ 0 & T\mathcal{I}_3 \end{pmatrix}, \quad (181a)$$

$$Q = \begin{pmatrix} \Sigma_a & 0 & 0 \\ 0 & \Sigma_a & 0 \\ 0 & 0 & \Sigma_\omega \end{pmatrix}. \quad (181b)$$

Similar to the update of the linearization point in (117a), we also update the estimates of the position and velocity before starting the optimization algorithm, such that $e_{f,t}$ is equal to zero for iteration $k = 0$.

B.3 EKF with quaternion states

Following the notation in Section 4.3, the following matrices are needed for implementing the EKF for pose estimation with quaternion states

$$F_t = \begin{pmatrix} \mathcal{I}_3 & T\mathcal{I}_3 & \frac{T^2}{2} \frac{\partial R_{t|t}^{\text{nb}}}{\partial q_{t|t}^{\text{nb}}} \Big|_{q_{t|t}^{\text{nb}} = \hat{q}_{t|t}^{\text{nb}}} y_{a,t} \\ 0 & \mathcal{I}_3 & T \frac{\partial R_{t|t}^{\text{nb}}}{\partial q_{t|t}^{\text{nb}}} \Big|_{q_{t|t}^{\text{nb}} = \hat{q}_{t|t}^{\text{nb}}} y_{a,t} \\ 0 & 0 & (\exp_q(\frac{T}{2} y_{\omega,t}))^R \end{pmatrix}, \quad G_t = \begin{pmatrix} \mathcal{I}_6 & 0 \\ 0 & -\frac{T}{2} (\hat{q}_{t|t}^{\text{nb}})^L \frac{d \exp_q(e_{\omega,t})}{d e_{\omega,t}} \end{pmatrix}, \quad (182a)$$

$$Q = \begin{pmatrix} \Sigma_a & 0 & 0 \\ 0 & \Sigma_a & 0 \\ 0 & 0 & \Sigma_\omega \end{pmatrix}, \quad H = \mathcal{I}_3, \quad R = \Sigma_p. \quad (182b)$$

B.4 EKF with orientation deviation states

In the time update of the pose estimation algorithm with orientation deviation states, the linearization point is again directly updated as in (134). The position and velocity states are updated according to the dynamic model (86a). Furthermore, the matrices Q , H and R from (182b) are needed for implementing the EKF for pose estimation in combination with the matrices F_t and G_t from (181a).

C Gyroscope bias estimation

In this appendix, we will introduce the necessary components to extend Algorithms 1 – 4 to estimate an unknown gyroscope bias. Note that the measurement models are independent of the gyroscope bias. The dynamic models and time update equations are adapted as described below.

C.1 Smoothing in an optimization framework

To include a gyroscope bias in Algorithm 1, a prior on the bias is included, leading to an additional term in the objective function as

$$e_{\delta_\omega} = \delta_\omega, \quad e_{\delta_\omega} \sim \mathcal{N}(0, \Sigma_{\delta_\omega}). \quad (183)$$

The derivatives in (101) are complemented with

$$\frac{de_{\omega,t}}{d\delta_\omega} = \mathcal{I}_3, \quad \frac{de_{\delta_\omega}}{d\delta_\omega} = \mathcal{I}_3. \quad (184)$$

C.2 Filtering in an optimization framework

To include estimation of an unknown gyroscope bias in Algorithm 2, (110) is extended to include a random walk for the gyroscope bias. The matrices F_t , G_t and Q in (112) then become

$$F_t = \begin{pmatrix} \exp_R(-T(y_{\omega,t} - \hat{\delta}_{\omega,t|t})) & -T\mathcal{I}_3 \\ 0 & \mathcal{I}_3 \end{pmatrix}, \quad (185a)$$

$$G = \begin{pmatrix} T\mathcal{I}_3 & 0 \\ 0 & \mathcal{I}_3 \end{pmatrix}, \quad Q = \begin{pmatrix} \Sigma_\omega & 0 \\ 0 & \Sigma_{\delta_\omega} \end{pmatrix}. \quad (185b)$$

Note that a prior on the gyroscope bias at $t = 1$ needs to be included and that the relinearization of the covariance in (116) needs to be adjusted to the size of the covariance matrix. Because of this, the matrix J_t in (116) is extended to

$$J_t = \begin{pmatrix} \exp_R(\eta_t^{b,(k+1)}) & 0 \\ 0 & \mathcal{I}_3 \end{pmatrix}. \quad (186)$$

C.3 EKF with quaternion states

To include estimation of an unknown gyroscope bias in Algorithm 3, the matrices for the time update of the EKF need to be adapted to

$$F_t = \begin{pmatrix} \left(\exp_q\left(\frac{T}{2}(y_{\omega,t} - \hat{\delta}_{\omega,t|t})\right)\right)^R & -\frac{T}{2}(\hat{q}_{t|t}^{\text{nb}})^L \frac{d\exp_q(\delta_{\omega,t})}{d\delta_{\omega,t}} \\ 0_{3 \times 4} & \mathcal{I}_3 \end{pmatrix}, \quad (187a)$$

$$Q = \begin{pmatrix} \Sigma_\omega & 0 \\ 0 & \Sigma_{\delta_\omega} \end{pmatrix}, \quad G_t = \begin{pmatrix} -\frac{T}{2}(\hat{q}_{t|t}^{\text{nb}})^L \frac{d\exp_q(e_{\omega,t})}{de_{\omega,t}} & 0 \\ 0 & \mathcal{I}_3 \end{pmatrix}. \quad (187b)$$

Note that also for this algorithm a prior on the gyroscope bias needs to be included. Furthermore, the renormalization of the covariance in (130) needs to be adjusted to the size of the covariance matrix in a similar manner as in (186).

C.4 EKF with orientation deviation states

To include estimation of an unknown gyroscope bias in Algorithm 4, the update of the linearization point in the time update takes the estimate of the gyroscope bias into account as

$$\tilde{q}_{t+1}^{\text{nb}} = \tilde{q}_t^{\text{nb}} \odot \exp_{\mathbf{q}} \left(\frac{T}{2} (\mathcal{Y}_{\omega,t} - \hat{\delta}_{\omega,t|t}) \right). \quad (188)$$

The matrices F_t , G_t and Q for the time update are adapted to the expressions given in (185). A prior on the gyroscope bias needs to be included and the relinearization of the covariance in (140) needs to be adapted according to (186).

Bibliography

- S. Adler, S. Schmitt, K. Wolter, and M. Kyas. A survey of experimental evaluation in indoor localization research. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–10, Banff, Alberta, Canada, 2015.
- D. W. Allan. Statistics of atomic frequency standards. *Proceedings of the IEEE*, 54(2):221–230, 1966.
- K. J. Åström. Maximum likelihood and prediction error methods. *Automatica*, 16(5):551–574, 1980.
- M. J. Baker. EuclideanSpace – Mathematics and computing. <http://www.euclideanspace.com/maths/geometry/rotations/conversions/matrixToQuaternion/>, 2016. Accessed on October 4.
- N. Barbour and G. Schmidt. Inertial sensor technology trends. *IEEE Sensors Journal*, 1(4):332–339, 2001.
- T. D. Barfoot. *State Estimation for Robotics: A Matrix Lie Group Approach*. Draft in preparation for publication by Cambridge University Press, 2016.
- B. M. Bell. The iterated Kalman smoother as a Gauss-Newton method. *SIAM Journal on Optimization*, 4(3):626–636, 1994.
- D. P. Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena Scientific Belmont, MA, 3rd edition, 1995.
- C. M. Bishop. *Pattern recognition and machine learning*. Springer Science + Business Media, 2006.
- M. Bloesch, H. Sommer, T. Laidlow, M. Burri, G. Nuetzi, P. Fankhauser, D. Bellicoso, C. Gehring, S. Leutenegger, M. Hutter, and R. Siegwart. A primer on the differential calculus of 3D orientations. *ArXiv e-prints*, June 2016. arXiv:1606.05285.
- S. Bonnet, C. Bassompierre, C. Godin, S. Lesecq, and A. Barraud. Calibration methods for inertial and magnetic sensors. *Sensors and Actuators A: Physical*, 156(2):302–311, 2009.
- S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- J. Callmer. *Autonomous Localization in Unknown Environments*. Dissertation no. 1520, Linköping University, Linköping, Sweden, June 2013.
- H. Chao, Y. Cao, and Y. Chen. Autopilots for small unmanned aerial vehicles: A survey. *International Journal of Control, Automation and Systems*, 8(1):36–44, 2010.

- P. Corke, J. Lobo, and J. Dias. An introduction to inertial and visual sensing. *The International Journal of Robotics Research*, 26(6):519–535, 2007.
- J. A. Corrales, F. A. Candelas, and F. Torres. Hybrid tracking of human operators using IMU/UWB data fusion by a Kalman filter. In *Proceedings of the 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 193–200, 2008.
- J. L. Crassidis, F. L. Markley, and Y. Cheng. A survey of nonlinear attitude estimation methods. *Journal of Guidance, Control, and Dynamics*, 30(1):12–28, 2007.
- T. A. Davis. *Direct methods for sparse linear systems*. SIAM, 2006.
- A. De Angelis, J. O. Nilsson, I. Skog, P. Händel, and P. Carbone. Indoor positioning by ultrawide band radio aided inertial navigation. *Metrology and Measurement Systems*, 17(3):447–460, 2010.
- J. Diebel. Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Stanford University, Stanford, California, Tech. Rep. 94301?9010*, 2006.
- 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, 2008.
- L. Euler. Formulae generales pro translatione quacunque corporum rigidorum. *Novi Commentarii Academiae Scientiarum Imperialis Petropolitanae*, 20:189–207, 1776.
- C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza. On-manifold preintegration for real-time visual-inertial odometry. *IEEE Transactions on Robotics*, 99(PP):1–21, 2016.
- E. Foxlin. Inertial head-tracker sensor fusion by a complementary separate-bias Kalman filter. In *Proceedings of the IEEE Virtual Reality Annual International Symposium*, pages 185–194, Santa Clara, CA, USA, 1996.
- I. Gilitschenski, G. Kurz, S. J. Julier, and U. D. Hanebeck. Unscented orientation estimation based on the Bingham distribution. *IEEE Transactions on Automatic Control*, 61(1):172–177, 2016.
- J. Glover and L. P. Kaelbling. Tracking 3-D rotations with the quaternion Bingham filter. *Computer Science and Artificial Intelligence Laboratory Technical Report*, 2013.
- G. H. Golub and C. F. van Loan. *Matrix Computations*. The John Hopkins University Press, Baltimore, 4th edition, 2013.
- G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard. A tutorial on graph-based SLAM. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43, 2010a.

- G. Grisetti, R. Kümmerle, C. Stachniss, U. Frese, and C. Hertzberg. Hierarchical optimization on manifolds for online 2D and 3D mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 273–278, Anchorage, Alaska, 2010b.
- F. Gustafsson. *Statistical Sensor Fusion*. Studentlitteratur, 2012.
- F. Gustafsson and F. Gunnarsson. Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements. *IEEE Signal Processing Magazine*, 22(4):41–53, 2005.
- W. R. Hamilton. On quaternions; or on a new system of imaginaries in algebra. *Philosophical Magazine*, xxv, 1844.
- R. Harle. A survey of indoor inertial positioning systems for pedestrians. *IEEE Communications Surveys & Tutorials*, 15(3):1281–1293, 2013.
- J. Haverinen and A. Kemppainen. Global indoor self-localization based on the ambient magnetic field. *Robotics and Autonomous Systems*, 57(10):1028–1035, 2009.
- G. Hendeby, F. Gustafsson, and N. Wahlström. Teaching sensor fusion and Kalman filtering using a smartphone. In *Proceedings of the 19th World Congress of the International Federation of Automatic Control*, pages 10586–10591, Cape Town, South Africa, August 2014.
- J. D. Hol. *Sensor fusion and calibration of inertial sensors, vision, ultra-wideband and GPS*. Dissertation no. 1368, Linköping University, Linköping, Sweden, June 2011.
- 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(2-3):149–160, 2007.
- J. D. Hol, T. B. Schön, and F. Gustafsson. Modeling and calibration of inertial and vision sensors. *The international journal of robotics research*, 29(2-3):231–244, 2010.
- 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, 1987.
- IEEE. IEEE standard for inertial systems terminology, 2009. Technical report, IEEE Std 1559.
- InvenSense. <http://www.invensense.com>, 2016. Accessed on November 6.
- A. H. Jazwinski. *Stochastic processes and filtering theory*. Academic Press, New York and London, 1970.
- T. A. Johansen. Introduction to nonlinear model predictive control and moving horizon estimation. In *Selected Topics on Constrained and Nonlinear Control*, pages 187–239. STU Bratislava and NTNU Trondheim, 2011.

- S. J. Julier and J. J. LaViola Jr. On Kalman filtering with nonlinear equality constraints. *IEEE Transactions on Signal Processing*, 55(6):2774–2784, 2007.
- T. Kailath. *Linear systems*. Prentice-Hall Englewood Cliffs, NJ, 1980.
- E. D. Kaplan and C. J. Hegarty. *Understanding GPS: principles and applications*. Artech House, 1996.
- M. Kok and T. B. Schön. Magnetometer calibration using inertial sensors. *IEEE Sensors Journal*, 16(14):5679 – 5689, 2016.
- M. Kok, J. D. Hol, and T. B. Schön. An optimization-based approach to human body motion capture using inertial sensors. In *Proceedings of the 19th World Congress of the International Federation of Automatic Control*, pages 79–85, Cape Town, South Africa, August 2014.
- M. Kok, J. D. Hol, and T. B. Schön. Indoor positioning using ultrawideband and inertial measurements. *IEEE Transactions on Vehicular Technology*, 64(4):1293–1303, 2015.
- M. Kok, J. D. Hol, and T. B. Schön. Using inertial sensors for position and orientation estimation. Technical Report LiTH-ISY-R-3093, Department of Electrical Engineering, Linköping University, Linköping, Sweden, December 2016.
- J. B. Kuipers. *Quaternions and Rotation Sequences: a primer with applications to orbits, aerospace, and virtual reality*. Princeton University Press, 1999.
- G. Kurz, I. Gilitschenski, S. Julier, and U. D. Hanebeck. Recursive estimation of orientation based on the Bingham distribution. In *Proceedings of the 16th International Conference on Information Fusion*, pages 1487–1494, Istanbul, Turkey, July 2013.
- M. Li and A. I. Mourikis. High-precision, consistent EKF-based visual-inertial odometry. *The International Journal of Robotics Research*, 32(6):690–711, 2013.
- G. Ligorio and A. M. Sabatini. Dealing with magnetic disturbances in human motion capture: A survey of techniques. *Micromachines*, 7(3):43, 2016.
- F. Lindsten and T. B. Schön. Backward simulation methods for Monte Carlo statistical inference. *Foundations and Trends in Machine Learning*, 6(1):1–143, 2013.
- L. Ljung. *System Identification, Theory for the User*. Prentice Hall PTR, 2nd edition, 1999.
- J. Lobo and J. Dias. Relative pose calibration between visual and inertial sensors. *The International Journal of Robotics Research*, 26(6):561–575, 2007.
- H. J. Luinge. *Inertial Sensing of Human Movement*. PhD thesis, University of Twente, Enschede, the Netherlands, October 2002.

- H. J. Luinge and P. H. Veltink. Measuring orientation of human body segments using miniature gyroscopes and accelerometers. *Medical and Biological Engineering and computing*, 43(2):273–282, 2005.
- S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan. Estimation of IMU and MARG orientation using a gradient descent algorithm. In *IEEE International Conference on Rehabilitation Robotics*, pages 1–7, Zürich, Switzerland, Jun – Jul 2011.
- J. L. Marins, X. Yun, E. R. Bachmann, R. B. McGhee, and M. J. Zyda. An extended Kalman filter for quaternion-based orientation estimation using MARG sensors. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 4, pages 2003–2011, Maui, Hawaii, USA, 2001.
- J. B. Marion and S. T. Thornton. *Classical dynamics of particles and systems*. Harcourt College Publishers, 4 edition, 1995.
- F. L. Markley. Attitude error representations for Kalman filtering. *Journal of guidance, control, and dynamics*, 26(2):311–317, 2003.
- A. Martinelli. Vision and IMU data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination. *IEEE Transactions on Robotics*, 28(1):44–60, 2012.
- J. Mattingley and S. Boyd. Real-time convex optimization in signal processing. *IEEE Signal Processing Magazine*, 27(3):50–61, 2010.
- J. L. Meriam and L. G. Kraige. *Engineering mechanics: dynamics*. John Wiley & Sons, 4 edition, 1998.
- 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, 2008.
- National Aeronautics and Space Administration. Earth fact sheet. <http://nssdc.gsfc.nasa.gov/planetary/factsheet/earthfact.html>, 2016. Accessed on September 7.
- National Centers for Environmental Information. <https://www.ngdc.noaa.gov/geomag/geomag.shtml>, 2016. Accessed on October 10.
- J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research, 2nd edition, 2006.
- H. Nurminen, T. Ardeschiri, R. Piché, and F. Gustafsson. Robust inference for state-space models with skewed measurement noise. *IEEE Signal Processing Letters*, 22(11):1898–1902, 2015.
- B. Palais, R. Palais, and S. Rodi. A disorienting look at Euler’s theorem on the axis of a rotation. *American Mathematical Monthly*, 116(10):892–909, 2009.

- M. Perlmutter and L. Robin. High-performance, low cost inertial MEMS: A market in motion! In *Proceedings of the IEEE/ION Position Location and Navigation Symposium (PLANS)*, pages 225–229, Myrtle Beach, South Carolina, USA, 2012.
- S. Pittet, V. Renaudin, B. Merminod, and M. Kasser. UWB and MEMS based indoor navigation. *Journal of Navigation*, 61(3):369–384, July 2008.
- M. Raibert, K. Blankespoor, G. Nelson, Rob Playter, and the BigDog Team. BigDog, the rough-terrain quadruped robot. In *Proceedings of the 17th World Congress of the International Federation of Automatic Control (IFAC)*, pages 10822–10825, Seoul, South Korea, July 2008.
- C. V. Rao, J. B. Rawlings, and J. H. Lee. Constrained linear state estimation - a moving horizon approach. *Automatica*, 37(10):1619–1628, 2001.
- C. V. Rao, J. B. Rawlings, and D. Q. Mayne. Constrained state estimation for nonlinear discrete-time systems: Stability and moving horizon approximations. *IEEE transactions on automatic control*, 48(2):246–258, 2003.
- H. Rehbinder and X. Hu. Drift-free attitude estimation for accelerated rigid bodies. *Automatica*, 40(4):653–659, 2004.
- V. Renaudin, M. H. Afzal, and G. Lachapelle. Complete triaxis magnetometer calibration in the magnetic domain. *Journal of Sensors*, 2010. Article ID 967245.
- P. Robertson, M. Frassl, M. Angermann, M. Doniec, B. J. Julian, M. Garcia Puyol, M. Khider, M. Lichtenstern, and L. Bruno. Simultaneous localization and mapping for pedestrians using distortions of the local magnetic field intensity in large indoor environments. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–10, 2013.
- D. Roetenberg, H. J. Luinge, C. T. M. Baten, and P. H. Veltink. Compensation of magnetic disturbances improves inertial and magnetic sensing of human body segment orientation. *IEEE Transactions on neural systems and rehabilitation engineering*, 13(3):395–405, 2005.
- Y. Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.
- A. M. Sabatini. Quaternion-based extended Kalman filter for determining orientation by inertial and magnetic sensing. *IEEE Transactions on Biomedical Engineering*, 53(7):1346–1356, 2006.
- S. Salehi, N. Mostofi, and G. Bleser. A practical in-field magnetometer calibration method for IMUs. In *Proceedings of the IROS Workshop on Cognitive Assistive Systems: Closing the Action-Perception Loop*, pages 39–44, Vila Moura, Portugal, October 2012.
- S. Särkkä. *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.
- P. G. Savage. Strapdown inertial navigation integration algorithm design part 1: Attitude algorithms. *Journal of Guidance, Control and Dynamics*, 21(1):19–28, 1998a.

- P. G. Savage. Strapdown inertial navigation integration algorithm design part 2: Velocity and position algorithms. *Journal of Guidance, Control and Dynamics*, 21(2):208–221, 1998b.
- T. B. Schön, F. Lindsten, J. Dahlin, J. Wägberg, C. A. Naesseth, A. Svensson, and L. Dai. Sequential monte carlo methods for system identification. In *Proceedings of the 17th IFAC Symposium on System Identification*, pages 775–786, Beijing, China, October 2015.
- S. Sczyslo, J. Schroeder, S. Galler, and T. Kaiser. Hybrid localization using UWB and inertial sensors. In *Proceedings of the IEEE International Conference on Ultra-Wideband (ICUWB)*, volume 3, pages 89–92, Hannover, Germany, September 2008.
- M. Segal and E. Weinstein. A new method for evaluating the log-likelihood gradient, the Hessian, and the Fisher information matrix for linear dynamic systems. *IEEE Transactions on Information Theory*, 35(3):682–687, 1989.
- M. D. Shuster. A survey of attitude representations. *The Journal of the Astronautical Sciences*, 41(4):439–517, oct–dec 1993.
- M. D. Shuster. The TRIAD algorithm as maximum likelihood estimation. *The Journal of the Astronautical Sciences*, 54(1):113–123, 2006.
- M. D. Shuster and S. D. Oh. Three-axis attitude determination from vector observations. *Journal of Guidance, Control, and Dynamics*, 4(1):70–77, 1981.
- I. Skog and P. Händel. In-car positioning and navigation technologies – A survey. *IEEE Transactions on Intelligent Transportation Systems*, 10(1):4–21, 2009.
- M. A. Skoglund, G. Hendeby, and D. Axehill. Extended Kalman filter modifications based on an optimization view point. In *18th International Conference on Information Fusion*, pages 1856–1861, Washington, D.C., USA, 2015.
- A. Solin, M. Kok, N. Wahlström, T. B. Schön, and S. Särkkä. Modeling and interpolation of the ambient magnetic field by Gaussian processes. *ArXiv e-prints*, September 2015. arXiv:1509.04634.
- D. H. Titterton and J. L. Weston. *Strapdown inertial navigation technology*. IEE radar, sonar, navigaton and avionics series 5. Peter Peregrinus Ltd. on behalf of the Institution of Electrical Engineers, 1997.
- Trivisio Prototyping GmbH. <http://www.trivisio.com>, 2016. Accessed on October 24.
- J. F. Vasconcelos, G. Elkaim, C. Silvestre, P. Oliveira, and B. Cardeira. Geometric approach to strapdown magnetometer calibration in sensor frame. *IEEE Transactions on Aerospace and Electronic systems*, 47(2):1293–1306, April 2011.
- M. Verhaegen and V. Verdult. *Filtering and system identification: a least squares approach*. Cambridge university press, 2007.

Vicon. <http://www.vicon.com>, 2016. Accessed on August 5.

A. Vydhyathan, G. Bellusci, H. Luinge, and P. Slycke. The next generation Xsens motion trackers for industrial applications. https://www.xsens.com/download/pdf/documentation/mti/mti_white_paper.pdf, 2015.

O. J. Woodman. An introduction to inertial navigation. *University of Cambridge, Computer Laboratory, Tech. Rep. UCAMCL-TR-696*, 2007.

Xsens Technologies B.V. <http://www.xsens.com>, 2016. Accessed on October 4.

A. M. Zoubir, V. Koivunen, Y. Chakhchoukh, and M. Muma. Robust estimation in signal processing: A tutorial-style treatment of fundamental concepts. *IEEE Signal Processing Magazine*, 29(4):61–80, 2012.

Paper B

An optimization-based approach to human body motion capture using inertial sensors

Authors: Manon Kok, Jeroen D. Hol and Thomas B. Schön

Edited version of the paper:

M. Kok, J. D. Hol, and T. B. Schön. An optimization-based approach to human body motion capture using inertial sensors. In *Proceedings of the 19th World Congress of the International Federation of Automatic Control*, pages 79–85, Cape Town, South Africa, August 2014.

An optimization-based approach to human body motion capture using inertial sensors

Manon Kok^{*}, Jeroen D. Hol[†] and Thomas B. Schön[‡]

^{*}Dept. of Electrical Engineering,
Linköping University,
SE-581 83 Linköping, Sweden
manko@isy.liu.se

[†]Xsens Technologies B. V.
P.O. Box 559,
7500 AN Enschede, the Netherlands
jeroen.hol@xsens.com

[‡]Dept. of Information Technology
Uppsala University,
SE-751 05 Uppsala, Sweden
thomas.schon@it.uu.se

Abstract

In inertial human motion capture, a multitude of body segments are equipped with inertial measurement units, consisting of 3D accelerometers, 3D gyroscopes and 3D magnetometers. Relative position and orientation estimates can be obtained using the inertial data together with a biomechanical model. In this work we present an optimization-based solution to magnetometer-free inertial motion capture. It allows for natural inclusion of biomechanical constraints, for handling of nonlinearities and for using all data in obtaining an estimate. As a proof-of-concept we apply our algorithm to a lower body configuration, illustrating that the estimates are drift-free and match the joint angles from an optical reference system.

1 Introduction

Human body motion capture is used for many applications such as character animation, sports and biomechanical analysis (Xsens Technologies B.V., 2013). It focuses on simultaneously estimating the relative position and orientation of the different body segments (expressed in terms of the joint angles) and estimating the absolute position of the body. Motion capture is often performed using either vision-based technologies (Moeslund et al., 2006) or using inertial sensors. The main advantage of using inertial sensors over vision-based technologies is that they are not restricted in space and do not require line of sight visibility (Welch and Foxlin, 2002). In inertial human body motion capture, the human body is equipped with inertial measurement units (IMUs), consisting of 3D accelerometers, 3D gyroscopes and 3D magnetometers as shown in Figure 1. Each body segment's position and orientation (pose) can be estimated by integrating the



Figure 1: Examples of inertial motion capture. Upper left: olympic and world champion speed skating Ireen Wüst wearing an inertial motion capture suit with 17 inertial sensors. Upper right: graphical representation of the estimated orientation and position of the body segments. Lower left and right: experiment showing that line of sight visibility is not necessary for inertial motion capture.

gyroscope data and double integrating the accelerometer data in time and combining these integrated estimates with a biomechanical model. Inertial sensors are successfully used for full body motion capture in many applications (Xsens Technologies B.V., 2013; Roetenberg et al., 2013; Kang et al., 2011; Yun and Bachmann, 2006).

Inertial sensors inherently suffer from integration drift. When using inertial sensors for orientation estimation they are therefore generally combined with magnetometers. Magnetometer measurements, however, are known to cause problems in motion capture applications since the magnetic field measured at the different sensor locations is typically different (Luinge et al., 2007; Cooper et al., 2009; Favre et al., 2008). Including information from biomechanical constraints, i.e. information about the body segments being rigidly connected, can eliminate the need of using magnetometer measurements. Incorporating these constraints, the

sensor's *relative* position and orientation become observable as long as the subject is not standing completely still (Hol, 2011). Estimating joint angles using a pair of inertial sensors, where each sensor estimates its own orientation using an extended Kalman filter (EKF) (Yuan and Chen, 2013) is therefore computationally cheap, but valuable information from biomechanical constraints is lost. Existing approaches therefore include the biomechanical constraints like for instance in Luinge et al. (2007) where an EKF is run using only the accelerometer and gyroscope measurements and a least-squares filter is added to incorporate the biomechanical constraints.

To allow for natural inclusion of biomechanical constraints, we introduce a new optimization-based approach for inertial motion capture. Compared to filtering approaches, optimization-based approaches are computationally expensive. Recent developments in both computational power and in available algorithms have, however, opened up possibilities for solving large-scale problems efficiently and even in real-time (Mattingley and Boyd, 2010). Using an optimization formulation of the problem, a smoothing estimate can be obtained and nonlinearities can be handled. It also opens up possibilities for simultaneously estimating calibration parameters and for incorporating non-Gaussian noise.

The paper is organized as follows. After introducing the problem formulation in Section 2, in Section 3 we will introduce the biomechanical model, discussing the relevant coordinate frames, variables and biomechanical constraints. In Section 4 we will subsequently introduce the dynamic and sensor models. In Section 6 we will discuss experimental results, focusing on a subproblem, namely a lower body configuration consisting of 7 sensors, assuming a known calibration and not including any position aiding. These experiments are intended to serve as a proof-of-concept. A more in-depth analysis including a comparison with other methods is planned for future work.

Note that using inertial sensors and biomechanical constraints only, the *absolute* position is not observable, i.e. any translation of the body's position estimates will lead to an equally valid solution of the estimation problem. For example in the case of the speed skater in Figure 1, the estimated pose of the speed skater will resemble the "true" motion, but the exact location on the ice rink is not observable. This unobservability typically results in a drift of the body's absolute position over time. Because of this, it is not possible to compare our position estimates with those of the optical reference system and for now we focus on analysis of the joint angles. To estimate absolute position it is necessary to include e.g. GPS, ultra-wideband (Hol, 2011) or zero velocity updates when the foot is at stand still (Callmer, 2013; Woodman, 2010) and this is planned for future work.

2 Problem formulation

The use of inertial sensors for human body motion capture requires inertial sensors to be placed on different body segments. The knowledge about the placement of the sensors on the body segments and the body segments' connections to each other by joints can be incorporated using a biomechanical model.

The problem of estimating the relative position and orientation of each body segment is formulated as a constrained estimation problem. Given N measurements $y_{1:N} = \{y_1, \dots, y_N\}$, a point estimate of the variables z can be obtained as a constrained maximum a posteriori (MAP) estimate, maximizing the posterior density function

$$\begin{aligned} \max_z \quad & p(z \mid y_{1:N}) \\ \text{s.t.} \quad & c_e(z) = 0, \end{aligned} \quad (1)$$

where $c_e(z)$ represents the equality constraints. In our problem, z consists of both static parameters θ and time-varying variables $x_{1:N}$. Using this together with the Markov property of the time-varying variables and the fact that the logarithm is a monotonic function, we can rewrite (1) as

$$\begin{aligned} \min_{z=\{x_{1:N}, \theta\}} \quad & \underbrace{-\log p(x_1 \mid y_1) - \log p(\theta)}_{\text{initialization}} \\ & - \underbrace{\sum_{t=2}^N \log p(x_t \mid x_{t-1}, \theta)}_{\text{dynamic model}} - \underbrace{\sum_{t=1}^N \log p(y_t \mid x_t, \theta)}_{\text{biomechanical/sensor model}} \\ \text{s.t.} \quad & c_{\text{bio}}(z) = 0. \end{aligned} \quad (2)$$

Obtaining the MAP estimate thus amounts to solving a constrained optimization problem where the constraints $c_{\text{bio}}(z)$ originate from a biomechanical model. The cost function consists of different parts related to the initialization of the variables, a dynamic model for the time-varying states and a biomechanical and sensor model. More details about the variables, the different parts of the cost function and the constraints are provided in Sections 3 and 4.

The optimization problem (2) is solved using an infeasible start Gauss-Newton method (Boyd and Vandenberghe, 2004). The number of variables in the problem will become large already for short experiments and a small number of segments. The problem (2) can, however, still be solved efficiently due to its inherent sparseness.

3 Biomechanical model

A biomechanical model represents the human body as consisting of body segments connected by joints. In the example application in Figure 1 the body is modeled as consisting of 23 segments, whereas Figure 2 illustrates two of these body segments. These can be thought of as the upper and lower leg, each with a sensor attached to it. The main purpose of Figure 2 is to introduce the different coordinate frames, variables and calibration parameters. These definitions can straightforwardly be extended to any sensor and any body segment. The relevant coordinate frames are:

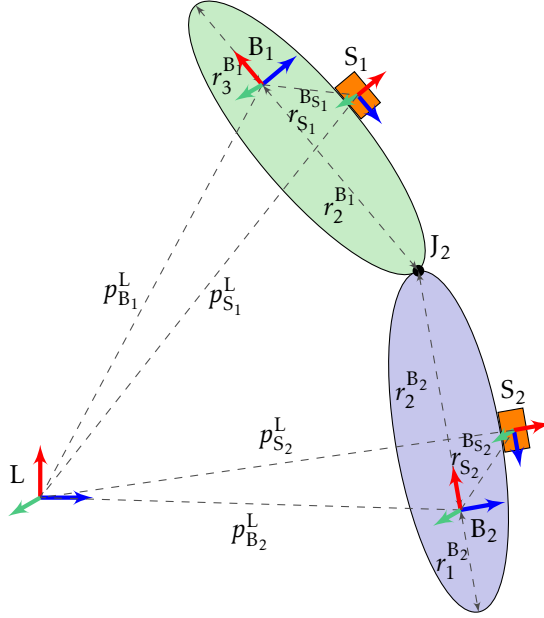


Figure 2: Connection of two segments and definition of the variables and coordinate frames.

The local coordinate frame L aligned with the local gravity vector, with the z -axis pointing up. The horizontal directions are defined according to any convenient choice of local coordinates.

The body segment coordinate frame B_j fixed to the bone in body segment B_j . Its origin can be anywhere along the bone, but it is usually in the center of rotation of a joint.

The sensor coordinate frame S_i of the moving IMU S_i . Its origin is located in the center of the accelerometer triad and its axes are aligned to the casing. All measurements of the IMU are resolved in this frame.

In setting up the optimization problem (2), the first step is to define the set of sensors $\mathcal{S}$, the set of body segments $\mathcal{B}$ and the set of joints $\mathcal{J}$ in the problem. Each inertial sensor needs to be mounted on the body, and sensor S_i is assumed to be placed on body segment B_{S_i} . The distance $r_{S_i}^{B_{S_i}}$ and orientation $q^{B_{S_i}S_i}$ of sensor S_i with respect to body segment B_{S_i} are without loss of generality assumed to be known from calibration.

Our knowledge of the human body can be used to identify which body segments are connected by which joints, i.e. the set $\mathcal{B}_{J_k}$ needs to be determined for each joint J_k . To express the location of the joint in the body frames of the connected body segments, the distances $r_k^{B_j}$ from the body frame B_j to joint k , need

to be defined for all joints $J_k \in \mathcal{J}$ and all $B_j \in \mathcal{B}_{J,k}$. We assume without loss of generality that they are known from calibration. Generally, all joints are assumed to be ball-and-socket joints, but we incorporate additional knowledge about a subset of the joints, denoted by $\mathcal{H}$, which we assume to be hinge joints.

For reasons that will be discussed in Section 4, we define the set of time steps in the optimization as $\mathcal{T}$ rather than explicitly summing over all time steps $t = 1 \dots N$ as in (2). The variables in the optimization problem are then given by

- the position $p_{S_i,t}^L$ and velocity $v_{S_i,t}^L$ of sensor S_i in the local frame L , $\forall S_i \in \mathcal{S}$ and $\forall t \in \mathcal{T}$,
- the orientation $q_t^{LS_i}$ of sensor S_i with respect to the local frame L , $\forall S_i \in \mathcal{S}$ and $\forall t \in \mathcal{T}$,
- the position $p_{B_j,t}^L$ of body segment B_j in the local frame L , $\forall B_j \in \mathcal{B}$ and $\forall t \in \mathcal{T}$,
- the orientation $q_t^{LB_j}$ of body segment B_j with respect to the local frame L , $\forall B_j \in \mathcal{B}$ and $\forall t \in \mathcal{T}$,
- the gyroscope bias b_{ω,S_i} of sensor S_i , $\forall S_i \in \mathcal{S}$,
- the mean acceleration state of one of the sensors $S_i \in \mathcal{S}$, $\forall t \in \mathcal{T}$.

Defining the number of sensors as N_S and the number of body segments as N_B , the number of variables in the optimization problem is $z \in \mathbb{R}^{(9N_S+6N_B+3)N+3N_S}$. When we solve the optimization problem, we encode the rotation states using a three-dimensional state vector (Crassidis et al., 2007; Grisetti et al., 2010; Hol, 2011). Throughout the paper, we typically interchangeably make use of the unit quaternion q^{LS} and the rotation matrix R^{LS} as representations of the orientation. The quaternion conjugate, representing the inverse rotation will be represented by $(q^{LS})^c = q^{SL}$. Similarly for the rotation matrix, $(R^{LS})^T = R^{SL}$.

More details about the gyroscope bias variables and the reason for the inclusion of the mean acceleration state will be given in Section 4.2.

Based on the biomechanical model it is possible to derive relations between the different variables. We will categorize them in three classes.

Joints between the body segments. The constraints $c_{\text{bio}}(z)$ in the optimization problem (2) enforce the body segments to be connected at the joint locations at all times,

$$c_{\text{bio}}(z) = p_{B_m,t}^L + R_t^{LB_m} r_k^{B_m} - p_{B_n,t}^L - R_t^{LB_n} r_k^{B_n}, \quad B_n, B_m \in \mathcal{B}_{J_k}, \quad (3)$$

which is included for all $J_k \in \mathcal{J}$ and $t \in \mathcal{T}$. This leads to N_J constraints at each time step t in the optimization problem (2), where N_J is the number of joints.

Placement of the sensors on the body segments. The position and orientation of sensor S_i can be expressed in terms of its position and orientation on body

segment B_{S_i} . Ideally, this can be incorporated using equality constraints in (2). However, it is physically impossible to place the sensor directly on the bone. Hence, it has to be placed on the soft tissue and the sensor will inevitably move slightly with respect to the bone. We therefore model the position and orientation of sensor S_i on body segment B_{S_i} as

$$p_{S_i,t}^L = p_{B_{S_i},t}^L + R_t^{LB_{S_i}} \left(r_{S_i}^{B_{S_i}} + e_{p,t}^{B_{S_i}} \right), \quad (4a)$$

$$q_t^{LS_i} = q_t^{LB_{S_i}} q^{B_{S_i}S_i} \exp \left(\frac{1}{2} e_{q,t}^{S_i} \right), \quad (4b)$$

where we assume $e_{p,t}^{B_{S_i}} \sim \mathcal{N}(0, \Sigma_p)$ and $e_{q,t}^{S_i} \sim \mathcal{N}(0, \Sigma_q)$.

Rotational freedom of the joints. For some joints, it is known that their rotation is (mainly) limited to one or two axes. An example of this is the knee which is a hinge joint, although it can in practice flex a little around the other axes too. Minimizing

$$e_{J_k,t} = \begin{bmatrix} n_1^T \\ n_2^T \\ n_3^T \end{bmatrix} \left(R_t^{LB_{n_2}} \right)^T R_t^{LB_{n_2}} n_2, \quad B_n, B_m \in \mathcal{B}_{J_k}, \quad (5)$$

where n_1, n_2 and n_3 denote the different axis directions and $e_{J_k,t} \sim \mathcal{N}(0, \Sigma_k)$, will minimize the rotation around any but the n_2 -axis. This cost function can be included at any time t for any joint k that is a hinge joint, i.e. $\forall J_k \in \mathcal{H}, \forall t \in \mathcal{T}$. Note that inclusion of this knowledge is optional in the algorithm.

4 Dynamic and sensor models

The sensor's position, velocity and orientation at each time instance can be related by a dynamic model in which the accelerometer and gyroscope measurements are used as inputs (Gustafsson, 2012; Hol, 2011). In this work we choose a slightly different approach to reduce the number of variables in the optimization problem (2). To achieve high update rates using a relatively small number of variables, we use an approach similar to the one discussed by Savage (1998a,b). Hence, strap-down inertial integration, in which the accelerometer and gyroscope signals are integrated, is run at high update rates. This leads to accelerometer measurements Δp and Δv representing a difference in position and velocity and gyroscope measurements Δq representing a difference in orientation. These are integrated for $\frac{T_s}{T}$ times, where T_s is the sampling time of the inertial sensors and T is the sampling time used in the optimization problem (2).

4.1 Dynamic model

The position, velocity and orientation of each sensor S_i are related from time t to time $t + T$ using the accelerometer measurements $\Delta p_t^{S_i}, \Delta v_t^{S_i}$ and the gyroscope

measurements $\Delta q_t^{S_i}$. The position and velocity states at each time step are modeled according to

$$p_{S_i,t+T}^L = p_{S_i,t}^L + T v_{S_i,t}^L + R_t^{LS_i} \left(\Delta p_t^{S_i} + w_{p,t}^{S_i} \right) + \frac{T^2}{2} g^L, \quad (6a)$$

$$v_{S_i,t+T}^L = v_{S_i,t}^L + R_t^{LS_i} \left(\Delta v_t^{S_i} + w_{v,t}^{S_i} \right) + T g^L, \quad (6b)$$

where $\Delta p_t^{S_i}$ and $\Delta v_t^{S_i}$ denote the inputs based on the accelerometer measurements. The noise terms are modeled as $w_{p,t} \sim \mathcal{N}(0, Q_p)$ and $w_{v,t} \sim \mathcal{N}(0, Q_v)$. The earth gravity is denoted by g^L . The orientation states are modeled as

$$q_{t+T}^{LS_i} = q_t^{LS_i} \Delta q_t^{S_i} \exp\left(\frac{1}{2} w_{q,t}^{S_i}\right), \quad (6c)$$

where $\Delta q_t^{S_i}$ denotes the gyroscope measurements, corrected for the estimated gyroscope bias, and $w_{q,t}^{S_i} \sim \mathcal{N}(0, Q_q)$.

Since (6) models the states in terms of their value at the previous time step, the state at the first time instance needs to be treated separately. The orientation $q_1^{LS_i}$ of each sensor S_i is estimated using the first accelerometer and magnetometer sample of that sensor. Note that this is the only place in the algorithm where magnetometer measurements are used. The variables $q_1^{LS_i}$ are then initialized around this estimated orientation with additive noise $e_{q_1}^{S_i} \sim \mathcal{N}(0, \Sigma_{q_1})$. The position $p_{S_i,1}^L$ of one of the sensors is without loss of generality initialized around zero with additive noise $e_{p_1} \sim \mathcal{N}(0, \Sigma_{p_1})$. This defines the origin of the local coordinate frame L.

4.2 Sensor model

The gyroscope measurements are affected by a slowly time-varying sensor bias. For relatively short experiments, the sensor biases of all sensors $S_i \in \mathcal{S}$ can be assumed to be constant. Hence, we include only one three-dimensional variable for each sensor to represent the gyroscope bias. This variable b_{ω, S_i} is modeled as $b_{\omega, S_i} \sim \mathcal{N}(0, \Sigma_{b_\omega})$.

As described in Section 1, we do not include position aiding in our problem, resulting in only *relative* position and orientation observability. A problem that can be encountered for this case is that of so-called gravity leakage. Because the subject's *absolute* inclination is unobservable, the gravity vector risks being misinterpreted as an acceleration. In the case of stationary measurements, when the accelerometer only measures the gravity vector, the accelerometer measurements can be used as a source of absolute inclination information. In case of motion, the accelerometer measurements will measure an additional acceleration. It can, however, still be assumed that the mean acceleration over a certain time period is zero (Luinge, 2002). We therefore assume that one sensor follows this acceleration model for all $t \in \mathcal{T}$, up to some noise $e_a \sim \mathcal{N}(0, \Sigma_a)$.

5 Resulting algorithm

The biomechanical model from Section 3 and the dynamic and sensor models from Section 4 can be combined and used to describe the probability density functions in (2). Eliminating all constant terms from the optimization, this results in a constrained weighted least-squares problem. This problem is given by

$$\begin{aligned}
 \min_z \sum_{t \in \mathcal{T}} \sum_{S_i \in \mathcal{S}} & \left(\underbrace{\|e_{p,t}^{S_i}\|_{\Sigma_p^{-1}}^2 + \|e_{q,t}^{B_{S_i}}\|_{\Sigma_q^{-1}}^2}_{\text{placement sensors on body (4)}} + \underbrace{\|w_{p,t}^{S_i}\|_{Q_p^{-1}}^2 + \|w_{v,t}^{S_i}\|_{Q_v^{-1}}^2 + \|w_{q,t}^{S_i}\|_{Q_q^{-1}}^2}_{\text{dynamic model (6)}} \right) \\
 & + \underbrace{\sum_{S_i \in \mathcal{S}} \|b_{\omega, S_i}\|_{\Sigma_{b_\omega}^{-1}}^2}_{\text{gyroscope bias}} + \underbrace{\| \sum_{t \in \mathcal{T}} \sum_{J_k \in \mathcal{H}} e_{J_k, t} \|_{\Sigma_k^{-1}}^2}_{\text{hinge (5)}} \\
 & + \underbrace{\|e_{p_1}\|_{\Sigma_{p_1}^{-1}}^2 + \sum_{S_i \in \mathcal{S}} \|e_{q_1}^{S_i}\|_{\Sigma_{q_1}^{-1}}^2}_{\text{initialization}} + \underbrace{\| \sum_{t \in \mathcal{T}} e_{a,t} \|_{\Sigma_a^{-1}}^2}_{\text{acceleration model}}, \\
 \text{s.t. } c_{\text{bio}}(z) &= p_{B_m, t}^L + R_t^{\text{LB}_m} r_k^{B_m} - p_{B_n, t}^L - R_t^{\text{LB}_n} r_k^{B_n}, \\
 & B_n, B_m \in \mathcal{B}_{J_k} \forall J_k \in \mathcal{J}, \forall t \in \mathcal{T},
 \end{aligned} \tag{7}$$

where the constraints are based on (3).

The complete algorithm is summarized in Algorithm 1. Note that in our current implementation the optimization is performed over the entire data set and the computations are therefore done offline. We plan to extend the approach to a moving horizon approach (Rao et al., 2001) to enable processing of longer data sets and to allow for online estimation.

The covariance matrices in (7) representing the sensor covariances are determined using Allan variance analysis (El-Sheimy et al., 2008). The covariance matrices related to the placement of the sensors on the body, the hinge constraint and the acceleration model, do not represent any physical quantities and are chosen more or less ad hoc. Experiments have shown that the solution of the optimization problem is not very sensitive to the tuning of these values.

The optimization (7) is started using an initial estimate of the variables z_0 . All variables are initialized at zero except for the orientations at the first time step, which are initialized around their estimated orientation, as described in Section 4.2. This is an infeasible solution, justifying the need for an infeasible start optimization algorithm.

6 Experiments

We validated our approach with experiments using an MVN Awinda system (Xsens Technologies B.V., 2013) which is a wireless inertial motion capture system with 17 sensors attached to different body segments as shown in Figure 3. An optical motion capture system has been used as a source of reference data. Since our

Algorithm 1 Inertial human motion capture

1. Define the set of sensors $\mathcal{S}$, the set of body segments $\mathcal{B}$ and the set of joints $\mathcal{J}$. Mount the inertial sensors on the body and
 - (a) define for each sensor $S_i \in \mathcal{S}$ on which body segment $B_{S_i} \in \mathcal{B}$ it is placed. Calibrate the system to obtain the position $r_{S_i}^{B_{S_i}}$ and orientation $q^{B_{S_i}S_i}$ of each sensor $S_i \in \mathcal{S}$ on body segment $B_{S_i} \in \mathcal{B}$,
 - (b) define the set of body segments $\mathcal{B}_{J_k}$ connected to each joint k for all $J_k \in \mathcal{J}$. Calibrate the system to obtain the distances $r_k^{B_j}$ of each body segment coordinate frames $B_j \in \mathcal{B}_{J_k}$ to the different joints k ,
 - (c) define the subset $\mathcal{H}$ of joints that are restricted in their rotations and can be regarded as a hinge joint.
2. Perform an experiment collecting inertial measurements $\Delta p_t^{S_i}, \Delta v_t^{S_i}$ and $\Delta q_t^{S_i}$ and a magnetometer measurement at $t = 1, y_{m,1}^{S_i}$.
3. Postprocess the data
 - (a) Initialize z_0 and set $l = 0$.
 - (b) Determine the values of the cost functions and the constraints in (2), their Jacobians and the approximate Hessian of the cost function. Determine a step direction using an infeasible start Gauss-Newton algorithm and update $z_l \rightarrow z_{l+1}$.
 - (c) Set $l := l + 1$ and iterate from 3(b) until the algorithm is converged and the solution z_{l+1} is feasible.

focus is on the legs, one leg has been equipped with optical markers, providing reference position and orientation of the foot sensor, lower leg sensor, upper leg sensor and – not visible in the figure – the pelvis sensor.

Inertial data has been collected at 30 Hz. The sensors, however, run the strap-down integration algorithm discussed in Section 4 internally at 600 Hz to capture the high bandwidth of the measurement signals during impact, for instance during foot impact on the ground. To speed up the computations, the optimization algorithm itself has been run at a frequency of 10 Hz.

The optimization problem typically converges in a few iterations. To solve the problem for an experiment of 10 seconds takes about 5 minutes on an AMD X4 2.8 GHz processor for a first inefficient Matlab implementation of the algorithm. Initial tests with a C-implementation, however, show that speed improvements of up to 500 times are easily obtained. Taking into account that at the moment we postprocess the whole data set while for a real-time application a moving horizon can be used, we think that a real-time implementation of the algorithm is indeed quite possible.

The collected inertial data has been postprocessed used in the optimization problem (2) for a lower body configuration consisting of a set $\mathcal{S}$ of 7 sensors placed on 7 body segments $\mathcal{B}$: both feet, both lower legs, both upper legs and



Figure 3: Experimental setup where the human body is equipped with inertial sensors on different body segments. Optical markers for the reference system have been placed on the right foot sensor, right lower and upper leg sensors and – not visible in the figure – the pelvis.

the pelvis. The position of each sensor $S_i \in \mathcal{S}$ on the body segment $r_{S_i}^{B_{S_i}}$ has been manually measured. The orientations of the sensors on the body segments $q_{S_i}^{B_{S_i}}$ for all $S_i \in \mathcal{S}$ have been determined by standing still in a pre-determined pose as described by Roetenberg et al. (2013). The 7 body segments are connected by 6 joints $\mathcal{J}$ of which the two knee joints are assumed to be hinge joints. Calibrating for the distances $r_k^{B_j}$ amounts to defining the distances between the different joint centers which is again done by manual measuring. We acknowledge that this is an inaccurate calibration method and as future work we therefore plan to extend the algorithm to automatically estimate these calibration parameters.

Figure 4 visualizes the pose of the lower body of a walking subject estimated using Algorithm 1 for parts of an experiment. Note that our experimental setup does allow for accurate *absolute* position estimates. The location of the different steps has therefore been corrected for one joint location using the position estimates from the optical reference system. The steps are taken from a short experiment and the optimization is run at 30 Hz for plotting purposes.

To compare our relative orientation results to those of the optical reference system, we focus on the estimated joint angle of the right knee during an experiment of around 37 seconds. Joint angles are defined as the angle between two connected body segments at the joint center. For the knee joint, the bending of the

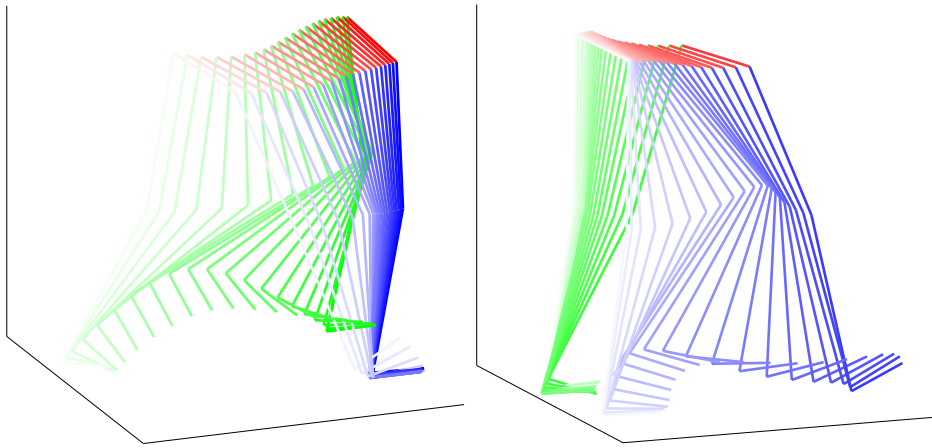


Figure 4: Estimated pose of the lower body at different times during a step of the left leg (left) and the right leg (right). The view is chosen such that we view the subject from the right, and the right leg is depicted in blue, the left leg in green and the connection between the hips in red.

knee during walking is referred to as flexion/extension. The rotation around the other two axes (abduction/adduction and internal/external rotation) are generally quite small for this joint. Because it is not possible to observe the joint center and sensors/markers are generally placed on the soft tissue instead of on the bone, computation of joint angles depends on a model of the joint locations in the body. Theoretically, it is possible to estimate the joint angle from the orientation results of the sensor if the exact location of the sensors with respect to the joints is known, i.e. in case of a perfect calibration, and if the sensors would be rigidly attached to the bone. In practice this is clearly not possible. However, since both the inertial sensors and the optical reference markers are placed on the same location on the body segments as shown in Figure 3, it is still possible to compare the angles to assess the quality of our estimates.

To be able to compare our joint angle estimates to those of the reference system, a coordinate frame alignment between the sensor coordinate frame and the coordinate frame of the optical markers needs to be performed. This has been done as described by Hol (2011). Note that due to limited excitation of the upper leg sensor, it was not possible to do this alignment based on the sensor signals. Instead, the alignment has been performed based on the joint angle estimates. The joint angle estimates from our algorithm can be seen to match the joint angles from the optical reference system. A more quantitative analysis can be performed when the calibration parameters are properly estimated and position aiding is included. Note that due to the limited size of the measurement volume of the

optical reference system, the movements are quite restricted and at some time instances in the experiment the optical reference data is not present.

From our optimization algorithm it is also possible to estimate the joint angles from the angles of the body segments. These are included in red in Figure 5. There is no validation for the angles obtained in this way, but the estimated abduction/adduction and internal/external rotation are considerably closer to zero, as we would expect from our knowledge that these rotations are quite small.

7 Conclusions and future work

An optimization approach to inertial human body motion capture has been developed, capable of estimating the relative position and orientation of the body segments. Experimental results show that the algorithm works well, quickly converging to a feasible solution and resulting in drift-free joint angle estimates which match the joint angles from an optical reference system.

We plan to extend the approach to also estimate the calibration parameters and to include position aiding in the form of zero velocity updates at stand still and ultra-wideband position aiding (Hol et al., 2009). This will also allow a more quantitative analysis of the results. Future work also includes adding more body segments, modeling of non-Gaussian noise where appropriate and implementing a moving horizon estimation version of the algorithm.

Acknowledgements

This work is supported by MC Impulse, a European Commission, FP7 research project, by CADICS, a Linnaeus Center funded by the Swedish Research Council (VR) and by BALANCE, a European Commission, FP7 research project. The authors would like to thank Dr. Daniel Roetenberg for valuable discussions and Raymond Zandbergen for implementation support.

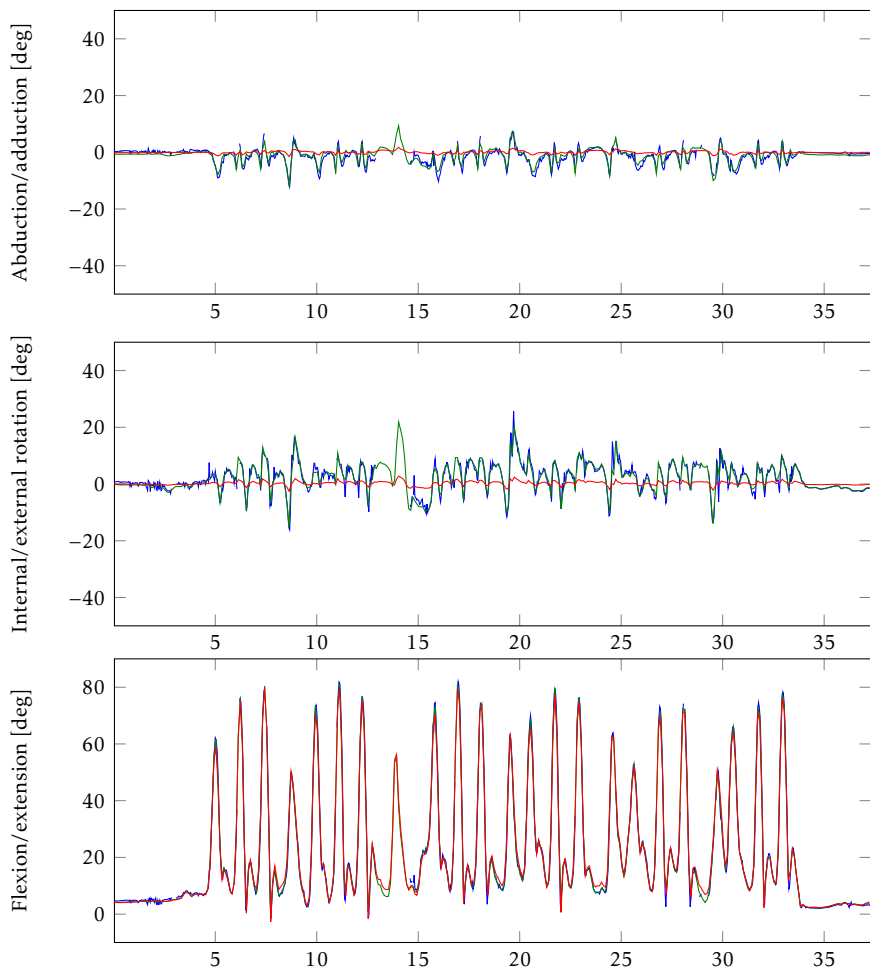


Figure 5: Knee joint angles for the right knee for an experiment consisting of 23 steps. The optical reference data is plotted in blue, the joint angle estimated from the sensor's orientations, using our algorithm is plotted in green, the joint angle from the body segment orientations is plotted in red. Best viewed in color.

Bibliography

- S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- J. Callmer. *Autonomous Localization in Unknown Environments*. Dissertation no. 1520, Linköping University, Linköping, Sweden, June 2013.
- G. Cooper, I. Sheret, L. McMillan, K. Siliverdis, N. Sha, D. Hodgins, L. Kenney, and D. Howard. Inertial sensor-based knee flexion/extension angle estimation. *Journal of Biomechanics*, 42(16):2678 – 2685, 2009.
- J. L. Crassidis, F. L. Markley, and Y. Cheng. A survey of nonlinear attitude estimation methods. *Journal of Guidance, Control, and Dynamics*, 30(1):12–28, 2007.
- 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, 2008.
- J. Favre, B. M. Jolles, R. Aissaoui, and K. Aminian. Ambulatory measurement of 3D knee joint angle. *Journal of biomechanics*, 41(5):1029–1035, 2008.
- G. Grisetti, R. Kümmerle, C. Stachniss, U. Frese, and C. Hertzberg. Hierarchical optimization on manifolds for online 2D and 3D mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 273–278, Anchorage, Alaska, 2010.
- F. Gustafsson. *Statistical Sensor Fusion*. Studentlitteratur, 2012.
- J. D. Hol. *Sensor fusion and calibration of inertial sensors, vision, ultra-wideband and GPS*. Dissertation no. 1368, Linköping University, Linköping, Sweden, June 2011.
- J. D. Hol, F. Dijkstra, H. Luinge, and T. B. Schön. Tightly coupled UWB/IMU pose estimation. In *Proceedings of the IEEE International Conference on Ultra-Wideband (ICUWB)*, pages 688–692, Vancouver, Canada, September 2009.
- D. H. Kang, Y. J. Jung, A. J. Park, and J. W. Kim. Human body motion capture system using magnetic and inertial sensor modules. In *Proceedings of the 5th international universal communication symposium (IUCS)*, Gumi, Korea, October 2011.
- M. Kok, J. D. Hol, and T. B. Schön. An optimization-based approach to human body motion capture using inertial sensors. In *Proceedings of the 19th World Congress of the International Federation of Automatic Control*, pages 79–85, Cape Town, South Africa, August 2014.
- H. J. Luinge. *Inertial Sensing of Human Movement*. PhD thesis, University of Twente, Enschede, the Netherlands, October 2002.

- H. J. Luinge, P. H. Veltink, and C. T. M. Baten. Ambulatory measurement of arm orientation. *Journal of Biomechanics*, 40(1):78–85, 2007.
- J. Mattingley and S. Boyd. Real-time convex optimization in signal processing. *IEEE Signal Processing Magazine*, 27(3):50–61, 2010.
- T. B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 104(2–3):90–126, 2006.
- C. V. Rao, J. B. Rawlings, and J. H. Lee. Constrained linear state estimation - a moving horizon approach. *Automatica*, 37(10):1619–1628, 2001.
- D. Roetenberg, H. J. Luinge, and P. Slycke. Xsens MVN: Full 6DOF human motion tracking using miniature inertial sensors, May 2013.
- P. G. Savage. Strapdown inertial navigation integration algorithm design part 1: Attitude algorithms. *Journal of Guidance, Control and Dynamics*, 21(1):19–28, 1998a.
- P. G. Savage. Strapdown inertial navigation integration algorithm design part 2: Velocity and position algorithms. *Journal of Guidance, Control and Dynamics*, 21(2):208–221, 1998b.
- G. Welch and E. Foxlin. Motion tracking: No silver bullet, but a respectable arsenal. *IEEE Computer Graphics and Applications*, 22(6):24–38, 2002.
- O. J. Woodman. *Pedestrian localisation for indoor environments*. PhD thesis, University of Cambridge, United Kingdom, September 2010.
- Xsens Technologies B.V. <http://www.xsens.com>, 2013. Accessed on November 17.
- Q. Yuan and I.-M. Chen. 3-D localization of human based on an inertial capture system. *IEEE Transactions on Robotics*, 29(3):806–812, 2013.
- X. Yun and E. R. Bachmann. Design, implementation, and experimental results of a quaternion-based Kalman filter for human body motion tracking. *IEEE Transactions on Robotics*, 22(6):1216–1227, 2006.

Paper C

A scalable and distributed solution to the inertial motion capture problem

Authors: Manon Kok, Sina Khoshfetrat Pakazad, Thomas B. Schön, Anders Hansson and Jeroen D. Hol

Edited version of the paper:

M. Kok, S. Khoshfetrat Pakazad, T. B. Schön, A. Hansson, and J. D. Hol. A scalable and distributed solution to the inertial motion capture problem. In *Proceedings of the 19th International Conference on Information Fusion*, pages 1348–1355, Heidelberg, Germany, July 2016b.

A scalable and distributed solution to the inertial motion capture problem

Manon Kok^{*}, Sina Khoshfetrat Pakazad^{*}, Thomas B. Schön[†], Anders Hansson^{*}
and Jeroen D. Hol[‡]

^{*}Dept. of Electrical Engineering,
Linköping University,
SE-581 83 Linköping, Sweden
{manko, sina.kh.pa, hansson}@isy.liu.se

[†]Dept. of Information Technology
Uppsala University,
SE-751 05 Uppsala, Sweden
thomas.schon@it.uu.se

[‡]Xsens Technologies B. V.
P.O. Box 559,
7500 AN Enschede, the Netherlands
jeroen.hol@xsens.com

Abstract

In inertial motion capture, a multitude of body segments are equipped with inertial sensors, consisting of 3D accelerometers and 3D gyroscopes. Using an optimization-based approach to solve the motion capture problem allows for natural inclusion of biomechanical constraints and for modeling the connection of the body segments at the joint locations. The computational complexity of solving this problem grows both with the length of the data set and with the number of sensors and body segments considered. In this work, we present a scalable and distributed solution to this problem using tailored message passing, capable of exploiting the structure that is inherent in the problem. As a proof-of-concept we apply our algorithm to data from a lower body configuration.

1 Introduction

Inertial motion capture focuses on estimating the relative position and orientation (pose) of different human body segments. To this end, inertial sensors (3D accelerometers and 3D gyroscopes) are placed on different body segments as shown in Figure 1. Each body segment’s pose can be estimated by integrating the gyroscope data and double integrating the accelerometer data in time and combining these integrated estimates with a biomechanical model. Inertial sensors are successfully used for full body motion capture in many applications such as character animation, sports and biomechanical analysis (Xsens Technologies B. V., 2016; Roetenberg et al., 2013; Kang et al., 2011; Yun and Bachmann, 2006).

In Kok et al. (2014), an optimization-based solution to the inertial motion capture problem was presented. It post-processes the data to obtain a smoothing estimate of the body's relative pose. The problem is solved using sequential quadratic programming (SQP) (Nocedal and Wright, 2006). The method was shown to result in drift-free and accurate pose estimates. Using an optimization-based approach allows for natural inclusion of biomechanical constraints and for modeling the connection between the body segment at the joint locations. Furthermore, it naturally handles nonlinearities and opens up the possibility for incorporating non-Gaussian noise and for simultaneous estimation of calibration parameters.



Figure 1: Example of inertial motion capture. Left: olympic and world champion speed skating Ireen Wüst wearing an inertial motion capture suit with 17 inertial sensors. Right: graphical representation of the estimated position and orientation of the body segments.

For applications which require real-time pose estimates, approximate solutions to the full smoothing problem need to be considered, for instance using filtering or moving horizon estimation (MHE) (Rao et al., 2001). In these approaches, data up to a current time point is used to estimate the current pose. However, in case real-time estimates are not required, all available data can be used to obtain a smoothing estimate. Compared to filtering and MHE, obtaining a smoothing estimate is computationally more expensive and can be challenging both due to the computational complexity of solving the problem and due to storage requirements for constructing the problem. This is specifically of concern when processing long data sets.

In this paper we solve the same problem as in Kok et al. (2014). Again we use SQP, but at each iteration we compute the search directions using the message passing algorithm presented in Khoshfetrat Pakazad et al. (2016). This allows us to efficiently make use of the structure inherent in the problem. We exploit this structure in two different ways:

1. We reorder the problem based on time. This allows us to solve the problem by solving a large number of small problems which enables us to process long data sets.
2. We reorder the problem based on sensors and body segments. This leads to less computational benefits – the number of sensors and body segments is typically much smaller than the number of time steps considered – but it allows for solving the problem in a *distributed* manner. It also relaxes the need for a centralized unit and streaming of data to it.

Using message passing for computing the search directions for the time-ordered problem has close connections to serial dynamic programming (Bertsekas, 1995). This is due to the chain-like coupling structure in the problem. In fact, using serial dynamic programming, the search directions can be computed by sweeping through the available data forward and backwards, similar to the approach used for Rauch-Tung-Striebel (RTS) smoothing (Rauch et al., 1965). Using message passing, we compute the search directions by simultaneously starting from the first and final time steps and sweeping towards the middle of the data set and back. This allows us to speed up the search direction computation by a factor of two. Notice that unlike existing scalable algorithms for solving big data problems that rely on first-order methods, see e.g. Cevher et al. (2014), the proposed algorithm solely relies on second-order methods. Consequently, this algorithm enjoys a far superior superlinear convergence rate (Wright, 1997), in comparison to at best linear convergence of other algorithms.

If we only consider the lower body for the sensor-ordered problem, the chain-like coupling structure will also be present in the problem. Instead of running through time, this chain runs from one foot through both legs to the other foot. Consequently, it enjoys the same similarities to serial dynamic programming as discussed above. For the full body, the coupling structure will not be chain-like. It will, however, have an inherent tree structure. Hence, we can still use message passing for computing the search directions. In this paper, we focus on the lower body to simplify both the notation and the biomechanical modeling. The presented material can, however, straightforwardly be extended to the full body problem.

The paper is organized as follows. In Section 2 we introduce the inertial motion capture problem for which the models are subsequently introduced in Section 3. In Section 4, we reorder the problem in the two ways described above. These two equivalent formulations of the original problem enjoy a special structure which allows us to use message passing to compute the search directions. The message passing algorithm will be introduced in Section 5. The resulting algorithm that can be used to solve the reordered problems is subsequently discussed in Section 6. In Section 7, we will discuss experimental results where the algorithm is applied to data from inertial sensors placed on the lower body.

Table 1: Notation to refer to the variables and the constraints in our problem, introduced in Sections 2 and 3.3, respectively.

Symbol	Definition	Explanation
x^{S_i}	$x^{S_i} = \{x_1^{S_i}, \dots, x_{N_T}^{S_i}\}$	Time-varying variables pertaining to sensor S_i
x^{B_i}	$x^{B_i} = \{x_1^{B_i}, \dots, x_{N_T}^{B_i}\}$	Time-varying variables pertaining to body segment B_i
x^i	$x^i = \{x^{B_i}, x^{S_i}\}$	Time-varying variables pertaining to sensor S_i and body segment B_i
x_t	$x_t = \{x_t^{S_1}, \dots, x_t^{S_{N_S}}, x_t^{B_1}, \dots, x_t^{B_{N_S}}\}$	Time-varying variables pertaining to time t
x	$x = \{x_1, \dots, x_{N_T}\}$	All time-varying variables
θ	$\theta = \{\theta^{S_1}, \dots, \theta^{N_S}\}$	Static parameters
$c^i(x^i, x^{i+1})$	$c^i(x^i, x^{i+1}) = \{c_1^i(x_1^i, x_1^{i+1}), \dots, c_{N_T}^i(x_{N_T}^i, x_{N_T}^{i+1})\}$	Biomechanical constraints for joint i at time $t = 1, \dots, N_T$
$c_t(x_t)$	$c_t(x_t) = \{c_t^1(x_t^1, x_t^2), \dots, c_t^{N_S-1}(x_t^{N_S-1}, x_t^{N_S})\}$	Biomechanical constraints at time t
$c(x)$	$c(x) = \{c_t(x_1), \dots, c_t(x_{N_T})\}$	All biomechanical constraints

2 Problem formulation

The problem of estimating the relative pose of each body segment is formulated as a constrained estimation problem. Given N_T measurements $y = \{y_1, \dots, y_{N_T}\}$, a point estimate of the static parameters θ and the time-varying variables $x = \{x_1, \dots, x_{N_T}\}$ can be obtained as a constrained maximum a posteriori (MAP) estimate,

$$\begin{aligned}
 & \underset{x, \theta}{\text{maximize}} && p(x, \theta \mid y) \\
 & \text{subj. to} && c(x) = 0,
 \end{aligned} \tag{1}$$

where $c(x)$ represents the equality constraints. In this work we consider N_S sensors placed on N_S body segments, where sensor S_i is placed on body segment B_i . The time-varying variables x consist of variables both related to sensors (e.g. the pose of the sensor) and to body segments (the pose of the body segment), i.e. $x_t = \{x_t^{S_1}, \dots, x_t^{S_{N_S}}, x_t^{B_1}, \dots, x_t^{B_{N_S}}\}$. To refer to the time-varying variables for sensor S_i , we use the notation $x^{S_i} = \{x_1^{S_i}, \dots, x_{N_T}^{S_i}\}$, while $x^{B_i} = \{x_1^{B_i}, \dots, x_{N_T}^{B_i}\}$ denotes time-varying variables for body segment B_i . The set of time-varying variables pertaining to sensor S_i and segment B_i is denoted $x^i = \{x^{B_i}, x^{S_i}\}$. The static parameters are given by $\theta = \{\theta^{S_1}, \dots, \theta^{N_S}\}$. This notation will be used throughout this work and is summarized in Table 1.

Using the Markov property of the time-varying variables and the fact that the logarithm is a monotonic function, we can rewrite (1) as

$$\begin{aligned}
 \underset{x, \theta}{\text{minimize}} \quad & \underbrace{- \sum_{t=2}^{N_T} \sum_{i=1}^{N_S} \log p(x_t^{S_i} | x_{t-1}^{S_i}, \theta^{S_i}, y_t^{S_i})}_{\text{dynamics of the state } x_t^{S_i}} \quad \underbrace{- \sum_{t=1}^{N_T} \sum_{i=1}^{N_S} \log p(x_t^{B_i} | x_t^{S_i})}_{\text{placement of sensor } S_i \text{ on body segment } B_i} \\
 & \underbrace{- \sum_{i=1}^{N_S} \log p(x_1^{S_i} | y_1^{S_i}) - \sum_{i=1}^{N_S} \log p(\theta^{S_i})}_{\text{prior}} \quad (2)
 \end{aligned}$$

subj. to $c(x) = 0$.

The constraints $c(x)$ represent the connection between the body segments at the joint locations. The cost function consists of terms related to a dynamic model for the time-varying states $x_t^{S_i}$, a model regarding the placement of the sensors on the body segments and a prior on the initial states $x_1^{S_i}$ and the constant parameters θ^{S_i} for $i = 1, \dots, N_S$.

3 Model

To estimate the relative pose of the lower body, we assume that 7 sensors are placed on different body segments. For notational simplicity, we assume that sensor S_i is attached to body segment B_i . The body segments are connected at the joint locations. Figure 2 illustrates two body segments, which can be thought of as the upper leg (B_3) and the lower leg (B_2). A sensor is attached to each body segment and the body segments are connected at the joint J_2 (the knee). Estimating the relative pose of the body amounts to estimating the position and orientation of the sensors and the body segments using the sensor measurements and the information that the body segments are connected. The variables considered optimization problem (2) are given by:

- The time-varying variables $x_t^{S_i}$, consisting of the 3D position, velocity and orientation of sensor S_i at time t . Furthermore, for one of the sensors S_i , the variables $x_t^{S_i}$ also include variables to estimate its mean acceleration at time t .
- The time-varying variables $x_t^{B_i}$ consisting of the 3D position and orientation of body segment B_i at time t .
- The constant variables θ^{S_i} consisting of the gyroscope bias $b_\omega^{S_i} \in \mathbb{R}^3$ of sensor S_i .

Hence, the variables in the optimization problem are $x \in \mathbb{R}^{(15N_S+3)N_T}$ and $\theta \in \mathbb{R}^{3N_S}$, where it is assumed that the orientation variables are encoded using a three-dimensional vector, see e.g. Crassidis et al. (2007); Grisetti et al. (2010);

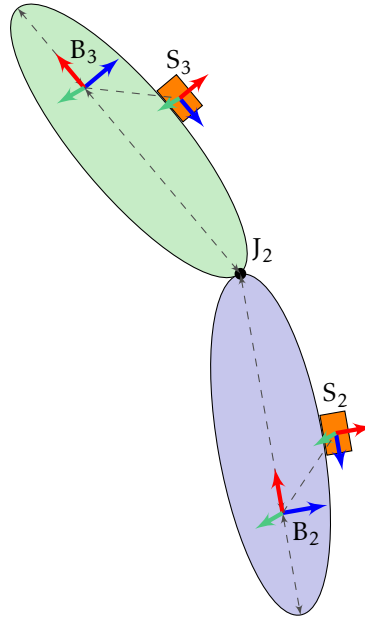


Figure 2: Connection of two body segments with a sensor attached to each of them.

Hol (2011). In the remainder of this section, we discuss the *structure* of the cost functions and of the constraints in (2). This structure will be exploited in our message passing algorithm. For an alternative and more explicit formulation of the problem, we refer to Kok et al. (2014).

3.1 Dynamics of the state $x_t^{S_i}$

The dynamics in (2) expresses the position, velocity and orientation of each sensor S_i in terms of their values at the time instance $t - 1$ and in terms of the constant variables θ^{S_i} . The *change* in position, velocity and orientation of sensor S_i is modeled in terms of the acceleration and angular velocity measured by sensor S_i . The mean acceleration is modeled in terms of $x_{t-1}^{S_i}$, θ^{S_i} and the accelerometer measurements. For more details on the acceleration model we refer to Kok et al. (2014). The dynamics of the state $x_t^{S_i}$ can hence be expressed as in (2).

3.2 Placement of the sensors on the body segments

As shown in Figure 2, sensor S_i is assumed to be attached to the body segment B_i . We assume that the relative position and orientation of the sensors on the body segments is known from calibration. At each time instance, the position and orientation of sensor S_i can therefore be expressed in terms of the position

Table 2: Summary of the body segments, sensors and joints used in the model.

Body segment	Sensor	Joint	Connecting body segments
B ₁ : Right foot	S ₁	J ₁ : Right ankle	B ₁ $\Leftrightarrow$ B ₂
B ₂ : Right lower leg	S ₂	J ₂ : Right knee	B ₂ $\Leftrightarrow$ B ₃
B ₃ : Right upper leg	S ₃	J ₃ : Right hip	B ₃ $\Leftrightarrow$ B ₄
B ₄ : Pelvis	S ₄	J ₄ : Left hip	B ₄ $\Leftrightarrow$ B ₅
B ₅ : Left upper leg	S ₅	J ₅ : Left knee	B ₅ $\Leftrightarrow$ B ₆
B ₆ : Left lower leg	S ₆	J ₆ : Left ankle	B ₆ $\Leftrightarrow$ B ₇
B ₇ : Left foot	S ₇		

and orientation of the body segment B_i . Ideally, this can be incorporated using equality constraints in (2). However, it is physically impossible to place the sensor directly on the bone. Hence, it has to be placed on the soft tissue and the sensor will inevitably move slightly with respect to the bone. To allow for small random movements of the sensor, we incorporate the knowledge about the placement of the sensors on the body segments in the cost function.

3.3 Biomechanical constraints

The constraints $c(x)$ in the optimization problem (2) enforce the body segments to be connected at the joint locations at all times. Hence, for joint J_i , they model the position and the orientation of body segment B_i in terms of the position and the orientation of body segment B_{i+1} for $i = 1, \dots, N_S - 1$. Here, the ordering of the indices of the joints and the body segments is assumed to be as in Table 2. Note that we assume that the length of the body segments is known either from calibration or from a biomechanical model.

Each joint J_i results in a constraint $c_t^i \in \mathbb{R}^3$ at time t . The set of constraints at time t is given by $c_t(x_t) = \{c_t^1(x_t^1, x_t^2), \dots, c_t^{N_S-1}(x_t^{N_S-1}, x_t^{N_S})\}$ and the set of constraints for joint J_i is given by $c^i(x^i, x^{i+1}) = \{c_1^i(x_1^i, x_1^{i+1}), \dots, c_{N_T}^i(x_{N_T}^i, x_{N_T}^{i+1})\}$. The complete set of biomechanical constraints is given by $c(x) = \{c_t(x_1), \dots, c_t(x_{N_T})\}$. This notation is summarized in Table 1. Note that we explicitly indicate which states are involved in the constraints using the ordering of body segments and joints in Table 2.

4 Problem reformulation enabling structure exploitation

In this section we focus on reordering the problem (2) in two different ways. In Section 4.1, we reorder the problem based on the time indices $t = 1, \dots, N_T$. In Section 4.2, we reorder the problem based on sensor and body segment indices $i = 1, \dots, N_S$. The inertial motion capture problem can be solved iteratively using

SQP, where at each iteration k we solve a quadratic approximation of (2). Hence, in each of the sections below, we also introduce an explicit formulation of the quadratic approximation that needs to be solved, where the reordering will allow us to exploit the structure inherent in the problem.

4.1 Reordering based on time

The objective function in (2) can be reordered based on time resulting in

$$\begin{aligned} \underset{x, \theta}{\text{minimize}} \quad & - \sum_{i=1}^{N_S} \left(\log p(x_1^{B_i} | x_1^{S_i}) + \log p(x_1^{S_i} | y_1^{S_i}) + \frac{1}{N_T} \log p(\theta^{S_i}) \right) \\ & - \sum_{t=2}^{N_T} \sum_{i=1}^{N_S} \left(\log p(x_t^{S_i} | x_{t-1}^{S_i}, \theta^{S_i}) + \log p(x_t^{B_i} | x_t^{S_i}) + \frac{1}{N_T} \log p(\theta^{S_i}) \right) \\ \text{subj. to} \quad & c(x) = 0. \end{aligned} \quad (3)$$

Let $f_1(x_1, \theta)$ and $f_t(x_t, x_{t-1}, \theta)$ for $t = 2, \dots, N_T$ correspond to different terms in the cost function of (3). We can then rewrite (3) more compactly as

$$\begin{aligned} \underset{x, \theta}{\text{minimize}} \quad & f_1(x_1, \theta) + \sum_{t=2}^{N_T} f_t(x_t, x_{t-1}, \theta) \\ \text{subj. to} \quad & c_t(x_t) = 0, \quad t = 1, \dots, N_T, \end{aligned} \quad (4)$$

where we use the notation $c_t(x_t)$ to denote the biomechanical constraints at time t as introduced in Table 1. It is beneficial to equivalently reformulate this problem as

$$\begin{aligned} \underset{x, \bar{\theta}}{\text{minimize}} \quad & f_1(x_1, \bar{\theta}_1) + \sum_{t=2}^{N_T} f_t(x_t, x_{t-1}, \bar{\theta}_t) \\ \text{subj. to} \quad & c_t(x_t) = 0, \quad t = 1, \dots, N_T, \\ & \bar{\theta}_t = \bar{\theta}_{t+1}, \quad t = 1, \dots, N_T - 1, \end{aligned} \quad (5)$$

where $\bar{\theta} = \{\bar{\theta}_1, \dots, \bar{\theta}_{N_T}\}$. This formulation models the *constant* variables θ in terms of *time-varying* variables $\bar{\theta}_t$. Inclusion of the additional equality constraints in (5) ensures that $\bar{\theta}_t$ will be equal for all t and makes the formulations (4) and (5) equivalent.

The reordered problem (5) enjoys a desirable structure that can be exploited. It can be solved iteratively using SQP, where at each iteration k we solve the quadratic approximation

$$\begin{aligned} \underset{\Delta x, \Delta \bar{\theta}}{\text{minimize}} \quad & \frac{1}{2} \begin{bmatrix} \Delta x \\ \Delta \bar{\theta} \end{bmatrix}^\top H(x^{(k)}, \bar{\theta}^{(k)}) \begin{bmatrix} \Delta x \\ \Delta \bar{\theta} \end{bmatrix} + \left(J_f(x^{(k)}, \bar{\theta}^{(k)}) \right)^\top \begin{bmatrix} \Delta x \\ \Delta \bar{\theta} \end{bmatrix} \\ \text{subj. to} \quad & c_t(x_t^{(k)}) + \left(J_{c_t}(x_t^{(k)}) \right)^\top \Delta x_t = 0, \quad t = 1, \dots, N_T, \\ & \Delta \bar{\theta}_t - \Delta \bar{\theta}_{t+1} = 0, \quad t = 1, \dots, N_T - 1, \end{aligned} \quad (6)$$

to compute a step, $[\Delta x^\top \Delta \bar{\theta}^\top]^\top$. This step will be used to update the estimates of the variables x and $\bar{\theta}$. The Jacobians of the objective function and of the constraints are given by

$$J_f(x, \bar{\theta}) = \nabla_{x, \bar{\theta}} f_1(x_1, \bar{\theta}_1) + \sum_{t=2}^{N_T} \nabla_{x, \bar{\theta}} f_t(x_t, x_{t-1}, \bar{\theta}_t), \quad (7a)$$

$$J_{c_t}(x_t) = \nabla_{x_t} c_t(x_t). \quad (7b)$$

For the Hessian of the objective function we use a Gauss-Newton approximation as

$$H(x, \bar{\theta}) \approx \nabla_{x, \bar{\theta}} f_1(x_1, \bar{\theta}_1) \nabla_{x, \bar{\theta}} f_1(x_1, \bar{\theta}_1)^\top + \sum_{t=2}^{N_T} \nabla_{x, \bar{\theta}} f_t(x_t, x_{t-1}, \bar{\theta}_t) \nabla_{x, \bar{\theta}} f_t(x_t, x_{t-1}, \bar{\theta}_t)^\top. \quad (8)$$

If we choose the ordering of variables as $(\Delta x_1, \Delta \bar{\theta}_1, \Delta x_2, \Delta \bar{\theta}_2, \dots, \Delta x_{N_T}, \Delta \bar{\theta}_{N_T})$, the Hessian $H(x, \bar{\theta})$ takes a special form as illustrated in Figure 3. In this case it is possible to find matrices H_t and h_t and write the problem in (6) equivalently as

$$\begin{aligned} \underset{\Delta x, \Delta \bar{\theta}}{\text{minimize}} \quad & \sum_{t=1}^{N_T-1} \left(\frac{1}{2} \begin{bmatrix} \Delta x_t \\ \Delta \bar{\theta}_t \\ \Delta x_{t+1} \\ \Delta \bar{\theta}_{t+1} \end{bmatrix}^\top H_t \begin{bmatrix} \Delta x_t \\ \Delta \bar{\theta}_t \\ \Delta x_{t+1} \\ \Delta \bar{\theta}_{t+1} \end{bmatrix} + \begin{bmatrix} \Delta x_t \\ \Delta \bar{\theta}_t \\ \Delta x_{t+1} \\ \Delta \bar{\theta}_{t+1} \end{bmatrix}^\top h_t \right) \\ \text{subj. to} \quad & c_t(x_t^{(k)}) + \left(J_{c_t}(x_t^{(k)}) \right)^\top \Delta x_t = 0, \quad t = 1, \dots, N_T, \\ & \Delta \bar{\theta}_t - \Delta \bar{\theta}_{t+1} = 0, \quad t = 1, \dots, N_T - 1. \end{aligned} \quad (9)$$

This time-ordered equivalent formulation of the problem (2) enjoys a special structure which allows us solve it efficiently using message passing. Before introducing this approach, we will first reorder the problem (2) in a second way, based on sensors and body segments.

4.2 Reordering based on sensors and body segments

The problem (2) can also be rearranged or reordered based on sensors and body segments. Here, we group the terms in the cost function related to sensor S_i and body segment B_i for $i = 1, \dots, N_S$, resulting in

$$\begin{aligned} \underset{x, \theta}{\text{minimize}} \quad & - \sum_{i=1}^{N_S} \left(\log p(x_1^{S_i} | y_1^{S_i}) + \sum_{t=2}^{N_T} \log p(x_t^{S_i} | x_{t-1}^{S_i}, \theta^{S_i}) \right. \\ & \left. + \sum_{t=1}^{N_T} \log p(x_t^{B_i} | x_t^{S_i}) + \log p(\theta^{S_i}) \right) \\ \text{subj. to} \quad & c(x) = 0. \end{aligned} \quad (10)$$

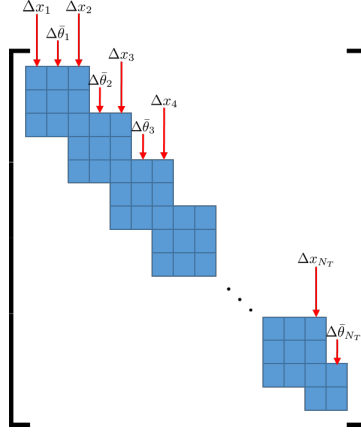


Figure 3: Form of the Hessian $H(x, \bar{\theta})$ for the quadratic approximation (6) with the time-ordered variables as described in Section 4.1. The blue blocks indicate the non-zero terms in the Hessian. For clarity, we indicate which variables are associated with which blocks.

Letting each term in the cost function be denoted by $g^i(x^i, \theta^{S_i})$, we can write (10) compactly as

$$\begin{aligned} & \underset{x, \theta}{\text{minimize}} && \sum_{i=1}^{N_S} g^i(x^i, \theta^{S_i}) \\ & \text{subj. to} && c^i(x^i, x^{i+1}) = 0, \quad i = 1, \dots, N_S - 1, \end{aligned} \quad (11)$$

where the constraints are grouped per joint. Note again that x^i and $c^i(x^i, x^{i+1})$ are defined in Table 1.

Analogously to the development in Section 4.1, solving the problem in (11) using SQP amounts to solving

$$\begin{aligned} & \underset{\Delta x, \Delta \theta}{\text{minimize}} && \frac{1}{2} \begin{bmatrix} \Delta x \\ \Delta \theta \end{bmatrix}^T \bar{H}(x^{(k)}, \theta^{(k)}) \begin{bmatrix} \Delta x \\ \Delta \theta \end{bmatrix} + (J_g(x^{(k)}, \theta^{(k)}))^T \begin{bmatrix} \Delta x \\ \Delta \theta \end{bmatrix} \\ & \text{subj. to} && c^i(x^{i,(k)}, x^{i+1,(k)}) \\ & && + (J_{c^i}(x^{i,(k)}, x^{i+1,(k)}))^T \begin{bmatrix} \Delta x^i \\ \Delta x^{i+1} \end{bmatrix} = 0, \\ & && i = 1, \dots, N_S - 1, \end{aligned} \quad (12)$$

at each iteration, where

$$J_g(x^{(k)}, \theta^{(k)}) = \sum_{i=1}^{N_S} \nabla_{x, \theta} g^i(x^i, \theta^{S_i}), \quad (13a)$$

$$J_{c^i}(x^i, x^{i+1}) = \nabla_{x^i, x^{i+1}} c^i(x^i, x^{i+1}). \quad (13b)$$

The Hessian of the objective function of this problem is again based on a Gauss-Newton approximation,

$$\bar{H}(x, \theta) \approx \sum_{i=1}^{N_S} \nabla_{x, \theta} g^i(x^i, \theta^{S_i}) \nabla_{x, \theta} g^i(x^i, \theta^{S_i})^\top. \quad (14)$$

If we choose the ordering of variables as $(x^1, \theta^{S_1}, x^2, \theta^{S_2}, \dots, x^{N_S}, \theta^{S_{N_S}})$, the Hessian becomes block-diagonal with each block corresponding to sensor S_i and body segment B_i . This then enables us to write the problem in (12) as

$$\begin{aligned} \underset{\Delta x, \Delta \theta}{\text{minimize}} \quad & \sum_{i=1}^{N_S-1} \left(\frac{1}{2} \begin{bmatrix} \Delta x^i \\ \Delta \theta^{S_i} \\ \Delta x^{i+1} \\ \Delta \theta^{S_{i+1}} \end{bmatrix}^\top \bar{H}^i \begin{bmatrix} \Delta x^i \\ \Delta \theta^{S_i} \\ \Delta x^{i+1} \\ \Delta \theta^{S_{i+1}} \end{bmatrix} + \begin{bmatrix} \Delta x^i \\ \Delta \theta^{S_i} \\ \Delta x^{i+1} \\ \Delta \theta^{S_{i+1}} \end{bmatrix}^\top \bar{h}^i \right) \\ \text{subj. to} \quad & c^i(x^{i,(k)}, x^{i+1,(k)}) \\ & + (J_{c^i}(x^{i,(k)}, x^{i+1,(k)}))^\top \begin{bmatrix} \Delta x^i \\ \Delta x^{i+1} \end{bmatrix} = 0, \\ & i = 1, \dots, N_S - 1, \end{aligned} \quad (15)$$

through consistent choices of matrices $\bar{H}^i$ and vectors $\bar{h}^i$. The problem formulation (15) again enjoys a special structure which allows us to solve it efficiently using message passing. Next we briefly review this approach.

5 Tree structure in coupled problems and message passing

Consider the following coupled optimization problem

$$\underset{z}{\text{minimize}} \quad f_1(z) + f_2(z) + \dots + f_{N_C}(z), \quad (16)$$

where $z \in \mathbb{R}^{n_z}$ and $f_a : \mathbb{R}^{n_z} \rightarrow \mathbb{R}$ for $a = 1, \dots, N_C$. This problem can be seen as a combination of N_C subproblems, each of which is defined by a term in the cost function and depends only on a few elements of z . Note that f_a can include indicator functions on constraints. Hence, the problem formulations of the inertial motion capture problem (5), (9) for the time ordering and (11), (15) for the sensor and body segment ordering, are of the form (16).

Let us denote the ordered set of indices of z that each subproblem a depends on by C_a . We can then equivalently rewrite (16) as

$$\underset{z}{\text{minimize}} \quad \bar{f}_1(z_{C_1}) + \dots + \bar{f}_{N_C}(z_{C_{N_C}}), \quad (17)$$

where z_{C_a} is a $|C_a|$ -dimensional vector that contains the elements of z indexed by C_a , with $|C_a|$ denoting the number of elements in the set C_a . Also the functions $\bar{f}_a : \mathbb{R}^{|C_a|} \rightarrow \mathbb{R}$ are lower dimensional descriptions of f_a s such that $f_a(z) = \bar{f}_a(z_{C_a})$

for all z and $a = 1, \dots, N_C$. It is possible to describe the coupling structure of the problem graphically using undirected graphs. Particularly, let us define the *sparsity graph* of the problem as a graph $G_s(V_s, \mathcal{E}_s)$ with the vertex set $V_s = \{1, \dots, n_z\}$ and $(a, b) \in \mathcal{E}_s$ if and only if variables z_a and z_b appear in the same subproblem. Let us assume that each C_a for $a = 1, \dots, N_C$, be a *clique* of this graph, where a clique is a maximal subset of V_s that induces a complete subgraph on G_s . This in turn means that no clique is contained in another clique (Blair and Peyton, 1994). Assume furthermore that there exists a tree defined on C_{G_s} such that for every $C_a, C_b \in C_{G_s}$ where $a \neq b$, $C_a \cap C_b$ is contained in all the cliques in the path connecting the two cliques in the tree. This property is called the clique intersection property (Blair and Peyton, 1994). Graphs with this property have an *inherent tree structure* and can be represented using a *clique tree*.

Let us assume that the sparsity graph of the problem (17) has an inherent tree structure. The problem can then be solved distributedly using a message passing algorithm that utilizes the clique tree as its computational graph. This means that the nodes $V_c = \{1, \dots, N_C\}$ act as computational agents that communicate or collaborate with their neighbors defined by the edge set $\mathcal{E}_c$. The message-passing algorithm solves (17) by performing an upward-downward pass through the clique tree, see e.g., Khoshfetrat Pakazad et al. (2016); Koller and Friedman (2009) and references therein. The upward pass starts from the agents at the leaves of the tree, i.e., all agents $a \in \text{leaves}(T)$, where every such agent computes and communicates the message

$$m_{a \text{ par}(a)}(z_{A_{a \text{ par}(a)}}) = \min_{z_{C_a \setminus A_{a \text{ par}(a)}}} \left\{ \bar{f}_a(z_{C_a}) \right\}, \quad (18)$$

to its parent, denoted by $\text{par}(a)$. Here $A_{ab} = C_a \cap C_b$ is the so-called separator set of agents a and b . Then every agent a that has received all messages from its children, computes and communicates the message

$$m_{a \text{ par}(a)}(z_{A_{a \text{ par}(a)}}) = \min_{z_{C_a \setminus A_{a \text{ par}(a)}}} \left\{ \bar{f}_a(z_{C_a}) + \sum_{b \in \text{ch}(a)} m_{ba}(z_{A_{ba}}) \right\}, \quad (19)$$

with $\text{ch}(a)$ denoting the children of agent a , to its parent. This procedure is continued until we reach the agent, r , at the root. At this point, agent r computes its corresponding optimal solution by solving

$$z_{C_r}^* = \arg \min_{z_{C_r}} \left\{ \bar{f}_r(z_{C_r}) + \sum_{b \in \text{ch}(r)} m_{br}(z_{A_{br}}) \right\}, \quad (20)$$

and initiates the downward pass by communicating this solution to its children. During the downward pass each agent a having received the optimal solution $(z_{A_{\text{par}(a)a}}^*)^{\text{par}(a)}$ from its parent computes its corresponding optimal solution as

$$z_{C_a}^* = \arg \min_{z_{C_a}} \left\{ \bar{f}_a(z_{C_a}) + \sum_{b \in \text{ch}(a)} m_{ba}(z_{A_{ba}}) + \frac{1}{2} \left\| z_{A_{\text{par}(a)a}} - (z_{A_{\text{par}(a)a}}^*)^{\text{par}(a)} \right\|^2 \right\}, \quad (21)$$

Algorithm 1 Distributed Optimization Using Message Passing

```

1: Given a sparsity graph  $G_s$  of an optimization problem
2:   extract its cliques and a clique tree over the cliques;
3:   assign each subproblem to one and only one of the agents.
4: Set agents =  $\{1, \dots, N_C\} \setminus r$  and elim =  $\{\}$ .
5: Perform the upward pass as
6: while |agents|  $\neq 0$  do
7:   for  $i \in$  agents do
8:     if  $\text{ch}(a) \subseteq \text{elim}$  then
9:       This agent computes the message in (19) and communicates it to agent
          $\text{par}(a)$ .
10:      elim = elim  $\cup \{a\}$ .
11:    end if
12:  end for
13:  agents = agents  $\setminus$  elim.
14: end while
15: Set agents =  $\{1, \dots, N_C\}$  and elim =  $\{\}$ .
16: Perform the downward pass as
17: while |agents|  $\neq 0$  do
18:   for  $a \in$  agents do
19:     if  $\text{par}(a) \subseteq \text{elim}$  then
20:       This agent computes optimal solution as in (21) and communicates it to
         agents  $\text{ch}(a)$ .
21:      elim = elim  $\cup \{a\}$ .
22:    end if
23:  end for
24:  agents = agents  $\setminus$  elim.
25: end while
26: By the end of the downward pass all agents have computed their optimal solutions
    and the algorithm is terminated.

```

and communicates this solution to its children. Once the downward pass is accomplished, all agents have computed their respective optimal solution and the algorithm is terminated. We have summarized this scheme in Algorithm 1.

Remark 1. Notice that within the upward pass all agents that have received messages from their children can compute their messages simultaneously and in parallel. This also holds for the downward pass, as all agents that have received the optimal solution from their parents can compute their respective optimal solution in parallel.

6 Scalable and distributed solutions using message passing

We will now rewrite the problem reorderings (9) and (15) such that Algorithm 1 can be used to solve the problem. Let us first reconsider the problem in (9). We can rewrite this problem compactly as

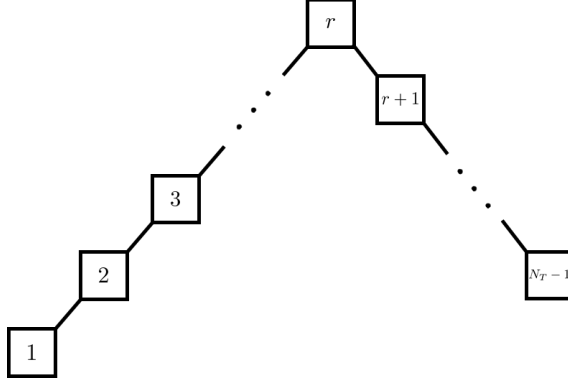


Figure 4: Clique tree corresponding to the sparsity graph of problem (9).

$$\begin{aligned} & \underset{\Delta x, \Delta \bar{\theta}}{\text{minimize}} && \sum_{t=1}^{N_T-1} \bar{f}_t(\Delta x_t, \Delta x_{t+1}, \Delta \bar{\theta}_t, \Delta \bar{\theta}_{t+1}) \\ & \text{subj. to} && \end{aligned} \quad (22a)$$

$$\left. \begin{aligned} & c_t(x_t^{(k)}) + \left(J_{c_t}(x_t^{(k)}) \right)^\top \Delta x_t = 0 \\ & \Delta \bar{\theta}_t - \Delta \bar{\theta}_{t+1} = 0 \end{aligned} \right\}, t = 1, \dots, r-1, \quad (22b)$$

$$\left. \begin{aligned} & c_t(x_t^{(k)}) + \left(J_{c_t}(x_t^{(k)}) \right)^\top \Delta x_t = 0 \\ & \Delta \bar{\theta}_t - \Delta \bar{\theta}_{t+1} = 0 \end{aligned} \right\}, t = r, \quad (22c)$$

$$\left. \begin{aligned} & c_t(x_{t+1}^{(k)}) + \left(J_{c_{t+1}}(x_{t+1}^{(k)}) \right)^\top \Delta x_{t+1} = 0 \\ & c_t(x_{t+1}^{(k)}) + \left(J_{c_{t+1}}(x_{t+1}^{(k)}) \right)^\top \Delta x_{t+1} = 0 \\ & \Delta \bar{\theta}_t - \Delta \bar{\theta}_{t+1} = 0 \end{aligned} \right\}, t = r+1, \dots, N_T-1, \quad (22d)$$

where $r = \lfloor N_T/2 \rfloor$. The sparsity graph of this problem has an inherent tree structure, with $N_T - 1$ cliques. Each clique C_a consists of the variables Δx_a , $\Delta \bar{\theta}_a$, Δx_{a+1} and $\Delta \bar{\theta}_{a+1}$. The clique tree for this problem is illustrated in Figure 4. Consequently, we can use Algorithm 1 for solving this problem. During the upward pass, each agent a sends a message as in (18) and (19) to its parent as a function of the variables it shares with its parents. Hence, if $a < r$ (the agent is on the left side of agent r in Figure 4) the message to its parents is a function of Δx_{a+1} and $\Delta \bar{\theta}_{a+1}$. Equivalently, if $a > r$ (the agent is on the right side of agent r in Figure 4) the message to its parents is a function of Δx_a and $\Delta \bar{\theta}_a$. As a result, each agent except agent r has to factorize a matrix of size $|x_a| + |\bar{\theta}_a|$ plus the number of constraints, which is equal to $6N_S - 3$, as can be seen in (22). The root agent instead has to factorize a matrix of size $2|x_a| + 2|\bar{\theta}_a| + 9N_S - 6$. The computational complexity of

Algorithm 2 Inertial Motion Capture

-
- 1: Place the sensors on the body, calibrate the system and collect inertial measurements.
 - 2: Initialize x and θ or $\bar{\theta}$ and set $k = 0$.
 - 3: **while** the algorithm has not converged and the solution is not feasible **do**
 - 4: Formulate the quadratic approximation (22) using the time reordering or (15) using the sensor / segment reordering.
 - 5: Use Algorithm 1 to solve the problem formulated in Step 4 and to obtain a step $\begin{bmatrix} \Delta x^\top & \Delta \bar{\theta}^\top \end{bmatrix}^\top$ for the time ordered problem or a step $\begin{bmatrix} \Delta x^\top & \Delta \theta^\top \end{bmatrix}^\top$ for the sensor / segment ordered problem.
 - 6: Update $x := x + \Delta x$, $\theta := \theta + \Delta \theta$ or $\bar{\theta} := \bar{\theta} + \Delta \bar{\theta}$.
 - 7: Set $k := k + 1$ and check for convergence and feasibility.
 - 8: **end while**
-

Algorithm 1 is dominated by the upward pass since the downward pass does not require a matrix factorization. For details on this, we refer to Khoshfetrat Pakazad et al. (2016). Hence, the computational complexity and storage requirements for the resulting algorithm grow linearly with N_T . The reduction in the memory requirements follows from the fact that using Algorithm 1 we have relaxed the need for even forming the problem in (9). The resulting algorithm to solve the problem (22) is summarized in Algorithm 2.

The problem in (15) is also a coupled problem but with $N_S - 1$ subproblems. The clique tree for this problem has the same structure as for (15), where the only differences are in the number of cliques which in this case is $N_S - 1$ and that $r = \lfloor N_S/2 \rfloor$. Each clique C_a consists of the variables Δx^a , $\Delta \theta^{S_a}$, Δx^{a+1} and $\Delta \theta^{S_{a+1}}$. Hence, we can solve the problem in (15) distributedly using Algorithm 1. This can be achieved using a network of computational agents, that can be installed on the body and that collaborate based on the clique tree.

Remark 2. Note that in (22) we have adopted a particular grouping of the constraints. This is to ensure that each of the subproblems is well-posed in terms of their local variables. This was not necessary for the problem in (15). _____

Remark 3. The reason that the clique trees for both problems in (22) and (15) have the same structure is due to the fact that we have focused on the motion capture problem for the lower body. For solving the full body problem we can use the same approach as presented in this paper, since the inherent tree structure will still be present in the problem. However, the clique tree for the corresponding problem will be more complicated than a chain and will correspond to the body formation. _____

7 Results and discussion

We consider experimental data from a subject walking around for approximately 37 seconds wearing inertial sensors as shown in Figure 5. We focus on estimating the pose of the lower body using data from 7 sensors attached to the different

body segments as described in Section 3. The estimated joint angles from this problem have previously been presented in Kok et al. (2014). In this work, we solve the same optimization problem but reorder the problem to efficiently make use of its structure. Hence, the estimates obtained using Algorithm 2 are equal to the ones presented in Kok et al. (2014).

The optimization problem is solved at 10 Hz with $N_T = 373$, leading to a total number of 40284 time-varying variables x and 21 constant variables θ and 6714 constraints.¹ Notice that if the inherent sparsity of the problem would not be exploited, the computational complexity of solving the SQP for the smoothing problem (6) or (12) would grow cubically with the number of sensors and body segments N_S and with the number of time steps N_T . The storage requirements for forming this problem would grow quadratically with N_T and N_S .

To solve the problem in a more scalable manner, we have reordered the variables based on time and formed the problem as in (22), which allows us to solve the problem using Algorithm 2. For each iteration k in Algorithm 2, we then form $N_T - 1$ subproblems. Computing the messages in the upward pass requires each agent except the root agent to factorize a matrix of size 168 since $|x_t| + |\bar{\theta}_t| = 129$ and 39 constraints are involved in the subproblem. The root agent needs to instead factorize a matrix of size 315 since $2|x_t| + 2|\bar{\theta}| = 258$ and 57 constraints are involved in this subproblem instead. Using message passing to solve the problem, it is no longer required to form and store the large problem of size 46998. Instead, it is only required to store one of these subproblems.

We have also solved the problem by reordering the variables based on sensors and segments. The computational benefits of this reordering are much less significant – the problem is split up in 6 subproblems. However, the approach no longer requires collecting all data at a centralized unit, which can be communication intensive, and hence can potentially hamper our ability to have a seamless solution for the motion capture problem. Instead, it allows for decentralized computation of the solution, where the computational power on the sensors can be used to compute solutions to the subproblems.

8 Conclusions and future work

In this work, we have introduced a method to exploit the structure inherent in the inertial motion capture problem. The method allows for a scalable solution where small subproblems for each time step are formed and hence longer data sets can be processed. The approach is successfully applied to experimental data to estimate the pose of the lower body. It also opens up for the possibility of distributedly solving the problem by making use of the computational resources of each of the sensors. The structure that we exploit in this work is not unique to the motion capture problem. We believe that the message passing algorithm can be applied to a large number of other problems appearing in signal processing and

¹Note that the inertial sensors themselves are sampled at a much higher rate but strapdown integration (Savage, 1998a,b) is used to capture the high frequency signals, allowing for a lower update frequency of the optimization problem.



Figure 5: Experimental setup where the human body is equipped with inertial sensors (orange boxes) on different body segments. High-accuracy reference measurements were obtained using an optical tracking system to validate the estimated joint angles. To this end, triangles with optical markers were placed on a number of sensors.

estimation, e.g., in large-scale signal processing and estimation application. This is because these problems commonly enjoy desirable sparsity structures arising from physical and / or dynamic properties in the problem, as we saw for the inertial motion capture problem in this work.

Acknowledgments

This work is supported by CADICS, a Linnaeus Center and by the project Probabilistic modeling of dynamical systems (Contract number: 621-2013-5524), both funded by the Swedish Research Council (VR) and by the Swedish Department of Education within the ELLIIT project.

Bibliography

- D. P. Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena Scientific Belmont, MA, 3rd edition, 1995.
- J. R. S. Blair and B. W. Peyton. An introduction to chordal graphs and clique trees. In J. A. George, J. R. Gilbert, and J. W-H. Liu, editors, *Graph Theory and Sparse Matrix Computations*, volume 56, pages 1–27. Springer-Verlag, 1994.
- V. Cevher, S. Becker, and M. Schmidt. Convex optimization for big data: Scalable, randomized, and parallel algorithms for big data analytics. *IEEE Signal Processing Magazine*, 31(5):32–43, Sept 2014.
- J. L. Crassidis, F. L. Markley, and Y. Cheng. A survey of nonlinear attitude estimation methods. *Journal of Guidance, Control, and Dynamics*, 30(1):12–28, 2007.
- G. Grisetti, R. Kümmerle, C. Stachniss, U. Frese, and C. Hertzberg. Hierarchical optimization on manifolds for online 2D and 3D mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 273–278, Anchorage, Alaska, 2010.
- J. D. Hol. *Sensor fusion and calibration of inertial sensors, vision, ultra-wideband and GPS*. Dissertation no. 1368, Linköping University, Linköping, Sweden, June 2011.
- D. H. Kang, Y. J. Jung, A. J. Park, and J. W. Kim. Human body motion capture system using magnetic and inertial sensor modules. In *Proceedings of the 5th international universal communication symposium (IUCS)*, Gumi, Korea, October 2011.
- S. Khoshfetrat Pakazad, A. Hansson, M. S. Andersen, and I. Nielsen. Distributed primal-dual interior-point methods for solving tree-structured coupled convex problems using message-passing. *Optimization Methods and Software*, pages 1–35, 2016.
- M. Kok, J. D. Hol, and T. B. Schön. An optimization-based approach to human body motion capture using inertial sensors. In *Proceedings of the 19th World Congress of the International Federation of Automatic Control*, pages 79–85, Cape Town, South Africa, August 2014.
- M. Kok, S. Khoshfetrat Pakazad, T. B. Schön, A. Hansson, and J. D. Hol. A scalable and distributed solution to the inertial motion capture problem. In *Proceedings of the 19th International Conference on Information Fusion*, pages 1348–1355, Heidelberg, Germany, July 2016.
- D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT press, 2009.
- J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research, 2nd edition, 2006.

- C. V. Rao, J. B. Rawlings, and J. H. Lee. Constrained linear state estimation - a moving horizon approach. *Automatica*, 37(10):1619–1628, 2001.
- H. E. Rauch, C. T. Striebel, and F. Tung. Maximum likelihood estimates of linear dynamic systems. *AIAA journal*, 3(8):1445–1450, 1965.
- D. Roetenberg, H. J. Luinge, and P. Slycke. Xsens MVN: Full 6DOF human motion tracking using miniature inertial sensors, May 2013.
- P. G. Savage. Strapdown inertial navigation integration algorithm design part 1: Attitude algorithms. *Journal of Guidance, Control and Dynamics*, 21(1):19–28, 1998a.
- P. G. Savage. Strapdown inertial navigation integration algorithm design part 2: Velocity and position algorithms. *Journal of Guidance, Control and Dynamics*, 21(2):208–221, 1998b.
- S. J. Wright. *Primal-dual interior-point methods*. SIAM, 1997.
- Xsens Technologies B. V. <http://www.xsens.com>, 2016. Accessed on March 5.
- X. Yun and E. R. Bachmann. Design, implementation, and experimental results of a quaternion-based Kalman filter for human body motion tracking. *IEEE Transactions on Robotics*, 22(6):1216–1227, 2006.

Paper D

Indoor positioning using ultrawideband and inertial measurements

Authors: Manon Kok, Jeroen D. Hol and Thomas B. Schön

Edited version of the paper:

M. Kok, J. D. Hol, and T. B. Schön. Indoor positioning using ultrawideband and inertial measurements. *IEEE Transactions on Vehicular Technology*, 64(4):1293–1303, 2015b.

Indoor positioning using ultrawideband and inertial measurements

Manon Kok^{*}, Jeroen D. Hol[†] and Thomas B. Schön[‡]

^{*}Dept. of Electrical Engineering,
Linköping University,
SE-581 83 Linköping, Sweden
manko@isy.liu.se

[†]Xsens Technologies B. V.
P.O. Box 559,
7500 AN Enschede, the Netherlands
jeroen.hol@xsens.com

[‡]Dept. of Information Technology
Uppsala University,
SE-751 05 Uppsala, Sweden
thomas.schon@it.uu.se

Abstract

In this work we present an approach to combine measurements from inertial sensors (accelerometers and gyroscopes) with time of arrival measurements from an ultrawideband system for indoor positioning. Our algorithm uses a tightly-coupled sensor fusion approach, where we formulate the problem as a maximum a posteriori problem that is solved using an optimization approach. It is shown to lead to accurate 6D position and orientation estimates when compared to reference data from an independent optical tracking system. To be able to obtain position information from the ultrawideband measurements, it is imperative that accurate estimates of the ultrawideband receivers' positions and their clock offsets are available. Hence, we also present an easy-to-use algorithm to calibrate the ultrawideband system using a maximum likelihood formulation. Throughout this work, the ultrawideband measurements are modeled by a tailored heavy-tailed asymmetric distribution to account for measurement outliers. The heavy-tailed asymmetric distribution works well on experimental data, as shown by analyzing the position estimates obtained using the ultrawideband measurements via a novel multilateration approach.

1 Introduction

Ultra-wideband (UWB) is a relatively new and promising radio technology with applications in for example radar, communication and localization. UWB technology typically makes use of impulse radio with very short pulses. These are typically in the order of 1 ns, opening up for high spatial resolution. This characteristic makes UWB very suitable for localization purposes. It has successfully been applied in

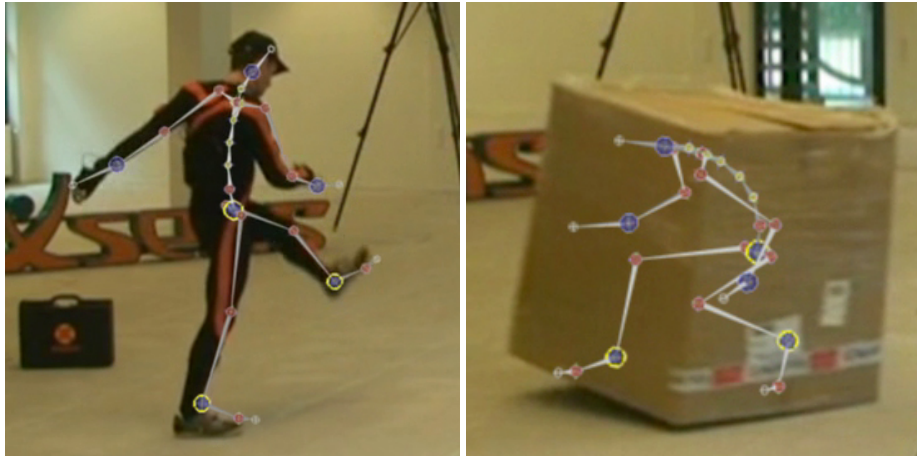


Figure 1: Example application showing a subject with 17 inertial sensors placed on the body and 3 UWB transmitters placed on the head and on both feet. The pose estimates are visualized by the “skeleton” overlaid in the images. As shown (right), the solution remains valid even in non-line-of-sight conditions.

a wide variety of localization applications, such as industrial (Ubisense, 2014), health-care (Time Domain, 2014; Gezici et al., 2005) and motion capture (Xsens Technologies B.V., 2014). UWB positioning accuracy is reported to be in the order of decimeters (Time Domain, 2014; Ubisense, 2014). Although UWB systems do not necessarily require line-of-sight visibility (Bellusci et al., 2011), the UWB measurements do suffer from multipath and non-line-of-sight (NLOS) conditions, resulting in measurement outliers.

Inertial sensors consist of accelerometers and gyroscopes measuring the acceleration and angular velocity of the sensor. The inertial sensor measurements need to be integrated to obtain position and orientation estimates. These position and orientation estimates are accurate on a short time scale, but suffer from integration drift. Inertial sensors have successfully been used to estimate 6D position and orientation (pose) in combination with systems providing position information such as GPS and UWB, see e.g. Hol (2011) and the references therein.

In this work we present an indoor positioning approach using inertial sensors and time of arrival (TOA) measurements from an UWB system. We use a setup where a number of UWB receivers are placed in an indoor environment. Our focus is on combining information from an UWB transmitter and an inertial measurement unit (IMU) to estimate the 6D pose of the IMU. The IMU and the transmitter are assumed to be rigidly attached to each other. Our approach can be extended to for example estimate the 6D pose of a human body where a subject wearing multiple IMUs and multiple UWB transmitters walks through the UWB measurement volume, as shown in Figure 1. This work builds on Hol et al. (2009); Hol (2011), where an extended Kalman filter (EKF) was used in combining the inertial

and UWB measurements to estimate the 6D pose of the sensor. Outlier rejection was used to remove UWB measurements affected by multipath and NLOS. In this work we instead solve the problem using an optimization-based approach similar to the approach used in Kok et al. (2014). Using this approach, it is possible to assume more general measurement distributions. Hence, we model the UWB measurements using a heavy-tailed asymmetric distribution which is specifically tailored for this particular application. This distribution naturally handles the possibility of measurement *delays* due to multipath and NLOS while not allowing for the possibility of measurements arriving *earlier*, i.e. traveling faster than the speed of light. We will show that accurate position and orientation estimates are obtained by comparing our results to those obtained from an independent optical reference system.

To be able to obtain position information from the UWB measurements, the positions of the receivers must be known and the receiver clocks have to be synchronized. To avoid the typically labor-intensive and time-consuming process of manually surveying the receiver positions, we present an easy-to-use calibration method that automates this process. Our previous solution presented in Hol et al. (2010) assumed “clean” measurements, i.e. it was assumed that no outliers were present. In this work, we will instead solve the calibration problem modeling the UWB measurements using the heavy-tailed asymmetric distribution mentioned earlier to naturally handle the measurement outliers.

To experimentally validate the UWB model using the heavy-tailed asymmetric distribution, we will use the UWB measurements in a novel multilateration approach to determine the position of a mobile transmitter. Here, we assume that the UWB system has previously been calibrated using our calibration algorithm. We will show that the position estimates obtained using the heavy-tailed asymmetric distribution are considerably better than the ones obtained using either a Gaussian distribution or a heavy-tailed symmetric Cauchy distribution.

In Section 2, we provide more background to our work by relating it to previous work in the area. In Section 3 we clearly formulate the problem. The sensors and their corresponding measurement models are introduced in Section 4. The UWB calibration problem is subsequently solved in Section 5. The solution to the sensor fusion problem, where we also make use of the inertial measurements can be found in Section 6. The experimental results and the conclusions are then provided in Section 7 and Section 8, respectively.

2 Related work

In this work we make use of TOA measurements from an UWB system. Our UWB setup consists of a network of synchronized and stationary (rigidly fixed, mounted) receivers, all acquiring very precise TOA measurements of signals originating from a mobile transmitter. The low-cost transmitters in our setup have an inaccurate clock and can hence not provide accurate information concerning the time of transmission. For a general introduction to UWB technology and its use in positioning applications, see e.g. Gezici et al. (2005); Sahinoglu et al. (2008).

The process of determining the transmitter position from TOA measurements is referred to as trilateration or, more accurately, *multilateration*. It is a well-studied topic and many algorithms have been reported in the literature, see e.g. Chan and Ho (1994); Gezici et al. (2005); Sayed et al. (2005); Sahinoglu et al. (2008). A common multilateration technique is to eliminate the time of transmission by constructing time difference of arrival (TDOA) measurements from pairs of TOA measurements. The resulting set of hyperbolic equations can then be solved for position (Gustafsson, 2012). The drawback of this approach is that the constructed TDOA measurements are no longer independently distributed which complicates the calculations. In this work we use a well-known equivalent approach where we instead model the time of transmission as an unknown quantity.

Ideally, the UWB signal travels directly from the transmitter to the different receivers. In that case, the TOA measurements are directly related to the distance traveled. In case the signal encounters a medium which delays or reflects the signal, however, the time of flight is prolonged and the pulse will be delayed. This can result in large estimation errors when assuming that the UWB measurements are Gaussian distributed, as will be illustrated in Section 7.1. The problem of how to robustly deal with outliers in the measurements has received a lot of attention, see e.g. Zoubir et al. (2012) for a good survey containing relevant entry-points into the literature on this topic commonly referred to as robust statistics. A common approach is to model the outliers in terms of the probability of NLOS and introduce a delay represented by a heavy-tailed positive-mean probability density function (PDF) such as a shifted Gaussian or an exponential, see e.g. Alsindi et al. (2009); Gustafsson and Gunnarsson (2005). Based on Alsindi et al. (2009), the localization approach presented in Prorok and Martinoli (2014) builds spatial models representing the probability of NLOS in a certain area. In our approach we do not specifically model the probability of the number of outliers. Instead, we model the UWB measurements using a specifically tailored asymmetric heavy-tailed noise distribution. By estimating the width of this distribution, the algorithm can automatically adapt the width to the specific measurement data. The use of this distribution can straightforwardly be incorporated in any maximum likelihood (ML) or maximum a posteriori (MAP) estimation algorithm, without including any additional parameters.

UWB approaches typically assume that the receiver positions are known and that their clocks are synchronized, explaining why there are relatively few UWB calibration algorithms available in the literature, see e.g. Hol (2011); Hol et al. (2009). However, ideas for UWB calibration can be obtained from the wide range of literature on sensor localization, see for instance Patwari et al. (2005). The challenge again comes down to the possibility of measurement errors due to NLOS and/or multipath, resulting in a reduced quality of the calibration results. Hence, we assume a tailored asymmetric heavy-tailed distribution in our calibration algorithm to represent these errors.

When combining inertial measurements with UWB measurements, a tightly-coupled or a loosely-coupled approach can be used. In a loosely-coupled approach, the UWB measurements are used to obtain position estimates using a multilateration approach. These position estimates are subsequently used as artificial posi-

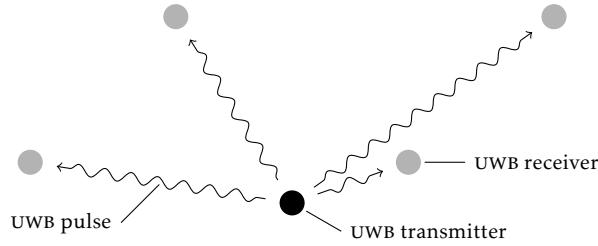


Figure 2: The UWB setup consists of a number of stationary receivers acquiring TOA measurements of signal pulses originating from a mobile transmitter.

tion measurements in the sensor fusion approach, see e.g. Sczyslo et al. (2008); Pit-tet et al. (2008); Corrales et al. (2008). A tightly-coupled approach instead makes direct use of each individual TOA measurement, see for instance De Angelis et al. (2010); Hol et al. (2009). An advantage of a tightly-coupled approach is that it does not suffer from the loss of information that typically arises from pre-processing of the UWB measurements that has to be performed in a loosely-coupled approach. This is mainly due to approximations of statistical distributions, but in extreme cases measurements are also ignored, for instance when there are not enough TOA measurements for multilateration. By instead making direct use of the sensor measurements, we can make maximal use of the available information. The advantage of a tightly-coupled approach is experimentally shown for the case of UWB measurements in for instance Zwirello et al. (2013).

One way to solve the sensor fusion problem is then to use an EKF. To allow for the presence of delayed measurements, approaches based on outlier rejection (Hol, 2011; Hol et al., 2009) and robust EKF formulations (Hammes et al., 2009) have previously been used. In this work we will instead formulate a tightly-coupled approach by formulating and solving an optimization problem. This also straightforwardly opens up for exploiting non-Gaussian distributions and for estimating additional parameters.

3 Problem formulation

To determine an IMU's 6D position and orientation, its measurements are combined with those originating from an UWB transmitter that is rigidly attached to the sensor. The UWB transmitter sends pulses to a number of stationary UWB receivers as illustrated in Figure 2. The receivers measure the times of arrival of the pulses at their different locations. Provided that the receiver positions are known and that their clocks are synchronized, the position of the transmitter can be inferred from these measurements. Although the receivers are synchronized to a central clock, they each have a small, constant clock offset due to for instance differences in cable lengths. The receiver positions and clock offsets are computed using our calibration algorithm.

Let us denote the model parameters estimated by the calibration algorithm by θ . The calibration algorithm computes an ML estimate $\hat{\theta}^{\text{ML}}$. For a setup with $m = 1, \dots, M$ receivers and $l = 1, \dots, L$ transmitters, the UWB measurements are denoted $y_{u,mlk}$ for $k = 1, \dots, K$ UWB pulses. Note that the *absolute* receiver positions and clock offsets are neither observable, nor relevant. We are interested in estimating the receivers' *relative* positions and clock offsets. Hence, an arbitrary choice of an UWB coordinate frame and a reference clock offset are used as constraints in the ML problem. The problem of calibrating the UWB setup is therefore formulated as a constrained ML problem according to

$$\begin{aligned} \max_{\theta \in \Theta} \quad & \prod_{k=1}^K \prod_{l=1}^L \prod_{m=1}^M p_{\theta}(y_{u,mlk}), \\ \text{s.t.} \quad & A\theta = b, \end{aligned} \quad (1)$$

where $p_{\theta}(y)$ denotes the PDF of y parametrized by θ . The matrix A and the vector b are used to describe the linear constraints on the parameter vector θ , due to the choice of the coordinate frame and the reference clock offset. The calibration problem (1) is solved in Section 5 where we also provide the resulting calibration algorithm. In that section, the constraints in (1) are also defined more explicitly.

When the UWB system has been calibrated, the UWB measurements can be combined with inertial measurements to determine the 6D pose of the sensor. Hence, we estimate the state vector $x_{1:N}$ which contains the position of the sensor, its orientation and additional information. Denoting the accelerometer measurements by $y_{a,t}$ for $t = 1, \dots, N$, the gyroscope measurements by $y_{\omega,t}$ and the UWB measurements by $y_{u,mk}$ for $k = 1, \dots, K$ pulses and $m = 1, \dots, M$ receivers, the state is computed by solving the following MAP problem,

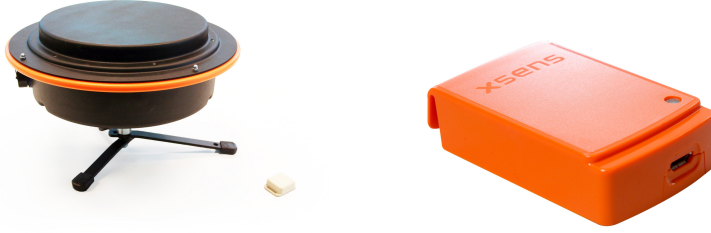
$$\max_{x_{1:N}} p(x_{1:N} \mid \{y_{a,t}, y_{\omega,t}\}_{t=1}^N, \{\{y_{u,mk}\}_{m=1}^M\}_{k=1}^K). \quad (2)$$

Here, $p(x_{1:N} \mid y_{1:N})$ denotes the conditional PDF of the state vector $x_{1:N}$ given the measurements $y_{1:N}$. The subscript k used for the UWB measurements indicates that the UWB measurements do not necessarily have the same sampling frequency as the inertial measurements. The solution to the sensor fusion problem (2) is provided in Section 6, together with the resulting algorithm.

4 Sensor models

Our UWB system consists of a network of stationary receivers which can track a large number of small, battery-powered inexpensive transmitters (Time Domain, 2014). A transmitter and a receiver are shown in Figure 3a. In our sensor fusion approach, we combine UWB measurements with inertial measurements. The IMU is shown in Figure 3b.

In this section we will introduce our sensor models, starting with the UWB measurement model in Section 4.1. Subsequently, the inertial measurement models will be introduced in Section 4.2.



(a) Hardware used in an UWB setup. (b) An IMU containing a 3-axis accelerometer and a 3-axis gyroscope. More specifically, an UWB receiver and a small, battery-powered UWB transmitter.

Figure 3: Hardware used to collect UWB and inertial measurements.

4.1 Modeling the ultrawideband measurements

For the UWB setup with $m = 1, \dots, M$ receivers and $l = 1, \dots, L$ transmitters, the TOA measurement $y_{u,mlk}$ of receiver m , originating from transmitter l and pulse k , is modeled as

$$y_{u,mlk} = \tau_{lk} + \frac{1}{c} \|r_m^n - t_{lk}^n\|_2 + \Delta\tau_m + e_{u,mlk}. \quad (3)$$

Here, c denotes the speed of light, τ_{lk} is the time of transmission of pulse k from transmitter l , t_{lk}^n is the position of transmitter l at the time of transmitting the k^{th} pulse, r_m^n is the position of the m^{th} receiver and $\Delta\tau_m$ is its clock-offset. The superscript n denotes the navigation frame. It is a local coordinate frame that is aligned with the earth's gravity and with the axes of the frame defined during the UWB calibration, as already discussed in Section 3.

Due to NLOS conditions and/or multipath we expect a small number of measurements to be delayed. Hence, it does not make sense to model $e_{u,mlk}$ using a Gaussian distribution. In Hol (2011), a new multilateration approach was presented, where the possibility of delayed measurements was modeled by including a positive parameter $\delta_{u,mlk}$ explicitly representing the delay of pulse k to receiver m in the measurement equation (3). The parameters $\delta_{u,mlk}$ were assumed to have an exponential prior. Hence, Hol (2011) models the delay of each pulse to each receiver as a parameter to be estimated. This was shown to lead to accurate position estimates, but it also introduced M additional model parameters for each pulse k .

In this work, we omit the parameters $\delta_{u,mlk}$ and instead model the possibility of delays in terms of the distribution of the noise $e_{u,mlk}$. We assume an asymmetric distribution where a heavy-tailed Cauchy distribution allows for measurement delays while a Gaussian distribution excludes the physically unreasonable possibility of pulses traveling faster than the speed of light as

$$e_{u,mlk} \sim \begin{cases} (2 - \alpha) \mathcal{N}(0, \sigma^2) & \text{for } e_{u,mlk} < 0 \\ \alpha \text{Cauchy}(0, \gamma) & \text{for } e_{u,mlk} \geq 0. \end{cases} \quad (4a)$$

$$(4b)$$

The presence of the constants α and $2 - \alpha$ is motivated by the fact that the proposed asymmetric PDF needs to integrate to one and hence

$$\begin{aligned} \int_{-\infty}^0 (2 - \alpha) \frac{1}{\sqrt{2\pi}\sigma^2} \exp\left(-\frac{e_{u,mlk}^2}{2\sigma^2}\right) d e_{u,mlk} + \\ \int_0^{\infty} \alpha \frac{1}{\pi\gamma} \left(\frac{1}{1 + \frac{e_{u,mlk}^2}{\gamma^2}} \right) d e_{u,mlk} = \frac{1}{2} (2 - \alpha) + \frac{1}{2} \alpha = 1, \end{aligned} \quad (5)$$

where we have made use of the fact that the Gaussian and Cauchy PDFs integrate to one and are symmetric. Imposing the constraint that the distribution is continuous at $e_{u,mlk} = 0$ allows us to express α in terms of σ and γ according to

$$\frac{\alpha}{\pi\gamma} = \frac{2-\alpha}{\sqrt{2\pi}\sigma^2} \Leftrightarrow \alpha = \frac{2\pi\gamma}{\sqrt{2\pi}\sigma^2 + \pi\gamma}. \quad (6)$$

The proposed asymmetric PDF and its corresponding negative log-likelihood, given by

$$-\log p(e_{u,mlk}) = \begin{cases} \mathcal{L}_G & \text{for } e_{u,mlk} < 0, \\ \mathcal{L}_C & \text{for } e_{u,mlk} \geq 0, \end{cases} \quad (7a)$$

$$\begin{aligned} \mathcal{L}_G &\triangleq \frac{e_{u,mlk}^2}{2\sigma^2} + \frac{1}{2} \log \sigma^2 + \frac{1}{2} \log 2\pi - \log(2 - \alpha), \\ \mathcal{L}_C &\triangleq \log\left(1 + \frac{e_{u,mlk}^2}{\gamma^2}\right) + \frac{1}{2} \log \gamma^2 + \log \pi - \log \alpha, \end{aligned} \quad (7b)$$

are both depicted in Figure 4 in red. For comparison, the Gaussian and Cauchy PDFs are also depicted, in blue and green, respectively.

From the experimental results in Section 7.1 it will be shown that exploiting the asymmetry of the actual noise distribution is especially helpful in the presence of a large number of outliers.

4.2 Modeling the inertial measurements

An IMU containing a 3-axis accelerometer and a 3-axis gyroscope was shown in Figure 3b. The inertial measurements are resolved in the body frame b . Its origin lies in the center of the accelerometer triad and its axes are aligned with the casing. The gyroscope measures the sensor's angular velocity ω_t . Its measurements $y_{\omega,t}$ are modeled as

$$y_{\omega,t} = \omega_t + \delta_{\omega} + e_{\omega,t}, \quad (8)$$

where δ_{ω} denotes the gyroscope bias and $e_{\omega,t} \sim \mathcal{N}(0, \Sigma_{\omega})$. We assume $\Sigma_{\omega} = \sigma_{\omega}^2 \mathcal{I}_3$ where $\mathcal{I}_3$ denotes the 3×3 identity matrix. The accelerometer measures the external specific force f_t^b exerted on the sensor. It consists of the sensor's linear

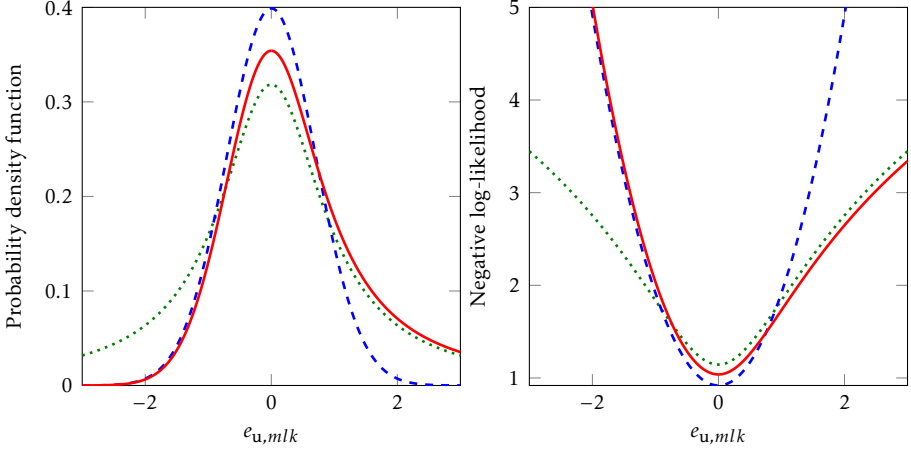


Figure 4: PDF (top) and negative log-likelihood (bottom) of a $\mathcal{N}(0, 1)$ distribution (blue, dashed), a $\text{Cauchy}(0, 1)$ distribution (green, dotted) and the asymmetric distribution (4) assuming $\sigma = \gamma = 1$ and α according to (6) (red).

acceleration a_t^n and the gravity vector g^n , both resolved in the navigation frame n . The accelerometer measurements $y_{a,t}$ are modeled as

$$y_{a,t} = f_t^b + \delta_a + e_{a,t} = R_t^{bn} (a_t^n - g^n) + \delta_a + e_{a,t}, \quad (9)$$

where δ_a denotes the accelerometer bias and $e_{a,t} \sim \mathcal{N}(0, \Sigma_a)$. We assume $\Sigma_a = \sigma_a^2 \mathcal{I}_3$. The rotation matrix R_t^{bn} represents the rotation from the navigation frame n to the body frame b at time t .

Both the gyroscope and the accelerometer biases δ_ω and δ_a are slowly time-varying, but we will treat them as constants motivated by the short experimental times used in this work. For longer experiments, δ_ω and δ_a can be assumed to be time-varying instead.

The inertial measurements provide information about the position and orientation of the sensor. Integration of the angular velocity measured by the gyroscope leads to information about the sensor's change in orientation. Subtracting gravity from the specific force measured by the accelerometer and double integrating the resulting signal leads to information about the sensor's change in position. The process of estimating position and orientation from the inertial measurements is schematically illustrated in Figure 5.

5 Ultrawideband calibration

In this section, we will derive a calibration algorithm to determine the positions $\{r_m^n\}_{m=1}^M$ and the clock offsets $\{\Delta\tau_m\}_{m=1}^M$ of the receivers using the ML formulation (1). The algorithm makes use of data obtained by moving a single transmitter

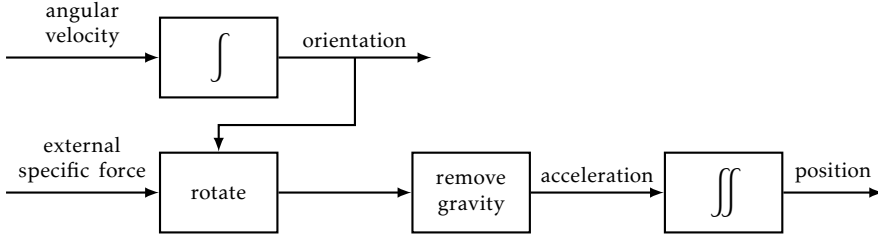


Figure 5: Schematic illustration of the process of determining position and orientation from inertial measurements, assuming a known initial position, velocity and orientation.

through the measurement volume. The data collected in this way is denoted $\mathcal{D}_1$. Since we do not place the transmitter at known positions, but instead move it around freely, the calibration algorithm aims at simultaneously localizing both the moving transmitter and the receivers. The transmission times of the different pulses $\{\tau_k^n\}_{k=1}^K$ are also considered unknown. Note that we have omitted the subscript l since we consider the case of using a single transmitter.

The UWB measurements are modeled according to (3), where the noise $e_{u,mk}$ is assumed to be distributed according to the asymmetric distribution (4). The parameters σ and γ are considered unknown and to be estimated, i.e. the algorithm tunes itself and does not rely on a priori knowledge about the accuracy of the UWB measurements. The resulting parameter vector is

$$\theta = \left(\{t_k^n\}_{k=1}^K, \{\tau_k^n\}_{k=1}^K, \{r_m^n, \Delta\tau_m\}_{m=1}^M, \sigma, \gamma \right). \quad (10)$$

To make use of the measurement model (4) within our calibration problem (1), we need the following relationship

$$p_{\theta}(y_{u,mk}) = p_{e_{u,mk}}(y_{u,mk} - \tau_k - \frac{1}{c} \|r_m^n - t_k^n\|_2 - \Delta\tau_m), \quad (11)$$

where θ is defined in (10).

The calibration problem is non-convex and hence needs proper initialization. In Sections 5.1 and 5.2, we introduce a two-step procedure to compute such an initial estimate. In a first step, we obtain an initial estimate of the receiver positions $\{r_{m,0}^n\}_{m=1}^M$ and their clock offsets $\{\Delta\tau_{m,0}\}_{m=1}^M$ using a second data set $\mathcal{D}_2$ for which the transmitter positions are known. In a second step, initial estimates of the transmitter positions $\{t_{k,0}^n\}_{k=1}^K$ and the transmission times $\{\tau_{k,0}\}_{k=1}^K$ are obtained by assuming that the receiver positions and clock offsets are known. This is done using a novel multilateration approach in which the UWB measurements are assumed to be distributed according to the asymmetric heavy-tailed distribution (4) with unknown σ and γ . In Section 5.3 we will then introduce the resulting calibration algorithm which is used to compute an ML estimate of all the unknown parameters θ defined in (10).

5.1 Initial estimate: step I

As a first step of the initial estimation, a second data set, denoted $\mathcal{D}_2$, is used to determine the positions and the clock offsets of the receivers. This data is collected by placing a number of UWB transmitters at known locations. To avoid manual measuring of the positions of the transmitters, they are rigidly attached to the receivers. Hence, the relative position of each transmitter with respect to the corresponding receiver is approximately known and constant. This relative distance is denoted d_{rt}^n . As discussed in Section 3, an arbitrary choice of the reference clock offset and the UWB coordinate frame needs to be used as a constraint to the calibration problem. Hence, the optimization problem can be written as

$$\hat{\theta}_1 = \arg \min_{\theta_1} - \sum_{m=1}^M \sum_{l=1}^L \sum_{k=1}^K \log p_{\theta_1}(y_{u,mlk}), \quad (12a)$$

$$\text{s.t. } A \begin{pmatrix} \text{vec}(r_{1:M}^n) \\ \Delta \tau_{1:M} \end{pmatrix} = b, \quad (12b)$$

$$r_m^n - t_m^n = d_{\text{rt}}^n, \quad m = 1, \dots, M, \quad (12c)$$

where the UWB measurements $y_{u,mlk}$ are modeled according to (3). Since it is not necessary to walk through the measurement volume during the collection of this data, the measurements in the data set $\mathcal{D}_2$ can typically be assumed to have little problems with outliers. Hence, $e_{u,mlk}$ in (3) can fairly accurately be modeled using a Gaussian PDF. Note that this is the only instance in this work where we assume that the UWB measurements are distributed according to a Gaussian. By assuming that the standard deviation of this Gaussian is the same for all m, l and k , we have reduced (12) to a constrained least-squares problem. The parameter vector θ_1 is given by

$$\theta_1 = \left(\{r_m^n\}_{m=1}^M, \{\tau_{lk}^n\}_{l=1}^L, \{\tau_{lk, \mathcal{D}_2}^n\}_{k=1}^K \}_{l=1}^L \right). \quad (13)$$

The subscript $\mathcal{D}_2$ on the (stationary) transmitter positions t_l^n and the transmission times τ_{lk}^n is added to stress that these parameters are only relevant for the dataset $\mathcal{D}_2$.

The constraints (12b) are defined as

$$A \triangleq \begin{pmatrix} \begin{array}{ccc|ccc|c} & & & 0_{1 \times 3M} & & & 1 \\ \hline & \mathcal{I}_3 & & 0_{3 \times (3M-3)} & & & 0_{1 \times (M-1)} \\ \hline 0_{2 \times 3} & e_2 & & 0_{2 \times (3M-6)} & & & \\ \hline 0_{1 \times 6} & e_3 & & 0_{1 \times (3M-9)} & & & \\ \hline & 0_{1 \times 9} & & \ddots & & 0_{1 \times (3M-12)} & \\ \hline \end{array} & 0_{(M+3) \times M} \end{pmatrix} \quad (14)$$

$$b \triangleq \begin{pmatrix} 1 & 0 & 0 & h_1 & 0 & h_2 & h_{3:M} \end{pmatrix}^\top$$

where $e_2 = (0_{2 \times 1} \quad \mathcal{I}_2)^\top$ and $e_3 = (0 \quad 0 \quad 1)^\top$. The first row in the matrix A is used to define the reference clock offset. Note that the choice of which receiver to use

for this is arbitrary. The remaining rows are used to define the UWB coordinate frame. The first receiver is used to define the origin. The second receiver is used to define the x-axis. The height of each receiver $m = 1, \dots, M$ is constrained to be equal to its measured height h_m . This ensures that the UWB coordinate frame is gravity-aligned which is beneficial for our sensor fusion approach in Section 6.

The constraint (12c) is used to incorporate the knowledge of the location of the transmitters with respect to the receivers. Note that we assume that each receiver has a transmitter attached to it, i.e. we have M constraints (12c).

The problem (12) is again a non-convex optimization problem and therefore requires a reasonably good starting point. Hence, we start the solver for (12) in a user-specified initial receiver configuration, a noisy, rotated and scaled estimate of the set of receiver positions.

5.2 Initial estimate: step II - multilateration

As a second step of the initialization, an initial estimate of the transmitter positions $\{t_{k,0}^n\}_{k=1}^K$ and the transmission times $\{\tau_{k,0}\}_{k=1}^K$ is determined for the calibration data set $\mathcal{D}_1$ in which a transmitter is moved around in the UWB measurement volume. This problem is solved using a novel multilateration approach, which can also be used stand-alone as will be done in Section 7.1. We model the UWB measurements using the asymmetric heavy-tailed distribution (4). To avoid ad hoc assumptions on σ and γ , we treat them as parameters in an ML problem where we estimate the parameters θ_2 with

$$\theta_2 = (\{t_k^n, \tau_k\}_{k=1}^K, \sigma, \gamma). \quad (15)$$

Using the fact that the logarithm is a monotonic function, the resulting optimization problem is given by

$$\hat{\theta}_2 = \arg \min_{\theta_2} - \sum_{k=1}^N \sum_{m=1}^M \log p_{\theta_2}(y_{u,mk}), \quad (16)$$

where the the UWB measurements are modeled as (3) and their noise $e_{u,mk}$ is given by the asymmetric noise distribution (4). Hence, instead of solving N individual multilateration problems, we solve one optimization problem to determine the transmitter positions, the transmission times as well as the parameters σ and γ .

The multilateration problem formulated in (16) can be solved¹ using standard Gauss-Newton solvers (Boyd and Vandenberghe, 2004; Nocedal and Wright, 2006) where the negative log-likelihood, its gradient and approximate Hessian are evaluated at the current iterate. Evaluating (3), the sign of $e_{u,mk}$ for each pulse $k = 1, \dots, N$ and each receiver $m = 1, \dots, M$ can be used to determine whether the Gaussian or Cauchy negative log-likelihood in (7) should be used.

¹As for any nonlinear optimization problem, good initial estimates help for convergence. Hence, we first estimate a part of the parameter vector (15), choosing $\sigma = \gamma = 1$. The resulting parameters are then used to determine a first estimate of σ and γ . Finally, the obtained parameter values are used as initial values for the final optimization.

5.3 Resulting calibration algorithm

The resulting calibration algorithm uses the data set $\mathcal{D}_1$ in which a transmitter is moved around in the measurement volume. To obtain an ML estimate of the parameter vector (10), the following constrained optimization problem is solved

$$\hat{\theta}^{\text{ML}} = \arg \min_{\theta} - \sum_{m=1}^M \sum_{k=1}^K \log p_{\theta}(y_{u,mk}), \quad (17a)$$

$$\text{s.t.} \quad A \begin{pmatrix} \text{vec}(r_{1:M}^n) \\ \Delta \tau_{1:M} \end{pmatrix} = b, \quad (17b)$$

where we make use of the UWB measurement model (3) and the asymmetric noise distribution (4). The constraints (17b) have already been defined in (14). The problem can be solved using standard constrained Gauss-Newton solvers (Boyd and Vandenberghe, 2004; Nocedal and Wright, 2006). The calibration algorithm is summarized in Algorithm 1.

Algorithm 1 Ultrawideband calibration

- 1: Construct a setup consisting of M stationary receivers.
- 2: Place M transmitters in close proximity to the receiver antennas and collect a data set $\mathcal{D}_2$.
- 3: Solve (12) using the data set $\mathcal{D}_2$ to obtain $\{r_{m,0}^n, \Delta \tau_{m,0}\}_{m=1}^M$. The optimization is initialized using a noisy, scaled and rotated estimate of the set of receiver positions provided by the user.
- 4: Collect a data set $\mathcal{D}_1$ by moving a single transmitter throughout the measurement volume.
- 5: Solve the multilateration problem (16) using the data set $\mathcal{D}_1$ with the calibration values of Step 3 to obtain $(\{t_{k,0}^n, \tau_{k,0}\}_{k=1}^K, \sigma_0, \gamma_0)$.
- 6: Solve (17) for $\mathcal{D}_1$. The optimization is started in

$$\theta_0 = (\{t_{k,0}^n, \tau_{k,0}\}_{k=1}^K, \{r_{m,0}^n, \Delta \tau_{m,0}\}_{m=1}^M, \sigma_0, \gamma_0),$$

using the results from Steps 3 and 5.

6 Sensor fusion

In this section we describe our approach to combine UWB measurements with inertial measurements to estimate the 6D pose of the sensor. It is based on tightly-coupled sensor fusion of the UWB and the inertial sensors. We formulate the sensor fusion problem as a MAP problem (2), estimating the state vector

$$x_{1:N} = \left\{ \{p_t^n, v_t^n, q_t^{\text{nb}}\}_{t=1}^N, \{\tau_k\}_{k=1}^K, \delta_a, \delta_{\omega}, \sigma_a, \sigma_{\omega}, \sigma, \gamma \right\}, \quad (18)$$

where p_t^n and v_t^n denote the sensor position and velocity at time t , respectively. Both of these are expressed in the navigation frame n . The sensor orientation is

denoted q_t^{nb} . Note that we encode the orientation state using a three-dimensional state vector around a linearization point represented by a unit quaternion (Kok, 2014; Crassidis et al., 2007; Grisetti et al., 2010; Hol, 2011). The UWB system typically obtains measurements at a lower frequency than the inertial measurement frequency. Hence, we use a subscript k to denote the UWB measurements while using a subscript t to denote the inertial measurements. The state $x_{1:N}$ is modeled for each time t .

In our approach, we make use of a dynamic model where the inertial measurements can be thought of as inputs. Hence, we model the position, the velocity and the orientation of the IMU in terms of the sensor acceleration and angular velocity as

$$\begin{aligned} p_{t+1}^{\text{n}} &= p_t^{\text{n}} + T v_t^{\text{n}} + \frac{T^2}{2} a_t^{\text{n}} \\ &= p_t^{\text{n}} + T v_t^{\text{n}} + \frac{T^2}{2} R_t^{\text{nb}} (y_{a,t} - \delta_a - e_{a,t}) + \frac{T^2}{2} g^{\text{n}}, \end{aligned} \quad (19a)$$

$$\begin{aligned} v_{t+1}^{\text{n}} &= v_t^{\text{n}} + T a_t^{\text{n}} \\ &= v_t^{\text{n}} + T R_t^{\text{nb}} (y_{a,t} - \delta_a - e_{a,t}) + T g^{\text{n}}, \end{aligned} \quad (19b)$$

$$q_{t+1}^{\text{nb}} = q_t^{\text{nb}} \odot \exp\left(\frac{T}{2} \omega_t\right), \quad (19c)$$

where T denotes the IMU sampling interval. The acceleration a_t^{n} is obtained from (9). The orientation q_t^{nb} is modeled in terms of the angular velocity ω_t obtained from (8). In (19c), $\odot$ denotes the quaternion product and $\exp$ denotes the vector exponential

$$\exp\left(\frac{T}{2} \omega_t\right) = \left(\cos \left\| \frac{T}{2} \omega_t \right\|_2 \quad \frac{\omega_t^{\text{T}}}{\|\omega_t\|_2} \sin \left\| \frac{T}{2} \omega_t \right\|_2 \right)^{\text{T}}. \quad (20)$$

For more details on quaternion algebra, see e.g. Hol (2011); Kuipers (1999). Note that we interchangeably make use of the unit quaternion q^{nb} and the rotation matrix R^{nb} as representations of the orientation. Furthermore, we use the notation $R^{\text{bn}} = (R^{\text{nb}})^{\text{T}}$ for the inverse rotation.

In the measurement model, the UWB measurements, modeled as in (3), are used to update the state. For this, the IMU and the UWB transmitter are assumed to be rigidly attached to each other. The position of the transmitter with respect to the IMU is assumed to be known.

The state is computed using the following MAP problem

$$\hat{x}_{1:N}^{\text{MAP}} = \arg \min_{x_{1:N}} - \sum_{t=1}^N \log p(x_{t+1} | x_t, y_{a,t}, y_{\omega,t}) - \sum_{k=1}^K \sum_{m=1}^M \log p(y_{u,mk} | x_{t(k)}), \quad (21)$$

where the first term denotes the dynamic model described by (19). The second term denotes the measurement model, using the UWB measurement model (3), the asymmetric heavy-tailed distribution (4) and the relative position of the trans-

mitter with respect to the IMU. The problem (21) can be solved² using a standard Gauss-Newton algorithm, see e.g. Boyd and Vandenberghe (2004); Nocedal and Wright (2006). Since the sensor fusion problem is nonlinear, parts of the problem are solved first to provide good initial estimates. The resulting pose estimation approach is summarized in Algorithm 2.

Algorithm 2 Pose estimation

- 1: Collect inertial data and UWB data.
 - 2: Initialize the quaternions as identity and all other parts of $x_{1:N}$ in (18) as zero.
 - 3: Solve the state estimation problem (21) to obtain $\hat{x}_{1:N}^{\text{MAP}}$.
-

7 Experimental results

In our experiments, we use an UWB setup consisting of 10 receivers deployed in a room with a size of approximately $8 \times 6 \times 2.5$ m. A test-subject walks around the measurement volume along a circular path for approximately 24 s. An IMU and an UWB transmitter have been attached to his foot. The UWB measurements are significantly affected by the NLOS conditions and multipath since the transmitter is quite close to the ground and since the body frequently blocks the direct path to the receivers. Optical markers have also been placed on the body to provide reference data. These optical markers are tracked by a camera system (Vicon, 2014). This industry-standard system has an accuracy that is an order of magnitude larger than the accuracy expected from our UWB system. Hence (using multiple markers), the camera system can provide both ground truth position and orientation estimates.

In Section 7.1 we will use the UWB data from the transmitter on the subject's foot to experimentally validate the proposed asymmetric distribution (4). We will show that using this distribution in the multilateration approach described in Section 5.2, considerably better position estimates are obtained than when the same approach would be used assuming Gaussian or Cauchy distributed noise. In Section 7.2 we will present our calibration results. In Section 7.3 we will present our sensor fusion results.

7.1 Experimental validation of the asymmetric noise distribution

In this section we will use the multilateration approach introduced in Section 5.2 while assuming that the system has previously been calibrated. We will use three different assumptions on the noise used in (16), namely

1. the asymmetric noise distribution (4) with parameters $\theta_A = (\{t_k^n, \tau_k\}_{k=1}^K, \sigma, \gamma)$,

²Also for the sensor fusion problem, good initial estimates help for convergence. Hence, we first estimate only a part of the state vector (18), choosing the parameters related to the noise characteristics as a fixed value. We also use a Gaussian prior for the sensor biases. The resulting parameters are then used to determine a first estimate of σ_a , σ_ω , σ and γ . Finally, the obtained estimates are used as initial values for the final optimization (21).

Table 1: RMSE for the position using the multilateration approach introduced in Section 5.2 as compared to data from an optical reference system assuming the noise to be distributed according to the asymmetric distribution (4), according to a Cauchy distribution and according to a Gaussian distribution.

	x [cm]	y [cm]	z [cm]
Asymmetric distribution (4)	5.9	7.2	12.2
Cauchy distribution	22.0*	9.2*	19.1*
Gaussian distribution	53.4*	75.9*	176.9*

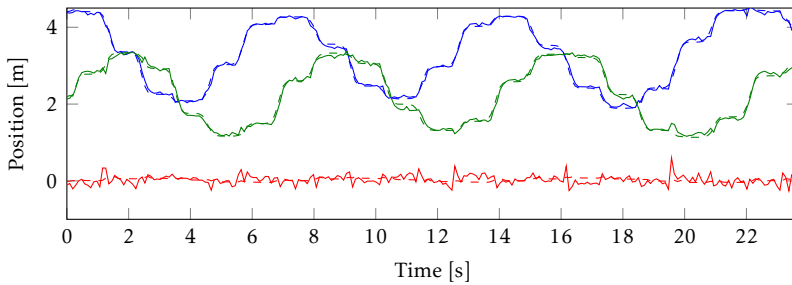
* One measurement has been discarded in computing these RMSE values. These measurements arise around 11 s for the Cauchy distribution and around 22 s for the Gaussian distribution and deviate from the reference positions by more than 100 m.

2. a Cauchy distribution with parameters $\theta_C = (\{t_k^n, \tau_k\}_{k=1}^K, \gamma)$,
3. a Gaussian distribution with parameters $\theta_G = (\{t_k^n, \tau_k\}_{k=1}^K, \sigma)$.

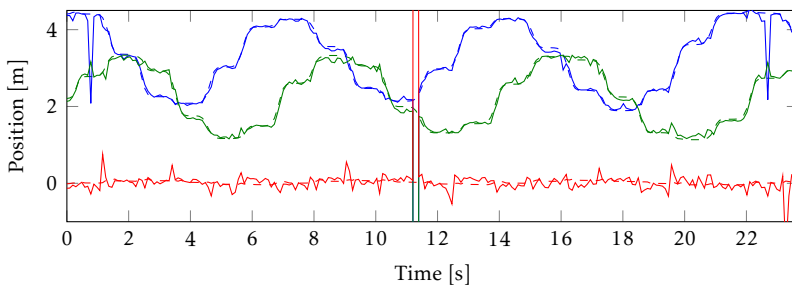
The resulting position estimates are shown in Figure 6. The dashed lines are the position estimates from the optical reference system. As can be seen, the position estimates are best for the asymmetric distribution and worst for the Gaussian distribution. This is also summarized in Table 1 in terms of the root mean square error (RMSE) for the position. In Figure 7 the residuals including their estimated PDFs are plotted for all three cases. As can be seen, the Gaussian is clearly not a good fit due to the large number of outliers. Although the estimated Cauchy distribution seems to describe the residuals reasonably well, this model allows for physically unreasonable negative residuals, i.e. pulses traveling faster than the speed of light. Hence, also from these histograms it can be concluded that the asymmetric noise distribution (4) offers the best model for the experimental data. Note that these results are highly dependent on the number of outliers in the UWB data. On “cleaner” UWB data, the difference in position accuracy between the different distributions would of course be less.

7.2 Calibration

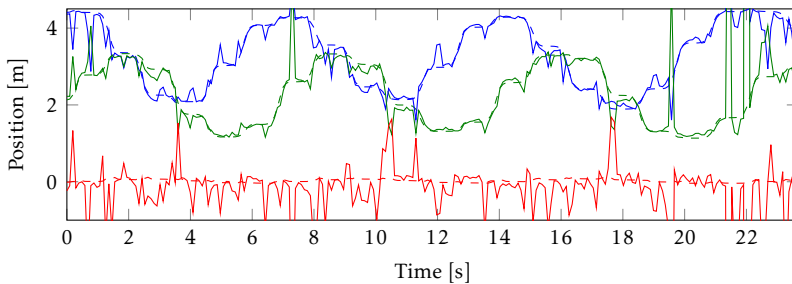
Algorithm 1 has been used to compute an estimate of the positions and clock offsets of the receivers in the UWB setup. The estimated trajectory of the transmitter and the ML estimates of the receiver positions are depicted in Figure 8. The smoothness of the estimated transmitter trajectory suggests that good multilateration results are obtained and hence gives confidence also in the resulting calibration results. Figure 9 shows a histogram of the residuals from the calibration algorithm. As can be seen, the estimated PDF (shown in red) fits the data reasonably well. Furthermore, the calibration has been used in the sensor fusion



(a) Asymmetric



(b) Cauchy



(c) Gaussian

Figure 6: Position estimates from the multilateration approach (16) using the measurements from the transmitter on the foot. The estimated positions using the asymmetric heavy-tailed noise distribution (4) are depicted in (a). The different colors denote the different directions with x in blue, y in green and z in red. The estimated positions using a Cauchy and a Gaussian noise distribution are depicted in (b) and (c), respectively. Data from an optical reference system is included as dashed lines in each plot for comparison.

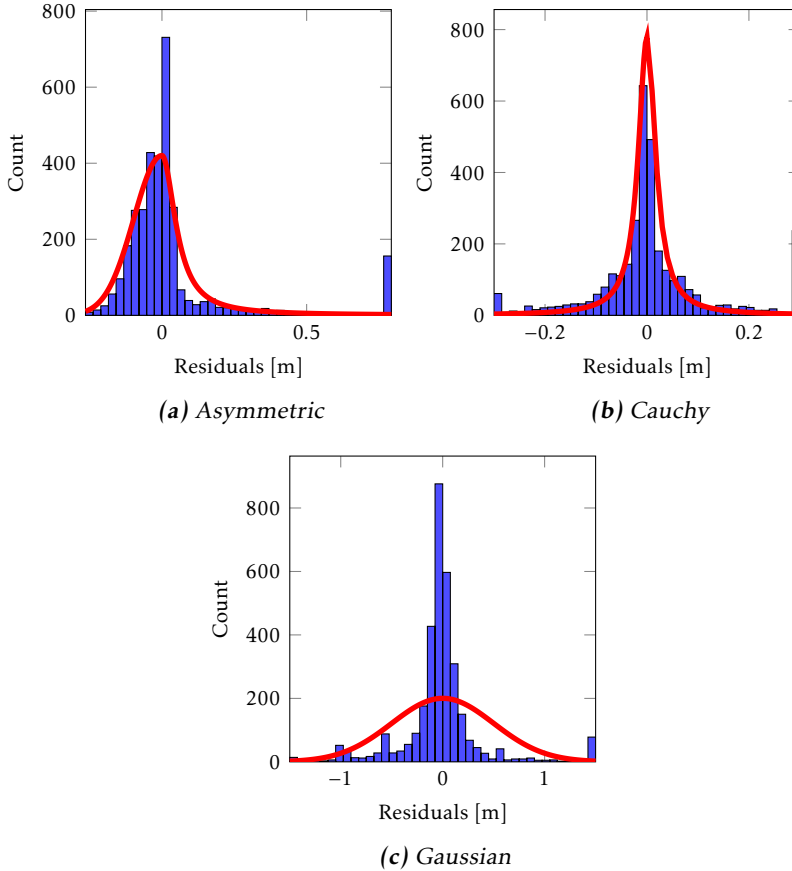


Figure 7: Residuals from the multilateration approach including their estimated PDFs using (a) the asymmetric distribution (4), a symmetric Cauchy distribution (b) and a symmetric Gaussian distribution (c). The residuals outside of the scope of the figures have been collected in the outermost bins. Note the different scales on the x -axes and the fact that the left plot is not centered around 0 to emphasize the asymmetric nature of this distribution. The quantities in the measurement equation (3) are all expressed in meters resulting in residuals in meters.

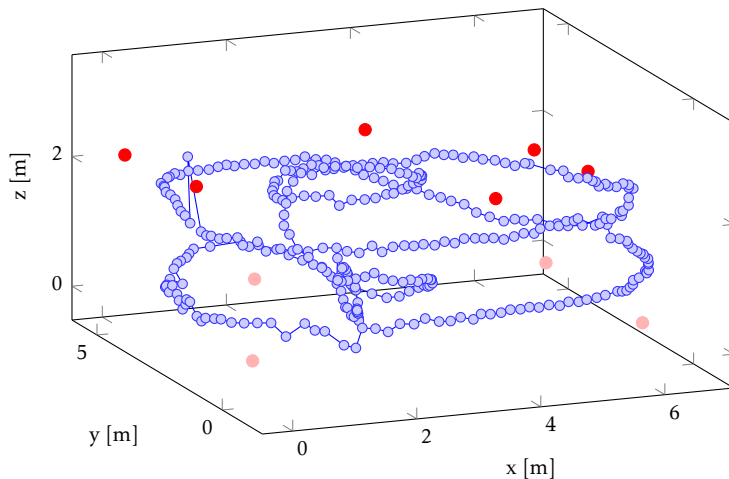


Figure 8: Calibration results of the UWB setup. The estimated transmitter positions are depicted in blue. The receivers in our UWB setup are either placed close to the ground or close to the ceiling. The positions of the receivers close to the ceiling are depicted in bright red and the positions of the receivers close to the ground are depicted in light red.

algorithm with good results (as will be discussed in the subsequent section), which indirectly validates the quality of the calibration.

The results described in this section have been obtained using an inefficient implementation of Algorithm 1 in Matlab. However, efficient implementation should be possible due to the sparsity inherent in the problem and the typical problem dimensions of less than 2 500 parameters (this corresponds to collecting calibration data $\mathcal{D}_1$ for one minute at 10 Hz). The sparsity pattern of the matrix that needs inversion in the constrained Gauss-Newton algorithm, consisting of the approximate Hessian and the gradients of the constraints (see e.g. Boyd and Vandenberghe (2004)), is shown in Figure 10. Note that since our calibration problem is nonlinear, this matrix inversion needs to be performed several times.

7.3 Pose estimation

To evaluate the proposed pose estimation solution (Algorithm 2), it has been used to track the motion of an IMU and an UWB transmitter placed on the foot of a test-subject walking in an indoor environment, using the experiment already introduced in Section 7.1. The IMU provides 120 Hz inertial measurements. The UWB pulses are transmitted at 10 Hz.

Figure 11 shows an overview of the position estimated using Algorithm 2. The positions of the UWB receivers are shown in red. The circular path is clearly recognizable. It only occupies a small part of the measurement volume of the UWB tracking system so that a performance comparison with an optical reference system is possible.

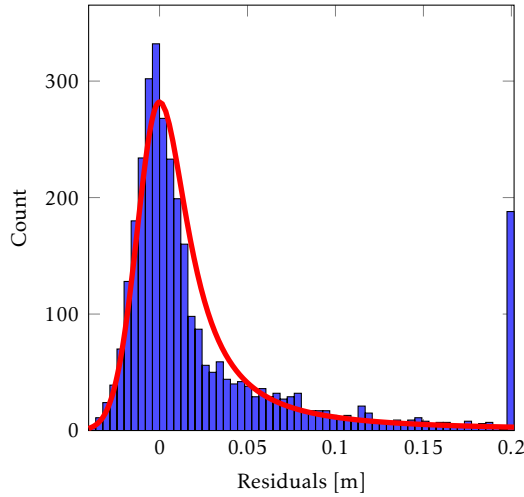


Figure 9: Residuals from the calibration algorithm and the estimated PDF. Note that the residuals outside of the scope of the figure have been collected in the outermost bins and that the plot is not centered around 0 to emphasize the asymmetric nature of the distribution. The quantities in the measurement equation (3) are all expressed in meters resulting in residuals in meters.

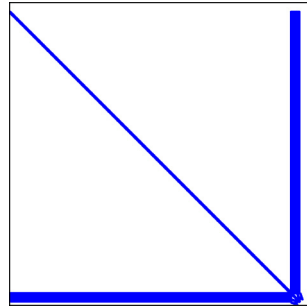


Figure 10: The sparsity pattern of the matrix that needs inversion in the constrained Gauss-Newton algorithm, consisting of the approximate Hessian and the gradients of the constraints. The block-diagonal part is due to the independency of the different UWB pulses. The arrow-point is due to the model dependency of each pulse on the receiver positions and their clock offsets (see (3)). Only 5.7% of the matrix elements is non-zero.

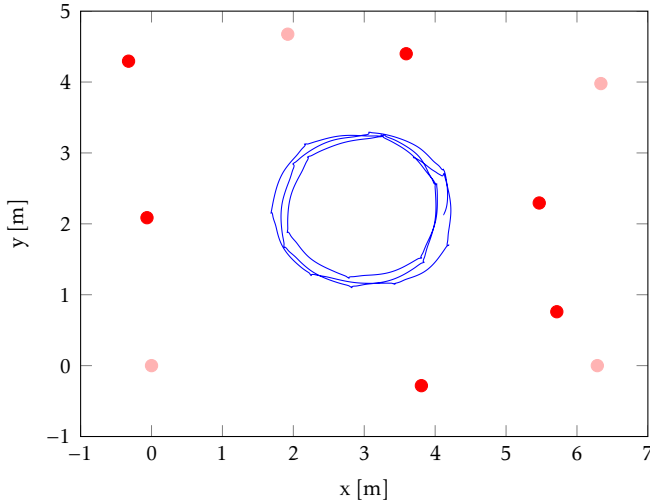


Figure 11: Top view of the experiment where the subject walked along a circular path. The estimated trajectory $p_{1:N}^n$ of the IMU on the subject's foot is shown in blue. The positions of the receivers close to the ceiling are depicted in bright red. The positions of the receivers close to the floor are depicted in light red.

Table 2: RMSE for the position and orientation estimates from Algorithm 2 as compared to data from the optical reference system.

	x	y	z
position [cm]	3.0	3.0	2.3
orientation [°]	0.37	0.44	0.69

Figures 12 and 13 show the estimated position and orientation as compared to those from the optical reference system. It can be concluded that our solution is capable of producing a drift-free and accurate pose estimate at a high output frequency. In fact, the comparison shows 3 cm RMSE for position and less than 1° RMSE for orientation, see Table 2.

As for the UWB calibration algorithm, our implementation has not been optimized in terms of computational speed. However, the sensor fusion problem is inherently sparse due to the Markov property of the state and can hence be solved efficiently.

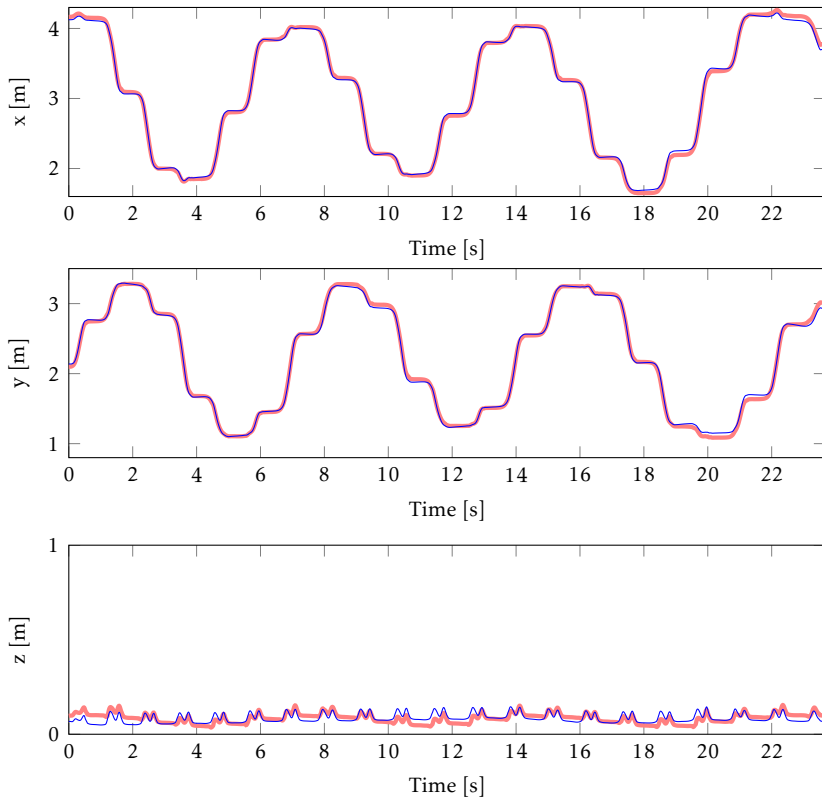


Figure 12: Position of the IMU $p_{1:N}^n$ on the subject's foot. The estimates from Algorithm 2 are depicted in blue. The estimates from the optical reference system are depicted in thick red.

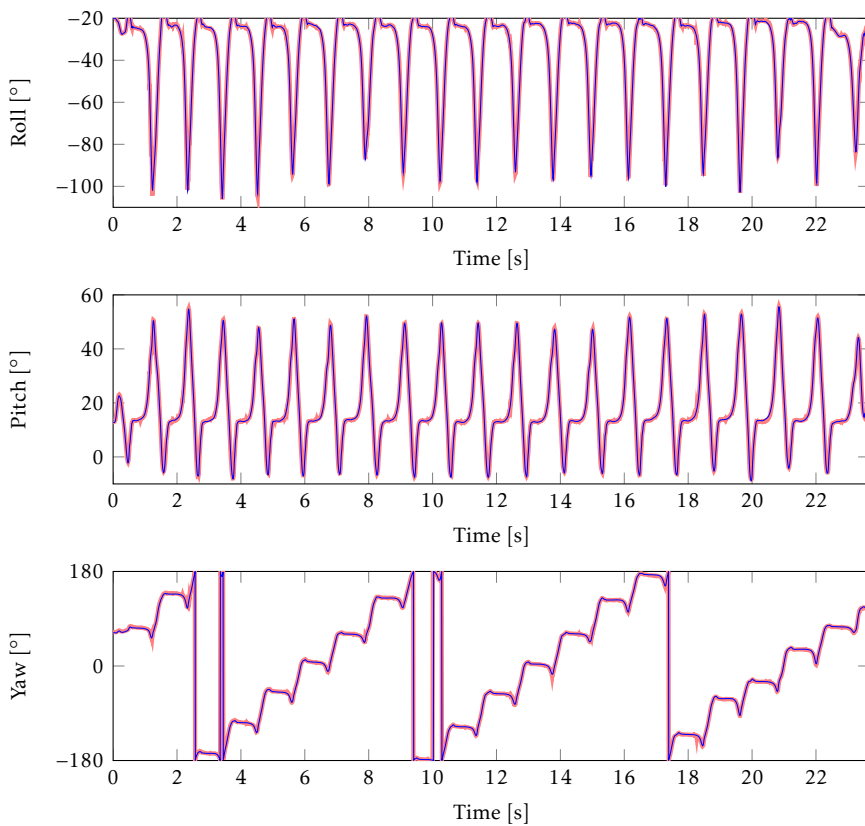


Figure 13: Orientation of the IMU $q_{1:N}^{nb}$ on the subject's foot expressed using Euler angles (roll, pitch, yaw). The estimates from Algorithm 2 are depicted in blue. The estimates from the optical reference system are depicted in thick red.

8 Conclusions and future work

We have presented a sensor fusion approach to combine inertial measurements with TOA measurements from an UWB system for 6D pose estimation. The approach is experimentally shown to result in accurate position and orientation estimates when compared to data from an independent optical reference system. To be able to use the UWB measurements in the sensor fusion approach, the UWB setup has to be calibrated, i.e. the receiver positions and their clocks offsets have to be computed. We have solved the UWB calibration problem using a novel approach, taking into account the possibility of delayed UWB measurements due to NLOS and/or multipath. Throughout this work, we have used an asymmetric heavy-tailed distribution to model the outliers in the UWB measurements. This model is shown to lead to accurate position estimates even from challenging data containing a fairly large amount of outliers in a new multilateration approach.

An interesting direction for future work is to combine the sensor fusion algorithm introduced in this work with the motion capture approach in Kok et al. (2014). This would open up for the possibility of combining information from multiple IMUs and multiple UWB transmitters to determine the pose of multiple body segments or even the entire human body or any other objects with multiple connected parts.

Acknowledgements

This work is supported by CADICS, a Linnaeus Center funded by the Swedish Research Council (VR), by BALANCE, a European Commission, FP7 research project and by the project *Probabilistic modeling of dynamical systems* (Contract number: 621-2013-5524) funded by the Swedish Research Council (VR).

Bibliography

- N. A. Alsindi, B. Alavi, and K. Pahlavan. Measurement and modeling of ultrawide-band TOA-based ranging in indoor multipath environments. *IEEE Transactions on Vehicular Technology*, 58(3):1046–1058, 2009.
- G. Bellusci, D. Roetenberg, F. Dijkstra, H. J. Luinge, and P. Slycke. Xsens MVN MotionGrid: Drift-free human motion tracking using tightly coupled ultrawideband and miniature inertial sensors, June 2011.
- S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- Y. T. Chan and K. C. Ho. A simple and efficient estimator for hyperbolic location. *IEEE Transactions on Signal Processing*, 42(8):1905–1915, 1994.
- J. A. Corrales, F. A. Candelas, and F. Torres. Hybrid tracking of human operators using IMU/UWB data fusion by a Kalman filter. In *Proceedings of the 3rd ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 193–200, 2008.
- J. L. Crassidis, F. L. Markley, and Y. Cheng. A survey of nonlinear attitude estimation methods. *Journal of Guidance, Control, and Dynamics*, 30(1):12–28, 2007.
- A. De Angelis, J. O. Nilsson, I. Skog, P. Händel, and P. Carbone. Indoor positioning by ultrawide band radio aided inertial navigation. *Metrology and Measurement Systems*, 17(3):447–460, 2010.
- S. Gezici, Z. Tian, G. B. Giannakis, H. Kobayashi, A. F. Molisch, H. V. 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, 2005.
- G. Grisetti, R. Kümmerle, C. Stachniss, U. Frese, and C. Hertzberg. Hierarchical optimization on manifolds for online 2D and 3D mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 273–278, Anchorage, Alaska, 2010.
- F. Gustafsson. *Statistical Sensor Fusion*. Studentlitteratur, 2012.
- F. Gustafsson and F. Gunnarsson. Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements. *IEEE Signal Processing Magazine*, 22(4):41–53, 2005.
- U. Hammes, E. Wolsztynski, and A. M. Zoubir. Robust tracking and geolocation for wireless networks in NLOS environments. *IEEE Journal of Selected Topics in Signal Processing*, 3(5):889–901, 2009.

- J. D. Hol. *Sensor fusion and calibration of inertial sensors, vision, ultra-wideband and GPS*. Dissertation no. 1368, Linköping University, Linköping, Sweden, June 2011.
- J. D. Hol, F. Dijkstra, H. Luinge, and T. B. Schön. Tightly coupled UWB/IMU pose estimation. In *Proceedings of the IEEE International Conference on Ultra-Wideband (ICUWB)*, pages 688–692, Vancouver, Canada, September 2009.
- J. D. Hol, T. B. Schön, and F. Gustafsson. Ultra-wideband calibration for indoor positioning. In *Proceedings of the IEEE International Conference on Ultra-Wideband (ICUWB)*, pages 1–4, Nanjing, China, September 2010.
- M. Kok. *Probabilistic modeling for positioning applications using inertial sensors*. Licentiate’s thesis no. 1656, Linköping University, Linköping, Sweden, June 2014.
- M. Kok, J. D. Hol, and T. B. Schön. An optimization-based approach to human body motion capture using inertial sensors. In *Proceedings of the 19th World Congress of the International Federation of Automatic Control*, pages 79–85, Cape Town, South Africa, August 2014.
- M. Kok, J. D. Hol, and T. B. Schön. Indoor positioning using ultrawideband and inertial measurements. *IEEE Transactions on Vehicular Technology*, 64(4):1293–1303, 2015.
- J. B. Kuipers. *Quaternions and Rotation Sequences: a primer with applications to orbits, aerospace, and virtual reality*. Princeton University Press, 1999.
- J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research, 2nd edition, 2006.
- N. Patwari, J. N. Ash, S. Kyperountas, A. O. Hero, R. L. Moses, and N. S. Correal. Locating the nodes: cooperative localization in wireless sensor networks. *IEEE Signal Processing Magazine*, 22(4):54–69, 2005.
- S. Pittet, V. Renaudin, B. Merminod, and M. Kasser. UWB and MEMS based indoor navigation. *Journal of Navigation*, 61(3):369–384, July 2008.
- A. Prorok and A. Martinoli. Accurate indoor localization with ultra-wideband using spatial models and collaboration. *The International Journal of Robotics Research*, 33(4):547–568, 2014.
- Z. Sahinoglu, S. Gezici, and I. Güvenc. *Ultra-wideband Positioning Systems: Theoretical Limits, Ranging Algorithms, and Protocols*. Cambridge University Press, 2008.
- A. H. 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, 2005.

- S. Sczyslo, J. Schroeder, S. Galler, and T. Kaiser. Hybrid localization using UWB and inertial sensors. In *Proceedings of the IEEE International Conference on Ultra-Wideband (ICUWB)*, volume 3, pages 89–92, Hannover, Germany, September 2008.
- Time Domain. <http://www.timedomain.com>, 2014. Accessed on April 7.
- Ubisense. <http://www.ubisense.net>, 2014. Accessed on April 8.
- Vicon. <http://www.vicon.com>, 2014. Accessed on October 6.
- Xsens Technologies B.V. <http://www.xsens.com>, 2014. Accessed on April 8.
- A. M. Zoubir, V. Koivunen, Y. Chakhchoukh, and M. Muma. Robust estimation in signal processing: A tutorial-style treatment of fundamental concepts. *IEEE Signal Processing Magazine*, 29(4):61–80, 2012.
- L. Zwirello, X. Li, T. Zwick, C. Ascher, S. Werling, and G. F. Trommer. Sensor data fusion in UWB-supported inertial navigation systems for indoor navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 3154–3159, Karlsruhe, Germany, 2013.

Paper E

Magnetometer calibration using inertial sensors

Authors: Manon Kok and Thomas B. Schön

Edited version of the paper:

M. Kok and T. B. Schön. Magnetometer calibration using inertial sensors. *IEEE Sensors Journal*, 16(14):5679 – 5689, 2016.

Earlier versions of this work were presented in:

M. Kok and T. B. Schön. Maximum likelihood calibration of a magnetometer using inertial sensors. In *Proceedings of the 19th World Congress of the International Federation of Automatic Control*, pages 92–97, Cape Town, South Africa, August 2014,

M. Kok, J. D. Hol, T. B. Schön, F. Gustafsson, and H. Luinge. Calibration of a magnetometer in combination with inertial sensors. In *Proceedings of the 15th International Conference on Information Fusion*, pages 787–793, Singapore, July 2012.

Magnetometer calibration using inertial sensors

Manon Kok^{*} and Thomas B. Schön[†]

^{*}Dept. of Electrical Engineering,
Linköping University,
SE-581 83 Linköping, Sweden
manko@isy.liu.se

[†]Dept. of Information Technology
Uppsala University,
SE-751 05 Uppsala, Sweden
thomas.schon@it.uu.se

Abstract

In this work we present a practical algorithm for calibrating a magnetometer for the presence of magnetic disturbances and for magnetometer sensor errors. To allow for combining the magnetometer measurements with inertial measurements for orientation estimation, the algorithm also corrects for misalignment between the magnetometer and the inertial sensor axes. The calibration algorithm is formulated as the solution to a maximum likelihood problem and the computations are performed offline. The algorithm is shown to give good results using data from two different commercially available sensor units. Using the calibrated magnetometer measurements in combination with the inertial sensors to determine the sensor's orientation is shown to lead to significantly improved heading estimates.

1 Introduction

Nowadays, magnetometers and inertial sensors (gyroscopes and accelerometers) are widely available, for instance in dedicated sensor units and in smartphones. Magnetometers measure the local magnetic field. When no magnetic disturbances are present, the magnetometer measures a constant local magnetic field vector. This vector points to the local magnetic north and can hence be used for heading estimation. Gyroscopes measure the angular velocity of the sensor. Integration of the gyroscope measurements gives information about the change in orientation. However, it does not provide absolute orientation estimates. Furthermore, the orientation estimates suffer from integration drift. Accelerometers measure the sensor's acceleration in combination with the earth's gravity. In the case of small or zero acceleration, the measurements are dominated by the gravity component. Hence, they can be used to estimate the inclination of the sensor.

Inertial sensors and magnetometers have successfully been used to obtain accurate 3D orientation estimates for a wide range of applications. For this, however, it is imperative that the sensors are properly calibrated and that the sensor axes

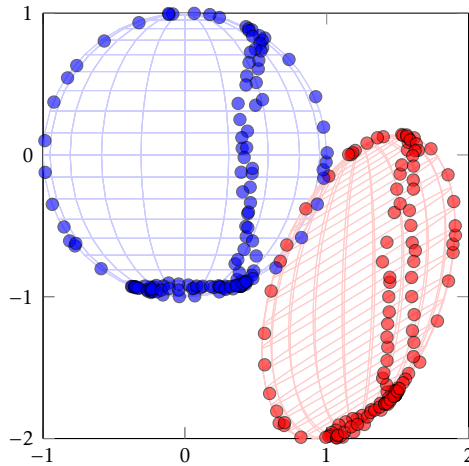


Figure 1: Example calibration results with an ellipsoid of magnetometer data before calibration (red) and a unit sphere of data after calibration (blue).

are aligned. Calibration is specifically of concern for the magnetometer, which needs recalibration whenever it is placed in a (magnetically) different environment. When the magnetic disturbance is a result of the mounting of the magnetometer onto a magnetic object, the magnetometer can be calibrated to compensate for the presence of this disturbance. This is the focus of this work.

Our main contribution is a practical magnetometer calibration algorithm that is designed to improve orientation estimates when combining calibrated magnetometer data with inertial data. The word *practical* refers to the fact that the calibration does not require specialized additional equipment and can therefore be performed by any user. More specifically, this means that the orientation of the sensor is not assumed to be known. Instead, the calibration problem is formulated as an orientation estimation problem in the presence of unknown parameters and is posed as a maximum likelihood (ML) problem. The algorithm calibrates the magnetometer for the presence of magnetic disturbances, for magnetometer sensor errors and for misalignment between the magnetometer and the inertial sensor axes. Using the calibrated magnetometer measurements to estimate the sensor's orientation is experimentally shown to lead to significantly improved heading estimates. We aggregate and extend the work from Kok and Schön (2014) and Kok et al. (2012) with improvements on the implementation of the algorithm. Furthermore, we include a more complete description and analysis, more experimental results and a simulation study illustrating the heading accuracy that can be obtained with a properly calibrated sensor.

To perform the calibration, the sensor needs to be rotated in all possible orientations. A perfectly calibrated magnetometer would in that case measure rotated versions of the local magnetic field vector. Hence, the magnetometer data would lie on a sphere. In practice, however, the magnetometer will often measure an ellipsoid of data instead. The calibration maps the ellipsoid of data to a sphere

as illustrated in Figure 1. The alignment of the inertial and magnetometer sensor axes determines the orientation of the sphere. Since we are interested in improving the heading estimates, the actual magnitude of the local magnetic field is of no concern. Hence, we assume without loss of generality that the norm is equal to 1, i.e. the sphere in Figure 1 is a unit sphere.

2 Related work

Traditional magnetometer calibration approaches assume that a reference sensor is available which is able to provide accurate heading information. A well-known example of this is compass swinging (Bowditch, 2002). To allow for any user to perform the calibration, however, a large number of approaches have been developed that remove the need for a source of orientation information. One class of these magnetometer calibration algorithms focuses on minimizing the difference between the magnitude of the measured magnetic field and that of the local magnetic field, see e.g. Alonso and Shuster (2002). This approach is also referred to as scalar checking (Lerner, 1978). Another class formulates the calibration problem as an ellipsoid fitting problem, i.e. as the problem of mapping an ellipsoid of data to a sphere, see e.g. Vasconcelos et al. (2011); Renaudin et al. (2010); Gebre-Egziabher et al. (2006). The benefit of using this formulation, is that there is a vast literature on solving ellipsoid fitting problems, see e.g. Gander et al. (1994); Markovsky et al. (2004). Outside of these two classes, a large number of other calibration approaches is also available, for instance Wu and Shi (2015), where different formulations of the calibration problem in terms of an ML problem are considered.

The benefit of the approaches discussed above is that they can be used with data from a magnetometer only. Our interest, however, lies in calibrating a magnetometer for improved heading estimation in combination with inertial sensors. Alignment of the sensor axes of the inertial sensors and the magnetometer is in this case crucial. This alignment can be seen as determining the orientation of the blue sphere of calibrated magnetometer data in Figure 1. Algorithms that only use magnetometer data can map the red ellipsoid of data to a sphere, but without additional information, the rotation of this sphere remains unknown.

A number of recent approaches include a second step in the calibration algorithm to determine the misalignment between different sensor axes (Vasconcelos et al., 2011; Li and Li, 2012; Salehi et al., 2012; Bonnet et al., 2009). A common choice to align the magnetometer and inertial sensor axes, is to use accelerometer measurements from periods of fairly small accelerations (Li and Li, 2012; Salehi et al., 2012). The downside of this approach is that a threshold for using accelerometer measurements needs to be determined. Furthermore, data from the gyroscope is hereby omitted. In Troni and Whitcomb (2013) on the other hand, the problem is reformulated in terms of the change in orientation, allowing for direct use of the gyroscope data.

In our algorithm we instead formulate the magnetometer calibration problem as a problem of estimating the sensor's orientation in the presence of unknown

(calibration) parameters. This formulation naturally follows from the fact that the problem of orientation estimation and that of magnetometer calibration are inherently connected: If the magnetometer is properly calibrated, good orientation estimates can be obtained. Reversely, if the orientation of the sensor is known accurately, the rotation of the sphere in Figure 1 can accurately be determined, resulting in a good magnetometer calibration. In this formulation, data from the accelerometer *and* the gyroscope is used to aid the magnetometer calibration.

Our formulation of the calibration problem requires solving a non-convex optimization problem to obtain ML estimates of the calibration parameters. To obtain good initial values of the parameters, an ellipsoid fitting problem and a misalignment estimation problem are solved. Solving the calibration problem as a two-step procedure is similar to the approaches in Li and Li (2012); Salehi et al. (2012). We analyze the quality of the initial estimates and of the ML estimates in terms of their heading accuracy, both for experimental and simulated data. Based on this analysis, we show that significant heading accuracy improvements can be obtained by using the ML estimates of the parameters.

3 Problem formulation

Our magnetometer calibration algorithm is formulated as a problem of determining the sensor's orientation in the presence of unknown model parameters θ . It can hence be considered to be a grey-box system identification problem. A nonlinear state space model on the following form is used

$$x_{t+1} = f_t(x_t, \omega_t, e_{\omega,t}, \theta), \quad (1a)$$

$$y_t = \begin{pmatrix} y_{a,t} \\ y_{m,t} \end{pmatrix} = \begin{pmatrix} h_{a,t}(x_t) \\ h_{m,t}(x_t, \theta) \end{pmatrix} + e_t(\theta), \quad (1b)$$

where the state x_t represents the sensor's orientation at time t . We use the change in orientation, i.e. the angular velocity ω_t , as an input to the dynamic model $f_t(\cdot)$. The angular velocity is measured by the gyroscope. However, the measurements $y_{\omega,t}$ are corrupted by a constant bias δ_ω and Gaussian i.i.d. measurement noise with zero mean and covariance Σ_ω , i.e. $e_{\omega,t} \sim \mathcal{N}(0_{3 \times 1}, \Sigma_\omega)$.

The measurement models $h_{a,t}(\cdot)$ and $h_{m,t}(\cdot)$ in (1b) describe the accelerometer measurements $y_{a,t}$ and the magnetometer measurements $y_{m,t}$, respectively. The accelerometer measurement model assumes that the acceleration of the sensor is small compared to the earth gravity. Since the magnetometer is not assumed to be properly calibrated, the magnetometer measurement model $h_{m,t}(\cdot)$ depends on the parameter vector θ . The exact details of the magnetometer measurement model will be introduced in Section 4. The accelerometer and magnetometer measurements are corrupted by Gaussian i.i.d. measurement noise

$$e_t = \begin{pmatrix} e_{a,t} \\ e_{m,t} \end{pmatrix} \sim \mathcal{N}\left(0_{6 \times 1}, \begin{pmatrix} \Sigma_a & 0_{3 \times 3} \\ 0_{3 \times 3} & \Sigma_m \end{pmatrix}\right). \quad (2)$$

The calibration problem is formulated as an ML problem. Hence, the parameters θ in (1) are found by maximizing the likelihood function $p_\theta(y_{1:N})$,

$$\hat{\theta}^{\text{ML}} = \arg \max_{\theta \in \Theta} p_\theta(y_{1:N}), \quad (3)$$

where $y_{1:N} = \{y_1, \dots, y_N\}$ and $\Theta \subseteq \mathbb{R}^{n_\theta}$. Using conditional probabilities and the fact that the logarithm is a monotonic function we have the following equivalent formulation of (3),

$$\hat{\theta}^{\text{ML}} = \arg \min_{\theta \in \Theta} - \sum_{t=1}^N \log p_\theta(y_t | y_{1:t-1}), \quad (4)$$

where we use the convention that $y_{1:0} \triangleq \emptyset$. The ML estimator (4) enjoys well-understood theoretical properties including strong consistency, asymptotic normality, and asymptotic efficiency (Ljung, 1999).

The state space model (1) is nonlinear, implying that there is no closed form solution available for the one step ahead predictor $p_\theta(y_t | y_{1:t-1})$ in (4). This can systematically be handled using sequential Monte Carlo methods (e.g. particle filters and particle smoothers), see e.g. Schön et al. (2011); Lindsten and Schön (2013). However, for the magnetometer calibration problem it is sufficient to make use of a more pragmatic approach; we simply approximate the one step ahead predictor using an extended Kalman filter (EKF). The result is

$$p_\theta(y_t | y_{1:t-1}) \approx \mathcal{N}(y_t; \hat{y}_{t|t-1}(\theta), S_t(\theta)), \quad (5)$$

where the mean value $\hat{y}_{t|t-1}(\theta)$ and the covariance $S_t(\theta)$ are obtained from the EKF (Gustafsson, 2012). Inserting (5) into (4) and neglecting all constants not depending on θ results in the following optimization problem,

$$\min_{\theta \in \Theta} \frac{1}{2} \sum_{t=1}^N \|y_t - \hat{y}_{t|t-1}(\theta)\|_{S_t^{-1}(\theta)}^2 + \log \det S_t(\theta), \quad (6)$$

which we can solve for the unknown parameters θ . The problem (6) is non-convex, implying that a good initial value for θ is required.

4 Magnetometer measurement model

In the case of perfect calibration, a magnetometer measures the local magnetic field and its measurements will therefore lie on a sphere with a radius equal to the local magnetic field. Since we are interested in using the magnetometer measurements to improve the orientation estimates from the state space model (1), the actual magnitude of the local magnetic field is of no concern. Hence, we assume without loss of generality that its norm is equal to one. We denote the normalized local magnetic field by m^n . Ideally, the magnetometer measurements then lie on a sphere with radius equal to one as

$$h_{m,t} = m_t^b = R_t^{\text{bn}} m^n, \quad (7)$$

where $h_{m,t}$ is defined in (1b). The explicit dependence on x_t and θ has been omitted for notational simplicity. The matrix R_t^{bn} is the rotation matrix representation of the orientation at time t . The superscript bn denotes that the rotation is from the *navigation frame* n to the *body frame* b . The body frame b is aligned with the sensor axes. The navigation frame n is aligned with the earth's gravity and the local magnetic field. In case the coordinate frame in which a vector is defined can be ambiguous, we explicitly indicate in which coordinate frame the vector is expressed by adding a superscript b or n . Hence, m^n denotes the normalized local magnetic field in the navigation frame n while m_t^b denotes the normalized local magnetic field in the body frame b . The latter is time-dependent and therefore also has a subscript t . Note that the rotation from navigation frame to body frame is denoted R_t^{nb} and $R_t^{nb} = (R_t^{bn})^T$.

In outdoor environments, the local magnetic field is equal to the local *earth* magnetic field. Its horizontal component points towards the earth's magnetic north pole. The ratio between the horizontal and vertical component depends on the location on the earth and can be expressed in terms of the *dip angle* δ . In indoor environments, the magnetic field can locally be assumed to be constant and points towards a *local* magnetic north. This is not necessarily the earth's magnetic north pole. Choosing the navigation frame n such that the x -axis is pointing towards the local magnetic north, m^n can be parametrized in terms of its vertical component m_z^n

$$m^n = \begin{pmatrix} \sqrt{1 - (m_z^n)^2} & 0 & m_z^n \end{pmatrix}^T, \quad (8a)$$

or in terms of the dip angle δ

$$m^n = \begin{pmatrix} \cos \delta & 0 & -\sin \delta \end{pmatrix}^T. \quad (8b)$$

Note that the two parametrizations do not encode exactly the same knowledge about the magnetic field; the first component of m^n in (8a) is positive by construction while this is not true for (8b). However, both parametrizations will be used in the remainder. It will be argued that no information is lost by using (8b) if the parameter estimates are properly initialized.

The main need for magnetometer calibration arises from the fact that a magnetometer needs recalibration each time it is placed in a magnetically different environment. Specifically, a magnetometer measures a superposition of the local magnetic field and of the magnetic field due to the presence of magnetic material in the vicinity of the sensor. In case this magnetic material is rigidly attached to the magnetometer, it is possible to calibrate the magnetometer measurements for this. The magnetic material can give rise to both hard and soft iron contributions to the magnetic field. Hard iron effects are due to permanent magnetization of the magnetic material and lead to a constant 3×1 offset vector o_{hi} . Soft iron effects are due to magnetization of the material as a result of an external magnetic field and therefore depend on the orientation of the material with respect to the local magnetic field. We model this in terms of a 3×3 matrix C_{si} . Hence, the magnetometer measurements do not lie on a sphere as in (7), but instead, they lie on a

translated ellipsoid as

$$h_{m,t} = C_{si} R_t^{bn} m^n + o_{hi}. \quad (9)$$

As discussed in Section 2, when calibrating the magnetometer to obtain better orientation estimates, it is important that the magnetometer and the inertial sensor axes are aligned. Let us now be more specific about the definition of the body frame b and define it to be located in the center of the accelerometer triad and aligned with the accelerometer sensor axes. Furthermore, let us assume that the accelerometer and gyroscope axes are aligned. Defining the rotation between the body frame b and the magnetometer sensor frame b_m as $R^{b_m b}$, the model (9) can be extended to

$$h_{m,t} = C_{si} R^{b_m b} R_t^{bn} m^n + o_{hi}. \quad (10)$$

Finally, the magnetometer calibration can also correct for the presence of sensor errors in the magnetometer. These errors are sensor-specific and can differ for each individual magnetometer. They can be subdivided into three components, see e.g. Gebre-Egziabher et al. (2006); Renaudin et al. (2010); Vasconcelos et al. (2011):

1. Non-orthogonality of the magnetometer axes, represented by a matrix C_{no} .
2. Presence of a zero bias or null shift, implying that the magnetometer will measure a non-zero magnetic field even if the magnetic field is zero, defined by o_{zb} .
3. Difference in sensitivity of the three magnetometer axes, represented by a diagonal matrix C_{sc} .

We can therefore extend the model (10) to also include the magnetometer sensor errors as

$$h_{m,t} = C_{sc} C_{no} \left(C_{si} R^{b_m b} R_t^{bn} m^n + o_{hi} \right) + o_{zb}. \quad (11)$$

To obtain a correct calibration, it is fortunately not necessary to identify all individual contributions of the different components in (11). Instead, they can be combined into a 3×3 distortion matrix D and a 3×1 offset vector o where

$$D = C_{sc} C_{no} C_{si} R^{b_m b}, \quad (12a)$$

$$o = C_{sc} C_{no} o_{hi} + o_{zb}. \quad (12b)$$

The resulting magnetometer measurement model in (1b) can be written as

$$y_{m,t} = D R_t^{bn} m^n + o + e_{m,t}. \quad (13)$$

In deriving the model we have made two important assumptions:

Algorithm 1 Magnetometer and inertial calibration

1. Determine an initial parameter estimate $\hat{D}_0, \hat{o}_0, \hat{m}_0^n, \hat{\delta}_{\omega,0}, \hat{\Sigma}_{\omega,0}, \hat{\Sigma}_{a,0}, \hat{\Sigma}_{m,0}$ using three steps
 - (a) Initialize $\hat{\delta}_{\omega,0}, \hat{\Sigma}_{\omega,0}, \hat{\Sigma}_{a,0}, \hat{\Sigma}_{m,0}$.
 - (b) Obtain an initial $\tilde{D}_0$ and $\hat{o}_0$ based on ellipsoid fitting (see Section 6.1).
 - (c) Obtain initial $\hat{D}_0, \hat{o}_0$ and $\hat{m}_0^n$ by initial determination of the sensor axis misalignment (see Section 6.2).
2. Set $i = 0$ and repeat,
 - (a) Run the EKF using the current estimates $\hat{D}_i, \hat{o}_i, \hat{m}_i^n, \hat{\delta}_{\omega,i}, \hat{\Sigma}_{\omega,i}, \hat{\Sigma}_{a,i}, \hat{\Sigma}_{m,i}$ to obtain $\{\hat{y}_{t|t-1}(\hat{\theta}_i), S_t(\hat{\theta}_i)\}_{t=1}^N$ and evaluate the cost function in (6).
 - (b) Determine $\hat{\theta}_{i+1}$ using the numerical gradient of the cost function in (6), its approximate Hessian and a backtracking line search algorithm.
 - (c) Obtain $\hat{D}_{i+1}, \hat{o}_{i+1}, \hat{m}_{i+1}^n, \hat{\delta}_{\omega,i+1}, \hat{\Sigma}_{\omega,i+1}, \hat{\Sigma}_{a,i+1}, \hat{\Sigma}_{m,i+1}$ from $\hat{\theta}_{i+1}$.
 - (d) Set $i := i + 1$ and repeat from Step 2a until convergence.

Assumption 1. The calibration matrix D and offset vector o in (12) are assumed to be time-independent. This implies that we assume that the magnetic distortions are constant and rigidly attached to the sensor. Also, the inertial and the magnetometer sensor axes are assumed to be rigidly attached to each other, i.e. their misalignment is represented by a constant rotation matrix. Additionally, in our algorithm we will assume that their misalignment can be described by a rotation matrix, i.e. that their axes are not mirrored with respect to each other. _____

Assumption 2. The local magnetic field m^n is assumed to be constant. In outdoor environments, this is typically a physically reasonable assumption. In indoor environments, however, the local magnetic field can differ in different locations in the building and care should be taken to fulfill the assumption. _____

5 Calibration algorithm

In our magnetometer calibration algorithm we solve the optimization problem (6) to estimate the parameter vector θ . In this section we introduce the resulting calibration algorithm which is summarized in Algorithm 1. In Section 5.1, we first discuss our optimization strategy. A crucial part of this optimization strategy is the evaluation of the cost function. Some details related to this are discussed in Section 5.2. Finally, in Section 5.3 we introduce the parameter vector θ in more detail.

5.1 Optimization algorithm

The optimization problem (6) is solved in Step 2 of Algorithm 1. Standard unconstrained minimization techniques are used, which iteratively update the parameter estimates as

$$\theta_{i+1} = \theta_i - \alpha_i [\mathcal{H}(\theta_i)]^{-1} \mathcal{G}(\theta_i), \quad (14)$$

where the step *length* of the update at iteration i is denoted by α_i . The *direction* of the parameter update at iteration i is determined by $[\mathcal{H}(\theta_i)]^{-1} \mathcal{G}(\theta_i)$.

Typical choices for the search direction include choosing $\mathcal{G}(\theta_i)$ to be the gradient of the cost function in (6) and $\mathcal{H}(\theta_i)$ to be its Hessian. This leads to a Newton optimization algorithm. However, computing the gradient and Hessian of (6) is not straightforward. Possible approaches are discussed in Åström (1980); Segal and Weinstein (1989) for the case of linear models. In the case of nonlinear models, however, they only lead to approximate gradients, see e.g. Kok et al. (2015); Kokkala et al. (2015). For this reason we make use of a numerical approximation of $\mathcal{G}(\theta_i)$ instead and use a Broyden-Fletcher-Goldfarb-Shanno (BFGS) method with damped updating (Nocedal and Wright, 2006) to approximate the Hessian. Hence, the minimization is performed using a quasi-Newton optimization algorithm. A backtracking line search is used to find a good step length α_i .

Proper initialization of the parameters is crucial since the optimization problem (6) is non-convex. Step 1 summarizes the three-step process used to obtain good initial estimates of all parameters.

5.2 Evaluation of the cost function

An important part of the optimization procedure is the evaluation of the cost function in (6). This requires running an EKF using the state space model (1) to estimate the orientation of the sensor. This EKF uses the angular velocity ω_t as an input to the dynamic model (1a). An estimate of the angular velocity is obtained from the gyroscope measurements $y_{\omega,t}$ which are modeled as

$$y_{\omega,t} = \omega_t + \delta_{\omega} + e_{\omega,t}. \quad (15)$$

The measurement model (1b) entails the accelerometer measurements and the magnetometer measurements. The magnetometer measurement model can be found in (13). The accelerometer measurements $y_{a,t}$ are modeled as

$$y_{a,t} = R_t^{\text{bn}}(a_t^{\text{n}} - g^{\text{n}}) + e_{a,t} \approx -R_t^{\text{bn}} g^{\text{n}} + e_{a,t}, \quad (16)$$

where a_t^{n} denotes the sensor's acceleration in the navigation frame and g^{n} denotes the earth's gravity. The rotation matrix R_t^{bn} has previously been introduced in Section 4.

The state in the EKF, which represents the sensor orientation, can be parametrized in different ways. In previous work we have used a quaternion representation as a 4-dimensional state vector (Kok and Schön, 2014). In this work we instead use an implementation of the EKF, which is sometimes called a multiplicative

EKF (Markley, 2003; Crassidis et al., 2007; Hol, 2011). Here, a 3-dimensional state vector represents the *orientation deviation from a linearization point*. More details on this implementation can be found in Kok (2014).

The EKF returns the one step ahead predicted measurements $\{\hat{y}_{t|t-1}(\theta)\}_{t=1}^N$ and their covariance $\{S_t(\theta)\}_{t=1}^N$ which can be used to evaluate (6). The cost function needs to be evaluated for the current parameter estimates in Step 2a but also needs to be evaluated once for each component of the parameter vector θ to compute the numerical gradient. Hence, each iteration i requires running the EKF at least $n_\theta + 1$ times. Note that the actual number of evaluations can be higher since the backtracking line search algorithm used to determine α_i can require a varying number of additional evaluations. Since $n_\theta = 34$, computing the numerical gradient is computationally rather expensive. However, it is possible to parallelize the computations.

5.3 The parameter vector θ

As apparent from Section 4, our main interest lies in determining the calibration matrix D and the offset vector o , which can be used to correct the magnetometer measurements to obtain more accurate orientation estimates. To solve the calibration problem, however, we also estimate a number of other parameters.

First, the local magnetic field m^n introduced in Section 4 is in general scenarios unknown and needs to be estimated. In outdoor environments, m^n is equal to the local earth magnetic field and is accurately known from geophysical studies, see e.g. National Centers for Environmental Information (2015). In indoor environments, however, the local magnetic field can differ quite significantly from the local earth magnetic field. Because of that, we treat m^n as an unknown constant. Second, the gyroscope measurements that are used to describe the change in orientation of the sensor in (1a) are corrupted by a bias δ_ω . This bias is slowly time varying but for our relatively short experiments it can be assumed to be constant. Hence, it is treated as part of the parameter vector θ . Finally, we treat the noise covariance matrices Σ_ω , Σ_a and Σ_m as unknown. In summary, the parameter vector θ consists of

$$D \in \mathbb{R}^{3 \times 3}, \quad (17a)$$

$$o \in \mathbb{R}^3, \quad (17b)$$

$$m^n \in \{\mathbb{R}^3 : \|m^n\|_2^2 = 1, m_x^n > 0, m_y^n = 0\}, \quad (17c)$$

$$\delta_\omega \in \mathbb{R}^3, \quad (17d)$$

$$\Sigma_\omega \in \{\mathbb{R}^{3 \times 3} : \Sigma_\omega \geq 0, \Sigma_\omega = \Sigma_\omega^T\}, \quad (17e)$$

$$\Sigma_a \in \{\mathbb{R}^{3 \times 3} : \Sigma_a \geq 0, \Sigma_a = \Sigma_a^T\}, \quad (17f)$$

$$\Sigma_m \in \{\mathbb{R}^{3 \times 3} : \Sigma_m \geq 0, \Sigma_m = \Sigma_m^T\}, \quad (17g)$$

where m_x^n and m_y^n denote the x - and y - component of m^n , respectively. The notation $\Sigma \geq 0$ denotes the assumption that the matrix Σ is positive semi-definite.

Although (17c) and (17e) – (17g) suggest that constrained optimization is needed, it is possible to circumvent this via suitable reparametrizations. The covariance matrices can be parametrized in terms of their Cholesky factorization, leading to only 6 parameters for each 3×3 covariance matrix. The local magnetic field can be parametrized using only one parameter as in (8). Note that in our implementation we prefer to use the representation (8b) for the ML problem (6). Although this latter parametrization does not account for the constraint $m_x^n > 0$, this is of no concern due to proper initialization. The procedure to obtain good initial estimates of all parameters is the topic of the next section.

6 Finding good initial estimates

Since the optimization problem is non-convex, the parameter vector θ introduced in Section 5 needs proper initialization. An initial estimate $\hat{\theta}_0$ is obtained using a three-step method. As a first step, the gyroscope bias δ_ω and the noise covariances of the inertial sensors, Σ_ω , Σ_a , and of the magnetometer, Σ_m , are initialized. This is done using a short batch of stationary data. Alternatively, they can be initialized based on prior sensor knowledge. As a second step, described in Section 6.1, an ellipsoid fitting problem is solved using the magnetometer data. This maps the ellipsoid of data to a sphere but can not determine the rotation of the sphere. The rotation of the sphere is determined in a third step of the initialization procedure. This step also determines an initial estimate of the normalized local magnetic field m^n .

6.1 Ellipsoid fitting

Using the definition of the normalized local magnetic field m^n , we would expect all calibrated magnetometer measurements to lie on the unit sphere,

$$\begin{aligned} \|m^n\|_2^2 - 1 &= \|R_t^{\text{bn}} m^n\|_2^2 - 1 \\ &= \|D^{-1} (y_{m,t} - o - e_{m,t})\|_2^2 - 1 = 0. \end{aligned} \quad (18)$$

In practice, the measurements are corrupted by noise and the equality (18) does not hold exactly. The ellipsoid fitting problem can therefore be written as

$$y_{m,t}^T A y_{m,t} + b^T y_{m,t} + c \approx 0, \quad (19)$$

with

$$A \triangleq D^{-T} D^{-1}, \quad (20a)$$

$$b \triangleq -2o^T D^{-T} D^{-1}, \quad (20b)$$

$$c \triangleq o^T D^{-T} D^{-1} o. \quad (20c)$$

Assuming that the matrix A is positive definite, this can be recognized as the definition of an ellipsoid with parameters A , b and c (see e.g. Gander et al. (1994)).

We can rewrite (19) as a linear relation of the parameters as

$$M\xi \approx 0, \quad (21)$$

with

$$M = \begin{pmatrix} y_{m,1} \otimes y_{m,1} & y_{m,1} & 1 \\ y_{m,2} \otimes y_{m,2} & y_{m,2} & 1 \\ \vdots & \vdots & \vdots \\ y_{m,N} \otimes y_{m,N} & y_{m,N} & 1 \end{pmatrix}, \quad \xi = \begin{pmatrix} \text{vec } A \\ b \\ c \end{pmatrix}, \quad (22)$$

where $\otimes$ denotes the Kronecker product and vec denotes the vectorization operator. This problem has infinitely many solutions and without constraining the length of the vector ξ , the trivial solution $\xi = 0$ would be obtained. A possible approach to solve the ellipsoid fitting problem is to make use of a singular value decomposition (Gander et al., 1994; Kok et al., 2012). This approach inherently poses a length constraint on the vector ξ , assuming that its norm is equal to 1. It does, however, not guarantee positive definiteness of the matrix A . Although positive definiteness of A is not guaranteed, there are only very few practical scenarios in which the estimated matrix A will not be positive definite. A non-positive definite matrix A can for instance be obtained in cases of very limited rotation of the sensor. The problem of allowing a non-positive definite matrix A can be circumvented by instead solving the ellipsoid fitting problem as a semidefinite program (Calafiore, 2002; Boyd and Vandenberghe, 2004)

$$\begin{aligned} \min_{A,b,c} \quad & \frac{1}{2} \|M \begin{pmatrix} \text{vec } A \\ b \\ c \end{pmatrix}\|_2^2, \\ \text{s.t.} \quad & \text{Tr } A = 1, \quad A \in S_{++}^{3 \times 3}, \end{aligned} \quad (23)$$

where $S_{++}^{3 \times 3}$ denotes the set of 3×3 positive definite symmetric matrices. By constraining the trace of the matrix A , (23) avoids the trivial solution of $\xi = 0$. The problem (23) is a convex optimization problem and therefore has a globally optimal solution and does not require an accurate initial guess of the parameter vector ξ . The optimization problem can easily be formulated and efficiently solved using freely available software packages like YALMIP (Löfberg, 2004) or CVX (Grant and Boyd, 2013).

Initial estimates of the calibration matrix D and the offset vector o can be obtained from the estimated $\hat{A}, \hat{b}, \hat{c}$ as

$$\beta = \left(\frac{1}{4} \hat{b}^\top \hat{A}^{-1} \hat{b} - \hat{c} \right)^{-1}, \quad (24a)$$

$$\tilde{D}_0^\top \tilde{D}_0 = \beta \hat{A}^{-1}, \quad (24b)$$

$$\hat{o}_0 = \frac{1}{2} \hat{A}^{-1} \hat{b}, \quad (24c)$$

where $\hat{o}_0$ denotes the initial estimate of the offset vector o . From (24b) it is not possible to uniquely determine the initial estimate of the calibration matrix D .

We determine an initial estimate of the calibration matrix D using a Cholesky decomposition, leading to a lower triangular $\tilde{D}_0$. However, any $\tilde{D}_0 U$ where $UU^\top = \mathcal{I}_3$ will also fulfill (24b). As discussed in Assumption 1 in Section 4, we assume that the sensor axes of the inertial sensors and the magnetometers are related by a rotation, implying that we restrict the matrix U to be a rotation matrix. The initial estimate $\tilde{D}_0$ can therefore be defined in terms of $\tilde{D}_0$ as

$$\hat{D}_0 = \tilde{D}_0 R_D. \quad (25)$$

The unknown rotation matrix R_D will be determined in Section 6.2.

6.2 Determine misalignment of the inertial and magnetometer sensor axes

The third step of the initial estimation aims at determining the misalignment between the inertial and the magnetometer sensor axes. It also determines an initial estimate of the normalized local magnetic field $\hat{m}_0^n$. These estimates are obtained by combining the magnetometer measurements with the inertial sensor measurements. The approach is based on the fact that the inner product of two vectors is invariant under rotation. The two vectors considered here are m^n and the vertical $v^n = \begin{pmatrix} 0 & 0 & 1 \end{pmatrix}^\top$. Hence, it is assumed that the inner product of the vertical v_t^b in the body frame b ,

$$v_t^b = R_t^{bn} v^n, \quad (26a)$$

and the normalized local magnetic field m_t^b in the body frame,

$$m_t^b = R_D^\top \tilde{D}_0^{-1} (y_{m,t} - \delta_0), \quad (26b)$$

is constant. The matrix R_D in (26b) denotes the rotation needed to align the inertial and magnetometer sensor axes. The rotation matrix R_t^{nb} in (26a) is a rotation matrix representation of the orientation estimate at time t obtained from an EKF. This EKF is similar to the one described in Section 5.2. It does not use the magnetometer measurements, since they have not properly been calibrated yet and can therefore not result in accurate heading estimates. However, to determine the vertical v_t^b , only the sensor's inclination is of concern, which can be determined using the inertial measurements only.

The inner product between m^n and v^n is equal to m_z^n (see also (8a)). Since this inner product is invariant under rotation, we can formulate the following minimization problem

$$\begin{aligned} \min_{R_D, m_{z,0}^n} \quad & \frac{1}{2} \sum_{t=1}^N \|m_{z,0}^n - (v^n)^\top R_t^{nb} R_D^\top \tilde{D}_0^{-1} (y_{m,t} - \delta_0)\|_2^2, \\ \text{s.t.} \quad & R_D \in \text{SO}(3). \end{aligned} \quad (27)$$

The rotation matrix R_D can be parametrized using an orientation deviation from a linearization point similar to the approach described in Section 5.2. Hence, (27) can be solved as an unconstrained optimization problem.



Figure 2: Left: experimental setup where a calibration experiment is performed outdoors. An Xsens MTi-100 IMU (orange box) together with a magnetic disturbance is placed in an aluminum block. Right: experimental setup using a Trivisio Colibri Wireless IMU (black box). A phone is used as a source of magnetic disturbance. To avoid saturation of the magnetometer, the phone is not attached directly to the IMU.

Based on these results and (25) we obtain the following initial estimates

$$\hat{D}_0 = \bar{D}_0 \hat{R}_D, \quad (28a)$$

$$\hat{m}_0^n = \begin{pmatrix} \sqrt{1 - (\hat{m}_{z,0}^n)^2} & 0 & \hat{m}_{z,0}^n \end{pmatrix}^T. \quad (28b)$$

Hence, we have obtained an initial estimate $\hat{\theta}_0$ of the entire parameter vector θ as introduced in Section 5.

7 Experimental results

7.1 Experimental setup

Experiments have been performed using two commercially available inertial measurements units (IMUs), an Xsens MTi-100 (Xsens Technologies B. V., 2016) and a Trivisio Colibri Wireless IMU (Trivisio Prototyping GmbH, 2016). The experimental setup of both experiments can be found in Figure 2. The experiment with the Xsens IMU was performed outdoors to ensure a homogeneous local magnetic field. The experiment with the Trivisio IMU was performed indoors. However, the experiment was performed relatively far away from any magnetic materials such that the local magnetic field is as homogenous as possible. The Xsens IMU was placed in an aluminum block with right angles which can be used to rotate the sensor 90° to verify the heading results. For both sensors, inertial and magnetometer measurements were collected at 100 Hz.

7.2 Calibration results

For calibration, the IMU needs to be slowly rotated such that the assumption of zero acceleration is reasonably valid. This leads to an ellipsoid of magnetometer

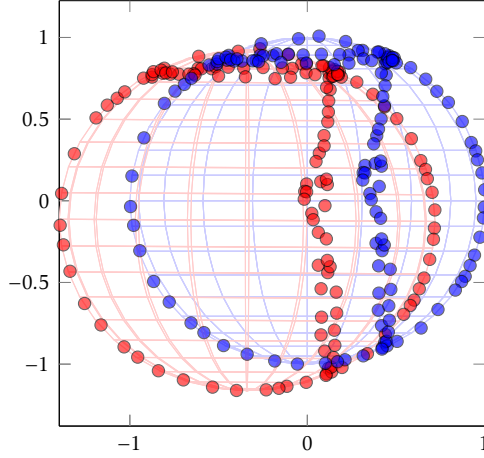


Figure 3: Calibration results from the experiment with the Trivisio IMU. The ellipsoid of magnetometer data (red) lies on a unit sphere after calibration (blue).

data as depicted in red in Figures 1 and 3. Note that for plotting purposes the data has been downsampled to 1 Hz. To emphasize the deviation of the norm from 1, the norm of the magnetometer data is depicted in red in Figure 4 for both experiments.

For the experiment with the Xsens IMU, the following calibration matrix $\hat{D}$ and offset vector $\hat{o}$ are found

$$\hat{D} = \begin{pmatrix} 0.74 & -0.13 & 0.01 \\ -0.12 & 0.68 & 0.01 \\ -0.03 & 0.43 & 1.00 \end{pmatrix}, \quad \hat{o} = \begin{pmatrix} 1.36 \\ 1.22 \\ -0.94 \end{pmatrix} \quad (29)$$

using Algorithm 1. Applying the calibration result to the magnetometer data leads to the unit sphere of data in blue in Figure 1. The norm of the magnetometer data after calibration can indeed be seen to lie around 1, as depicted in blue in Figure 4.

As a measure of the calibration quality, we analyze the normalized residuals $S_t^{-1/2}(y_t - \hat{y}_{t|t-1})$ after calibration from the EKF. For each time t , this is a vector in $\mathbb{R}^6$. In the case of correctly calibrated parameters that sufficiently model the magnetic disturbances, we expect the stacked normalized residuals $\{S_t^{-1/2}(y_t - \hat{y}_{t|t-1})\}_{t=1}^N \in \mathbb{R}^{6N}$ to be normally distributed with zero mean and standard deviation 1. The histogram and a fitted Gaussian distribution can be found in Figure 5a. The residuals resemble a $\mathcal{N}(0, 1)$ distribution except for the large peak around zero and – not visible in the plot – a small amount of outliers outside of the plotting interval. This small amount of outliers is due to the fact that there are a few measurement outliers in the accelerometer data. Large accelerations can for instance be measured when the setup is accidentally bumped into something and violate our assumption that the acceleration of the sensor is approximately

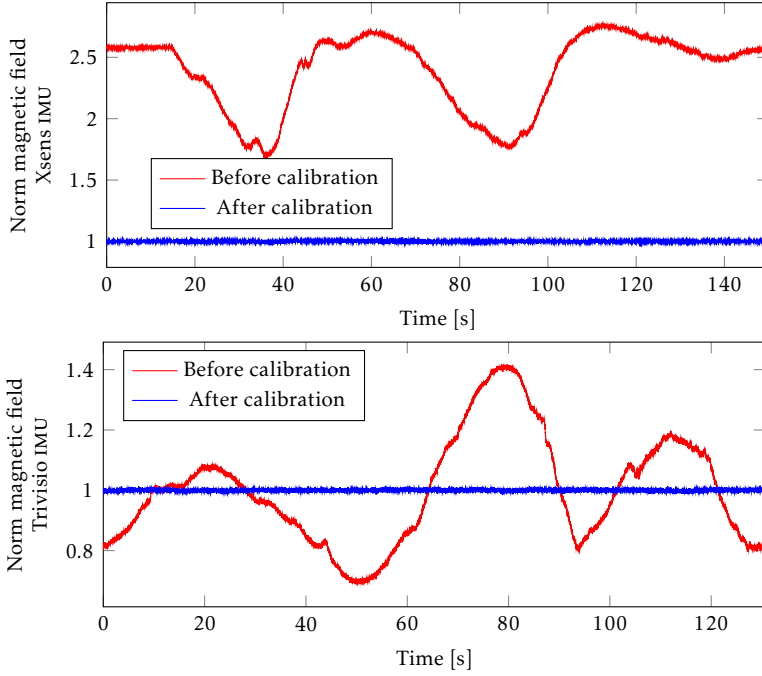


Figure 4: Norm of the magnetic field measurements before (red) and after (blue) calibration for (top) the experiment with the Xsens IMU and for (bottom) the experiment with the Trivisio IMU.

zero. We believe that the peak around zero is due to the fact that the algorithm compensates for the presence of the large residuals.

To analyze if the calibration is also valid for a different (validation) data set with the same experimental setup, the calibrated parameters have been used on a second data set. Figures of the ellipsoid of magnetometer data and the sphere of calibrated magnetometer data are not included since they look very similar to Figures 1 and 4. The residuals after calibration of this validation data set can be found in Figure 5b. The fact that these residuals look very similar to the ones for the original data suggests that the calibration parameters obtained are also valid for this validation data set.

The Trivisio IMU outputs the magnetometer data in microtesla. Since our algorithm scales the calibrated measurements to a unit norm, the obtained $\hat{D}$ and offset vector $\hat{o}$ from Algorithm 1 are in this case of much larger magnitude,

$$\hat{D} = \begin{pmatrix} 61.74 & 0.59 & 0.09 \\ -1.01 & 60.74 & 0.23 \\ -0.39 & 0.06 & 60.80 \end{pmatrix}, \quad \hat{o} = \begin{pmatrix} -19.77 \\ -1.68 \\ -6.98 \end{pmatrix}. \quad (30)$$

The sphere of calibrated data and its norm can be found in blue in Figures 3 and 4. Note that for plotting purposes, the magnetometer data before calibration is scaled

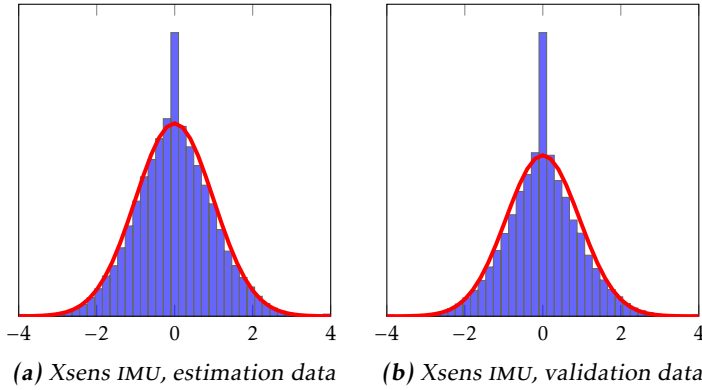


Figure 5: Histogram of the normalized residuals $S_t^{-1/2}(y_t - \hat{y}_{t|t-1})$ from the EKF after calibration for the estimation data set (left) and for a validation data set (right) for the experiments performed with the Xsens IMU. A Gaussian distribution (red) is fitted to the data.

such that its mean lies around 1. The obtained $\hat{D}$ and $\hat{\sigma}$ are scaled accordingly to plot the red ellipsoid in Figure 3. The normalized residuals $S_t^{-1/2}(y_t - \hat{y}_{t|t-1})$ of the EKF using both the estimation and a validation data set are depicted in Figure 6. For this data set, the accelerometer data does not contain any outliers and the residuals resemble a $\mathcal{N}(0, 1)$ distribution fairly well.

From these results we can conclude that Algorithm 1 gives good magnetometer calibration results for experimental data from two different commercially available IMUs. A good fit of the ellipsoid of data to a sphere is obtained and the algorithm seems to give good estimates analyzed in terms of its normalized residuals. Since magnetometer calibration is generally done to obtain improved heading estimates, it is important to also interpret the quality of the calibration in terms of the resulting heading estimates. In Section 7.3 this will be done based on experimental results. The heading performance will also be analyzed based on simulations in Section 8.

7.3 Heading estimation

An important goal of magnetometer calibration is to facilitate good heading estimates. To check the quality of the heading estimates after calibration, the block in which the Xsens IMU was placed (shown in Figure 2) is rotated around all axes. This block has right angles and it can therefore be placed in 24 orientations that differ from each other by 90 degrees. The experiment was conducted in Enschede, the Netherlands. The dip angle δ at this location is approximately 67° (National Centers for Environmental Information, 2015). Hence, we expect the calibrated magnetometer measurements to resemble rotations of the normalized magnetic field $m^n = \begin{pmatrix} 0.39 & 0 & -0.92 \end{pmatrix}^\top$ (see also (7) and (8b)). The calibrated magnetome-

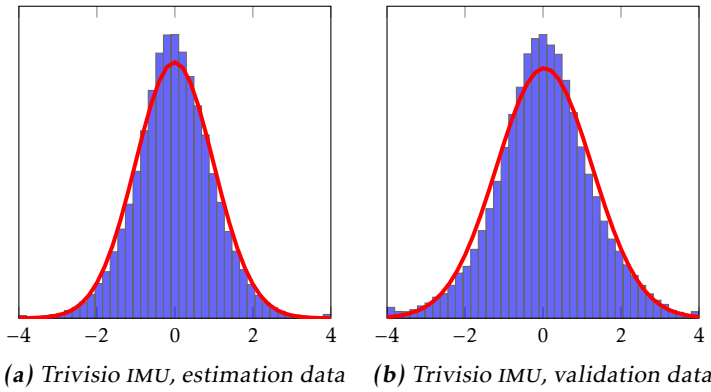


Figure 6: Histogram of the normalized residuals $S_t^{-1/2}(y_t - \hat{y}_{t|t-1})$ from the EKF after calibration for the estimation data set (left) and for a validation data set (right) for the experiments performed with the Trivisio IMU. A Gaussian distribution (red) is fitted to the data.

ter data from the experiment is shown in Figure 7 and consists of the following stationary time periods:

z-axis up During the period 0 – 105s, the magnetometer is flat with its z-axis pointing upwards. Hence, the z-axis (red) of the magnetometer measures the vertical component of the local magnetic field m_z^n . During this period, the sensor is rotated by 90° around the z-axis into 4 different orientations and subsequently back to its initial orientation. This results in the 5 steps for measurements in the x- (blue) and y-axis (green) of the magnetometer.

z-axis down A similar rotation sequence is performed with the block upside down at 110 – 195s, resulting in a similar pattern for measurements in the x- and y-axis of the magnetometer. During this time period, the z-axis of the magnetometer measures $-m_z^n$ instead.

x-axis up The procedure is repeated with the x-axis of the sensor pointing upwards during the period 200 – 255s, rotating around the x-axis into 4 different orientations and back to the initial position. This results in the 5 steps for measurements in the y- and z-axis of the magnetometer.

x-axis down A similar rotation sequence is performed with the x-axis pointing downwards at 265 – 325 seconds.

y-axis down Placing the sensor with the y-axis downwards and rotating around the y-axis results in the data at 350 – 430 seconds. The rotation results in the 5 steps for measurements in the x- and z-axis of the magnetometer.

y-axis up A similar rotation sequence is performed with the y-axis pointing upwards at 460 – 520 seconds.

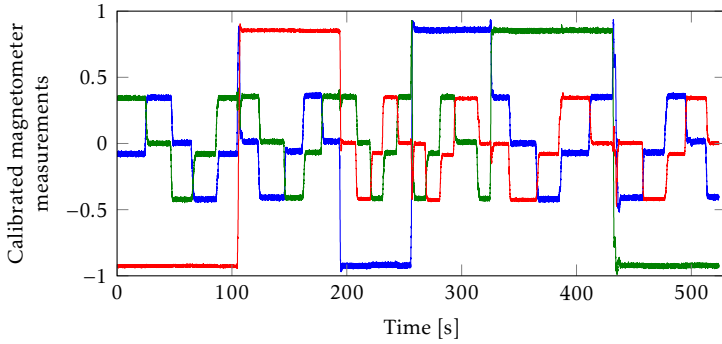


Figure 7: Calibrated magnetometer data of an experiment rotating the sensor into 24 different sensor orientations where the blue, green and red lines represent the data from the x -, y - and z -axis of the magnetometer, respectively.

Since the experimental setup was not placed *exactly* vertical, it is not possible to compare the *absolute* orientations. However, it is possible to compare the *difference in orientation* which is known to be 90° due to the properties of the block in which the sensor was placed. To exclude the effect of measurement noise, for each of the stationary periods in Figure 7, 500 samples of magnetometer and accelerometer data are selected. Their mean values are used to estimate the orientation of the sensor. Here, the accelerometer data is used to estimate the inclination. The heading is estimated from the horizontal component of the magnetometer data. This procedure makes use of the fact that the orientation of the sensor can be determined from two linearly independent vectors in the navigation frame – the gravity and the direction of the magnetic north – and in the body frame – the mean accelerometer and magnetometer data. It is referred to as the TRIAD algorithm (Shuster and Oh, 1981). Table 1 reports the deviation from 90° between two subsequent rotations. Note that the metal object causing the magnetic disturbance as shown in Figure 2 physically prevents the setup from being properly placed in all orientations around the y -axis. Rotation around the y -axis with the y -axis pointing upwards has therefore not been included in Table 1.

Our experiment investigates both the heading errors and the improvement of the heading estimates over the ones obtained after the initial calibration, i.e. Step 1 in Algorithm 1. In Table 1 we therefore include both the heading errors using the initial parameter estimates $\hat{D}_0$ (28a) and $\hat{o}_0$ (24c) and the heading errors using ML parameter estimates $\hat{D}$ and $\hat{o}$ (29) obtained using Algorithm 1. As can be seen, the deviation from 90° is small, indicating that good heading estimates are obtained after calibration. Also, the heading estimates using the initial parameter estimates are already fairly good. The mean error is reduced from 1.28° for the initial estimate to 0.76° for the ML estimate. The maximum error is reduced from 4.36° for the initial estimate to 2.48° for the ML estimate. Note that the results of the ML estimate from Algorithm 1 are slightly better than the results previously reported by Kok and Schön (2014). This can be attributed to the fact that

Table 1: Difference in estimated heading between two subsequent rotations around the sensor axes using calibrated magnetometer data. The values represent the deviation in degrees from 90° . Included are both the results using the ML estimates from Algorithm 1 and the results using initial estimates from Step 1 in the algorithm.

z-axis				x-axis				y-axis	
z up		z down		x up		x down		y down	
ML	init	ML	init	ML	init	ML	init	ML	init
0.11	0.36	0.69	1.34	0.22	0.16	0.86	1.01	0.18	1.57
0.22	0.90	2.48	4.36	0.07	0.20	1.57	1.45	0.29	0.76
0.46	1.52	1.53	3.57	0.97	0.94	0.61	0.71	0.20	0.78
0.30	0.94	1.92	2.40	0.29	0.59	1.78	1.70	0.50	0.45

we now use orientation error states instead of the quaternion states in the EKF (see Section 5.2). This results in slightly better estimates, but also in a smoother convergence of the optimization problem. The quality of the heading estimates is studied further in Section 8 based on a simulation study.

8 Simulated heading accuracy

Magnetometer calibration is typically performed to improve the heading estimates. It is, however, difficult to check the heading accuracy experimentally. In Section 7.3, for instance, we are limited to doing the heading validation on a different data set and we have a limited number of available data points. To get more insight into the orientation accuracy that is gained by executing all of Algorithm 1, compared to just its initialization phase (Step 1 in the algorithm), we engage in a simulation study. In this study we focus on the root mean square (RMS) heading error for different simulated sensor qualities (in terms of the noise covariances and the gyroscope bias) and different magnetic field disturbances (in terms of different values for the calibration matrix D and offset vector o).

In our simulation study, we assume that the local magnetic field is equal to that in Linköping, Sweden. The calibration matrix D , the offset vector o and the sensor properties in terms of the gyroscope bias and noise covariances are all sampled from a uniform distribution. The parameters of the distributions from which the sensor properties are sampled are chosen as physically reasonable values as considered from the authors' experience. The noise covariance matrices Σ_ω , Σ_a and Σ_m are assumed to be diagonal with three different values on the diagonal. The calibration matrix D is assumed to consist of three parts,

$$D = D_{\text{diag}} D_{\text{skew}} D_{\text{rot}}, \quad (31)$$

Table 2: Settings used in the Monte Carlo simulation.

D_{diag}	D_{skew}	D_{rot}	o	δ_ω
D_{11}, D_{22}, D_{33} $\sim \mathcal{U}(0.5, 1.5)$	ζ, η, ρ $\sim \mathcal{U}(-30^\circ, 30^\circ)$	ψ, θ, ϕ $\sim \mathcal{U}(-10^\circ, 10^\circ)$	o_1, o_2, o_3 $\sim \mathcal{U}(-1, 1)$	$\delta_{\omega,1}, \delta_{\omega,2}, \delta_{\omega,3}$ $\sim \mathcal{U}(-1, 1)$

Σ_ω	Σ_a	Σ_m
$\Sigma_{\omega,1}, \Sigma_{\omega,2}, \Sigma_{\omega,3}$ $\sim \mathcal{U}(10^{-3}, 10^{-2})$	$\Sigma_{a,1}, \Sigma_{a,2}, \Sigma_{a,3}$ $\sim \mathcal{U}(10^{-3}, 10^{-1})$	$\Sigma_{m,1}, \Sigma_{m,2}, \Sigma_{m,3}$ $\sim \mathcal{U}(10^{-3}, 10^{-1})$

where D_{diag} is a diagonal matrix with elements D_{11}, D_{22}, D_{33} and D_{rot} is a rotation matrix around the angles ψ, θ, ϕ . The matrix D_{skew} models the non-orthogonality of the magnetometer axes as

$$D_{\text{skew}} = \begin{pmatrix} 1 & 0 & 0 \\ \sin \zeta & \cos \zeta & 0 \\ -\sin \eta & \cos \eta \sin \rho & \cos \eta \cos \rho \end{pmatrix}, \quad (32)$$

where the angles ζ, η, ρ represent the different non-orthogonality angles. The exact simulation conditions are summarized in Table 2.

The simulated data consists of 100 samples of stationary data and subsequently 300 samples for rotation around all three axes. It is assumed that the rotation is exactly around the origin of the accelerometer triad, resulting in zero acceleration during the rotation. The first 100 samples are used to obtain an initial estimate of the gyroscope bias $\hat{\delta}_{\omega,0}$ by computing the mean of the stationary gyroscope samples. The covariance matrices $\hat{\Sigma}_{\omega,0}$, $\hat{\Sigma}_{a,0}$ and $\hat{\Sigma}_{m,0}$ are initialized based on the covariance of these first 100 samples. The initial estimate then consists of these initial estimates $\hat{\delta}_{\omega,0}$, $\hat{\Sigma}_{\omega,0}$, $\hat{\Sigma}_{a,0}$, $\hat{\Sigma}_{m,0}$ and the initial calibration matrix $\hat{D}_0$ (28a), the initial offset vector $\hat{o}_0$ (24c) and the initial estimate of the local magnetic field m_0^n (28b).

To study the heading accuracy, the EKF as described in Section 5.2 is run with both the initial parameter values $\hat{\theta}_0$ and their ML values $\hat{\theta}^{\text{ML}}$. The orientation errors Δq_t , encoded as a unit quaternion are computed using

$$\Delta q_t = \hat{q}_t^{\text{nb}} \odot \left(q_{\text{ref},t}^{\text{nb}} \right)^c, \quad (33)$$

where $\odot$ denotes a quaternion multiplication and the superscript c denotes the quaternion conjugate (see e.g. Hol (2011)). It is computed from the orientation $\hat{q}_t^{\text{nb}}$ estimated by the EKF and the ground truth orientation $q_{\text{ref},t}^{\text{nb}}$. Computing the orientation errors in this way is equivalent to subtracting Euler angles in the case of small angles. However, it avoids subtraction problems due to ambiguities in the Euler angles representation. To interpret the orientation errors Δq_t , they are

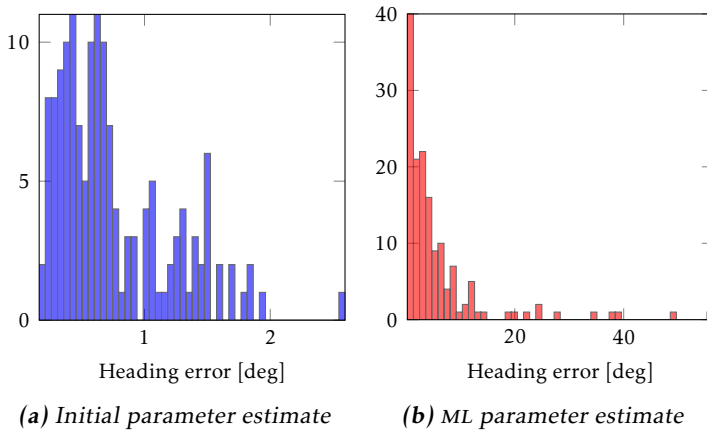


Figure 8: Histogram of the heading RMSE using the ML parameter estimate from Algorithm 1 (left, blue) and the initial parameter estimate from Step 1 in the algorithm (right, red). Note the different scales in the two plots.

converted to Euler angles. We focus our analysis on the heading error, i.e. on the third component of the Euler angles.

The RMS of the heading error is plotted for 150 Monte Carlo simulations in Figure 8. As can be seen, the heading root mean square error (RMSE) using the estimate of the calibration parameters from Algorithm 1 is consistently small. The heading RMSE based on the initialization phase in Step 1 of the algorithm, however, has a significantly larger spread. This clearly shows that orientation accuracy can be gained by executing all of Algorithm 1. Note that in all simulations, analysis of the norm of the calibrated magnetometer measurements as done in Figure 4 does not indicate that the ML estimate is to be preferred over the estimate from the initialization phase. Hence, analysis of the norm of the calibrated magnetometer measurements does not seem to be a sufficient analysis to determine the quality of the calibration in the case when the calibration is performed to improve the heading estimates.

9 Conclusions

We have developed a practical algorithm to calibrate a magnetometer using inertial sensors. It calibrates the magnetometer for the presence of magnetic disturbances, for magnetometer sensor errors and for misalignment between the inertial and magnetometer sensor axes. The problem is formulated as an ML problem. The algorithm is shown to perform well on real data collected with two different commercially available inertial measurement units.

In future work the approach can be extended to include GPS measurements. In that case it is not necessary to assume that the acceleration is zero. The algorithm

can hence be applied to a wider range of problems, like for instance the flight test example discussed in Kok et al. (2012). The computational cost of the algorithm would, however, increase, since to facilitate the inclusion of the GPS measurements, the state vector in the EKF needs to be extended.

Another interesting direction for future work would be to investigate ways of reducing the computational cost of the algorithm. The computational cost of the initialization steps is very small but actually solving the ML problem in Step 2 of Algorithm 1 is computationally expensive. The algorithm both needs quite a large number of iterations and each iteration is fairly expensive due to the computation of the numerical gradients. Interesting lines of future work would either explore different optimization methods or different ways to obtain gradient estimates.

Finally, it would be interesting to extend the work to online estimation of calibration parameters. This would allow for a slowly time-varying magnetic field and online processing of the data.

Acknowledgements

This work is supported by CADICS, a Linnaeus Center, and by the project *Probabilistic modeling of dynamical systems* (Contract number: 621-2013-5524), both funded by the Swedish Research Council (VR), and by MC Impulse, a European Commission, FP7 research project. The authors would like to thank Laurens Slot, Dr. Henk Luinge and Dr. Jeroen Hol from Xsens Technologies and Dr. Gustaf Hendebý from Linköping University for their support in collecting the data sets and for interesting discussions. The authors would also like to thank the reviewers for their constructive comments.

Bibliography

- R. Alonso and M. D. Shuster. Complete linear attitude-independent magnetometer calibration. *The Journal of the Astronautical Sciences*, 50(4):477–490, 2002.
- K. J. Åström. Maximum likelihood and prediction error methods. *Automatica*, 16(5):551–574, 1980.
- S. Bonnet, C. Bassompierre, C. Godin, S. Lesecq, and A. Barraud. Calibration methods for inertial and magnetic sensors. *Sensors and Actuators A: Physical*, 156(2):302–311, 2009.
- N. Bowditch. *The American Practical Navigator*. United States government, 2002.
- S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.
- G. Calafiore. Approximation of n-dimensional data using spherical and ellipsoidal primitives. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 32(2):269–278, 2002.
- J. L. Crassidis, F. L. Markley, and Y. Cheng. A survey of nonlinear attitude estimation methods. *Journal of Guidance, Control, and Dynamics*, 30(1):12–28, 2007.
- W. Gander, G. H. Golub, and R. Strebel. Least-squares fitting of circles and ellipses. *BIT Numerical Mathematics*, 34(4):558–578, 1994.
- D. Gebre-Egziabher, G. H. Elkaim, J. D. Powell, and B. W. Parkinson. Calibration of strapdown magnetometers in magnetic field domain. *Journal of Aerospace Engineering*, 19(2):87–102, April 2006.
- M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.0 beta. <http://cvxr.com/cvx>, September 2013.
- F. Gustafsson. *Statistical Sensor Fusion*. Studentlitteratur, 2012.
- J. D. Hol. *Sensor fusion and calibration of inertial sensors, vision, ultra-wideband and GPS*. Dissertation no. 1368, Linköping University, Linköping, Sweden, June 2011.
- M. Kok. *Probabilistic modeling for positioning applications using inertial sensors*. Licentiate’s thesis no. 1656, Linköping University, Linköping, Sweden, June 2014.
- M. Kok and T. B. Schön. Maximum likelihood calibration of a magnetometer using inertial sensors. In *Proceedings of the 19th World Congress of the International Federation of Automatic Control*, pages 92–97, Cape Town, South Africa, August 2014.
- M. Kok and T. B. Schön. Magnetometer calibration using inertial sensors. *IEEE Sensors Journal*, 16(14):5679 – 5689, 2016.

- M. Kok, J. D. Hol, T. B. Schön, F. Gustafsson, and H. Luinge. Calibration of a magnetometer in combination with inertial sensors. In *Proceedings of the 15th International Conference on Information Fusion*, pages 787–793, Singapore, July 2012.
- M. Kok, J. Dahlin, T. B. Schön, and A. Wills. Newton-based maximum likelihood estimation in nonlinear state space models. In *Proceedings of the 17th IFAC Symposium on System Identification*, pages 398–403, Beijing, China, October 2015.
- J. Kokkala, A. Solin, and S. Särkkä. Sigma-point filtering and smoothing based parameter estimation in nonlinear dynamic systems. *Pre-print*, 2015. arXiv:1504.06173.
- G. M. Lerner. Scalar checking. In J. R. Wertz, editor, *Spacecraft attitude determination and control*, pages 328–334. D. Reidel Publishing Company, Dordrecht, Holland, 1978.
- X. Li and Z. Li. A new calibration method for tri-axial field sensors in strap-down navigation systems. *Measurement Science and Technology*, 23(10), October 2012.
- F. Lindsten and T. B. Schön. Backward simulation methods for Monte Carlo statistical inference. *Foundations and Trends in Machine Learning*, 6(1):1–143, 2013.
- L. Ljung. *System Identification, Theory for the User*. Prentice Hall PTR, 2nd edition, 1999.
- J. Löfberg. YALMIP: A toolbox for modeling and optimization in MATLAB. In *Proceedings of the IEEE International Symposium on Computer Aided Control Systems Design, 2004*, pages 284 – 289, Taipei, Taiwan, September 2004.
- F. L. Markley. Attitude error representations for Kalman filtering. *Journal of guidance, control, and dynamics*, 26(2):311–317, 2003.
- I. Markovsky, A. Kukush, and S. Van Huffel. Consistent least squares fitting of ellipsoids. *Numerische Mathematik*, 98(1):177–194, 2004.
- National Centers for Environmental Information. <https://www.ngdc.noaa.gov/geomag/geomag.shtml>, 2015. Accessed on October 2.
- J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research, 2nd edition, 2006.
- V. Renaudin, M. H. Afzal, and G. Lachapelle. Complete triaxis magnetometer calibration in the magnetic domain. *Journal of Sensors*, 2010. Article ID 967245.
- S. Salehi, N. Mostofi, and G. Bleser. A practical in-field magnetometer calibration method for IMUs. In *Proceedings of the IROS Workshop on Cognitive Assistive Systems: Closing the Action-Perception Loop*, pages 39–44, Vila Moura, Portugal, October 2012.

- T. B. Schön, A. Wills, and B. Ninness. System identification of nonlinear state-space models. *Automatica*, 47(1):39–49, 2011.
- M. Segal and E. Weinstein. A new method for evaluating the log-likelihood gradient, the Hessian, and the Fisher information matrix for linear dynamic systems. *IEEE Transactions on Information Theory*, 35(3):682–687, 1989.
- M. D. Shuster and S. D. Oh. Three-axis attitude determination from vector observations. *Journal of Guidance, Control, and Dynamics*, 4(1):70–77, 1981.
- Trivisio Prototyping GmbH. <http://www.trivisio.com>, 2016. Accessed on January 7.
- G. Troni and L. L. Whitcomb. Adaptive estimation of measurement bias in three-dimensional field sensors with angular-rate sensors: Theory and comparative experimental evaluation. In *Proceedings of Robotics: Science and Systems (RSS)*, Berlin, Germany, June 2013.
- J. F. Vasconcelos, G. Elkaim, C. Silvestre, P. Oliveira, and B. Cardeira. Geometric approach to strapdown magnetometer calibration in sensor frame. *IEEE Transactions on Aerospace and Electronic systems*, 47(2):1293–1306, April 2011.
- Y. Wu and W. Shi. On calibration of three-axis magnetometer. *IEEE Sensors Journal*, 15(11), November 2015.
- Xsens Technologies B. V. <http://www.xsens.com>, 2016. Accessed on January 7.

Paper F

Modeling and interpolation of the ambient magnetic field by Gaussian processes

Authors: Arno Solin, Manon Kok, Niklas Wahlström, Thomas B. Schön and Simo Särkkä

Edited version of the paper:

A. Solin, M. Kok, N. Wahlström, T. B. Schön, and S. Särkkä. Modeling and interpolation of the ambient magnetic field by Gaussian processes. *ArXiv e-prints*, September 2015. arXiv:1509.04634.

Modeling and interpolation of the ambient magnetic field by Gaussian processes

Arno Solin^{*,†}, Manon Kok[‡], Niklas Wahlström^{**,†}, Thomas B. Schön^{**,†} and Simo Särkkä^{††}

^{*}IndoorAtlas Ltd.
Helsinki, Finland
arno.solin@indooratlas.com

[†]Dept. of Computer Science
Aalto University
02150 Espoo, Finland

[‡]Dept. of Electrical Engineering,
Linköping University,
SE-581 83 Linköping, Sweden
manko@isy.liu.se

^{**}Dept. of Information Technology
Uppsala University,
SE-751 05 Uppsala, Sweden
niklas.wahlstrom@it.uu.se
thomas.schon@it.uu.se

^{††}Dept. of Electrical Engineering and Automation
Aalto University
02150 Espoo, Finland
simo.sarkka@aalto.fi

Abstract

Anomalies in the ambient magnetic field can be used as features in indoor positioning and navigation. By using Maxwell's equations, we derive and present a Bayesian non-parametric probabilistic modeling approach for interpolation and extrapolation of the magnetic field. We model the magnetic field components jointly by imposing a Gaussian process (GP) prior on the latent scalar potential of the magnetic field. By rewriting the GP model in terms of a Hilbert space representation, we circumvent the computational pitfalls associated with GP modeling and provide a computationally efficient and physically justified modeling tool for the ambient magnetic field. The model allows for sequential updating of the estimate and time-dependent changes in the magnetic field. The model is shown to work well in practice in different applications: we demonstrate mapping of the magnetic field both with an inexpensive Raspberry Pi powered robot and on foot using a standard smartphone.

1 Introduction

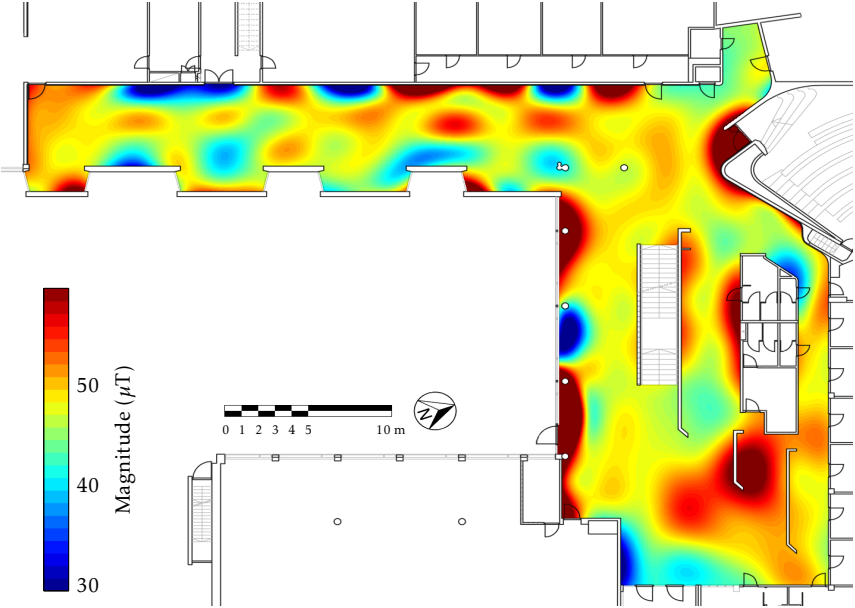
Magnetic material causes anomalies in the ambient magnetic field. In indoor environments, large amounts of such magnetic material are present in the structure of

buildings and in furniture. Our focus is on building maps of the indoor magnetic field these structures are inducing. These maps are constructed by interpolating three-dimensional magnetic field measurements obtained using magnetometers. An illustration of a map obtained using our proposed method is available in Figure 1.

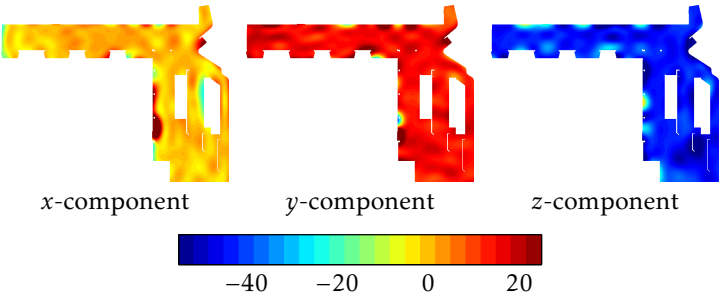
Magnetic maps of indoor environments can be used in indoor positioning and navigation applications (see, e.g. Solin et al., 2016). In these applications, measurements that are accurate on a short time-scale—but drift on longer time-horizons—are typically combined with other sources of absolute position information. Such sources can be data from wheel encoders, inertial sensors together with ultra-wideband, Wi-Fi, or optical measurement equipment such as cameras (see, e.g. Woodman, 2010; Hol, 2011). The downside of these sources of absolute position is that they typically rely on additional infrastructure or require certain conditions to be fulfilled such as line-of-sight. The advantage of using the magnetic field for positioning is that it can be measured by a small device, without additional infrastructure and without line-of-sight requirements. Furthermore, magnetometers are nowadays present in (almost) any inertial measurement unit (IMU) or smartphone. A requirement for localization using the ambient magnetic field as a source of position information is that accurate maps of the magnetic field can be constructed within reasonable computational complexity which is the focus of this work. This can also be regarded as a step towards simultaneous localization and mapping (SLAM) using magnetic fields in which localization is done while building the map (see, e.g. Leonard and Durrant-Whyte, 1991; Durrant-Whyte and Bailey, 2006).

We interpolate the magnetic field using a Bayesian non-parametric approach where prior knowledge about the properties of magnetic fields is incorporated in a Gaussian process (GP) prior. GPs (see, e.g. O’Hagan, 1978; Rasmussen and Williams, 2006) are powerful tools for Bayesian non-parametric inference and learning, and they provide a framework for fusing first-principles prior knowledge with quantities of noisy data. This has made them popular tools in signal processing, machine learning, robotics and control (Cressie, 1993; Cressie and Wikle, 2011; Deisenroth et al., 2015).

The contributions of this paper are three-fold. First, we model the ambient magnetic field using a Gaussian process prior in which we incorporate *physical knowledge* about the magnetic field. This extends the work by Wahlström et al. (2013) by presenting an approach where the GP prior is a *latent (unobservable) magnetic potential* function. Second, we use a computationally efficient GP implementation that allows us to use the large amounts of data provided by the magnetometer. To circumvent the well-known computational challenges with GPs (see, e.g. Rasmussen and Williams, 2006), we rewrite the model in terms of a Hilbert space representation introduced by Solin and Särkkä (2014). We extend the approach to allow for modeling of the bias caused by the Earth magnetic field. Third, we use this method in combination with the sequential approach introduced in Särkkä et al. (2013). This allows for *online updating* of the magnetic field estimate. It also opens up the possibility to focus on the spatio-temporal problem in which the magnetic field can change over time, for instance due to



(a) Interpolated magnetic field strength.



(b) Vector field components.

Figure 1: Interpolated magnetic field of the lobby of a building at the Aalto University campus.

furniture being moved around. An extensive evaluation of the proposed method is done using both simulated and empirical data. The simulation study and a small-scale experiment illustrate the feasibility and accuracy of the approach and allow for comparison with other methods. Experiments with a mobile robot and with a hand-held smartphone show the applicability to real-world scenarios.

This paper is structured as follows. The next section covers a survey of existing work, which also provides additional motivation for the approach. Section 3 provides a brief background of the properties of magnetic fields relevant to this work. The Gaussian process regression model is constructed in Section 4, which is then extended to explicit algorithms for batch and sequential estimation in the next section. Section 6 covers the experiments. The experimental results and some additional comments regarding the methodology are discussed at the end of the paper.

2 Related work

Spatial properties of the magnetic field have been of interest in a large variety of research domains. For instance, the magnetic field has been extensively studied in geology (see, e.g. Nabighian et al., 2005) but also in magnetospheric physics, geophysics, and astrophysics. In all of these domains, interpolation of the magnetic field is of interest (see, e.g. Guillen et al., 2008; Calcagno et al., 2008; Mackay et al., 2006; Bhattacharyya, 1969; Springel, 2010, for examples of magnetic field interpolation in the respective areas).

In recent years interest has emerged in using the magnetic field as a source of position information for indoor positioning (Haverinen and Kemppainen, 2009). Feasibility studies have been conducted, focusing both on the time-varying nature of the magnetic field and on the amount of spatial variation in the magnetic field. Li et al. (2012) report experiments showing that the magnetic field in a building typically shows large spatial variations and small time variations. This is also supported by the experimental study reported by Angermann et al. (2012) in which significant anomalies of the ambient magnetic field are reported. These experiments give confidence that the magnetic field provides sufficient information for localization purposes. However, Li et al. (2012) also report significant temporal changes in the magnetic field in the vicinity of mobile magnetic structures, in their case an elevator.

A number of approaches have been reported on building a map of the ambient magnetic field for indoor localization purposes. Le Grand and Thrun (2012) propose a method to build a map of the magnetic field by collecting magnetometer data in a grid and linearly interpolating between these points. This map is subsequently used for localization with a particle filter combining magnetometer and accelerometer measurements from a smartphone. Robertson et al. (2013) present a SLAM approach for pedestrian localization using a foot-mounted IMU. They use the magnetic field intensity which they model using spatial binning. Frassl et al. (2013) discuss the possibility of using more components of the magnetic field (for instance the full three-dimensional measurement vector) in the SLAM approach

instead. Vallivaara et al. (2010, 2011) present a SLAM approach for robot localization. They model the ambient magnetic field using a squared exponential GP prior for each of the magnetic field components. Wahlström et al. (2013) incorporate additional physical knowledge by making use of Maxwell's equations resulting in the use of curl- and divergence-free GP priors instead.

As can be concluded, there exists a wide range of existing literature when it comes to modeling the ambient magnetic field. The amount of information that is used differs between the approaches. For instance, some approaches use full three-dimensional magnetic field vectors while others only use a one-dimensional magnetic field intensity. Furthermore, the amount of physical information that is included differs. In this paper, we build on the approach by Wahlström et al. (2013) and use the full three-dimensional magnetometer measurements. We include physical knowledge in terms of the magnetic field potential.

As discussed above, GPs have frequently been used in modeling and interpolation of magnetic fields. GP regression has also successfully been applied to a wide range of applications (see, e.g. O'Callaghan and Ramos, 2012; Smith et al., 2011). Furthermore, it has previously been used for SLAM (see, e.g. Tong et al., 2013; Ferris et al., 2006; Barkby et al., 2012; Barfoot et al., 2014; Anderson et al., 2015). One of the challenges in using GPs is the computational complexity (see, e.g. Rasmussen and Williams, 2006), which scales cubically with the number of training data points. Considering the high sampling rate of the magnetometer and the fact that each observation contains three values, a large number of measurements is typically available for mapping. Because of these computational challenges, the data in Wahlström et al. (2013) was downsampled.

Attempts to speed up GP inference have spawned a wide range of methods which aimed at bringing GP regression to data-intensive application fields. These methods (see Quiñonero-Candela and Rasmussen, 2005, for a review) typically build upon reducing the rank of the Gram (covariance) matrix and using the matrix inversion lemma to speed up matrix inversion. For stationary covariance functions, the spectral Monte Carlo approximation by Lázaro-Gredilla et al. (2010), or the Laplace operator eigenbasis based method introduced by Solin and Särkkä (2014) can be employed. For uniformly spaced observations, fast Fourier transforms can provide computational benefits (Paciorek, 2007; Fritz et al., 2009). As will be shown later on in this paper, the Laplace operator approach by Solin and Särkkä (2014) falls natural to modeling of the magnetic field in terms of a magnetic field potential.

All approaches on mapping of magnetic fields discussed above assume that the magnetic field is constant over time. However, as shown by Li et al. (2012) significant temporal changes in the magnetic field occur in the vicinity of mobile magnetic structures. For GP models evolving in time, spatio-temporal GP models (see, e.g. Cressie and Wikle, 2011) can be solved efficiently using Kalman filtering methods (Hartikainen and Särkkä, 2010; Osborne, 2010; Särkkä et al., 2013; Huber, 2014). In this paper, we will take the approach of Särkkä et al. (2013) to compose a spatio-temporal GP prior for the model and solve the inference problem by a sequential Kalman filtering setup. This allows for online estimation of

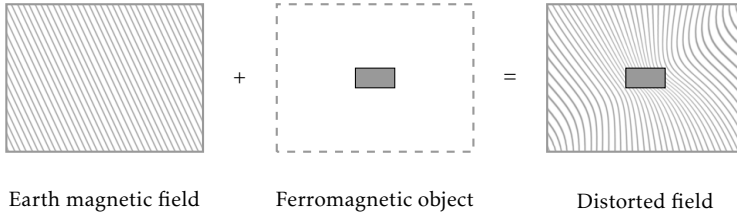


Figure 2: A ferromagnetic object deflects the Earth’s magnetic field and introduces distortions in the field.

the magnetic field estimate and can be used to allow for time variations in the magnetic field.

3 The ambient magnetic field

On a macroscopic scale, magnetic fields are *vector fields*, meaning that at any given location, they have a direction and strength (magnitude). These properties are familiar from everyday life: the force created by permanent magnets attracting and repelling ferromagnetic materials is used in various utensils, and the compass aligning itself with the direction of the Earth’s magnetic field has proved invaluable for mankind during the past centuries. The Earth’s magnetic field sets a background for the ambient magnetic field, but deviations caused by the bedrock and anomalies induced by man-built structures deflect the Earth’s magnetic field. This makes the magnetic field vary from point to point, see Figure 2.

We describe the magnetic field with a function $\mathbf{H}(\mathbf{x})$, where $\mathbf{H} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$. For each point in space $\mathbf{x}$, there will be an associated magnetic field $\mathbf{H}(\mathbf{x})$. Such a vector field can be visualized with field lines, where points in space are associated with arrows. The principles under which the magnetic field is affected by structures of buildings are well known and governed by the very basic laws of physics (see, e.g. Jackson, 1999; Vanderlinde, 2004; Griffiths, 1999).

In this work, we make use of the fact that the magnetic field $\mathbf{H}$ is curl-free

$$\nabla \times \mathbf{H} = \mathbf{0} \quad (1)$$

provided that there is no free current (current in wires for example) in the region of interest (see Wahlström, 2015, for more details). This assumption is valid in most indoor environments where the major source for variations in the ambient field is caused by metallic structures rather than free currents in wires.

One property of curl-free vector fields is that the line integral along a path P only depends on its starting point A and end point B , and not on the route taken

$$\int_P \mathbf{H}(\mathbf{x}) \cdot d\mathbf{x} = \varphi(A) - \varphi(B), \quad (2)$$

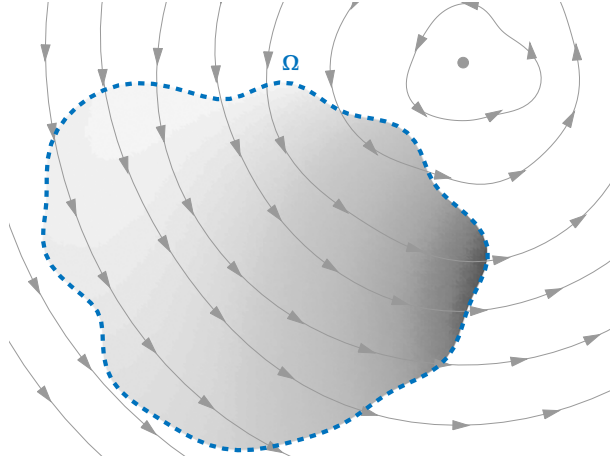


Figure 3: Illustration of a vector field with non-zero curl. The vortex point makes it non-curl-free as the vector field curls around it. However, the subset Ω excludes the vortex point and the vector field is curl-free in this region. To this region a scalar potential φ can be associated, here illustrated with shading.

where $\varphi : \mathbb{R}^3 \rightarrow \mathbb{R}$. This can be rewritten by interpreting φ as a scalar potential

$$\mathbf{H} = -\nabla\varphi. \quad (3)$$

Figure 3 illustrates the curl-free property and the scalar potential. Domain Ω is curl-free and has an associated scalar potential, while the entire domain is not curl-free due to the vortex point. Note that in a non-curl-free vector field no such scalar potential exists since a line integral around the swirl is non-zero. For the magnetic field $\mathbf{H}$, the swirl corresponds to a wire of free current pointing perpendicular to the plane, which we assume is not included in the region of interest.

The relation (3) is the key equation that we will exploit in our probabilistic model of the ambient magnetic field. We will choose to model the scalar potential φ instead of the magnetic field $\mathbf{H}$ directly. This implicitly imposes the constraints on the magnetic field that the physics is providing. This model will be explained in the next section.

4 Modeling the magnetic field using Gaussian process priors

In this section, we introduce our approach to modeling and interpolation of the ambient magnetic field. We use a Bayesian non-parametric model in which we use knowledge about the physical properties of the magnetic field as prior information. We tackle the problem of interpolating the magnetic field using GP regression. In Section 4.1 we first give a brief background on GPs. After this, we

introduce the problem of modeling the magnetic field in Section 4.2. A commonly used GP model of the magnetic field is introduced in Section 4.3. In Section 4.4, we subsequently introduce our proposed method for modeling the magnetic field in which we encode the physical properties that were presented in the previous section.

4.1 Gaussian process regression

In GP regression (Rasmussen and Williams, 2006) the model functions $f(\mathbf{x})$ are assumed to be realizations from a Gaussian random process prior with a given covariance function $\kappa(\mathbf{x}, \mathbf{x}')$. Learning amounts to computing the posterior process at some test inputs $\mathbf{x}_*$ given a set of noisy measurements $y_1, y_2, \dots, y_n$ observed at $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n$, respectively. This model is often written in the form

$$\begin{aligned} f(\mathbf{x}) &\sim \mathcal{GP}(0, \kappa(\mathbf{x}, \mathbf{x}')), \\ y_i &= f(\mathbf{x}_i) + \varepsilon_i, \end{aligned} \quad (4)$$

where the observations y_i are corrupted by Gaussian noise $\varepsilon_i \sim \mathcal{N}(0, \sigma_{\text{noise}}^2)$, for $i = 1, 2, \dots, n$. Because both the prior and the measurement model are Gaussian, the posterior process will also be Gaussian. Hence, the learning problem amounts to computing the conditional means and covariances of the process evaluated at the test inputs.

Prediction of yet unseen process outputs at an input location $\mathbf{x}_*$ amounts to the following in GP regression: $p(f(\mathbf{x}_*) \mid \mathcal{D}) = \mathcal{N}(f(\mathbf{x}_*) \mid \mathbb{E}[f(\mathbf{x}_*)], \mathbb{V}[f(\mathbf{x}_*)])$. The conditional mean and variance can be computed in closed-form as (see Rasmussen and Williams, 2006, here the conditioning is left out in the notation for brevity)

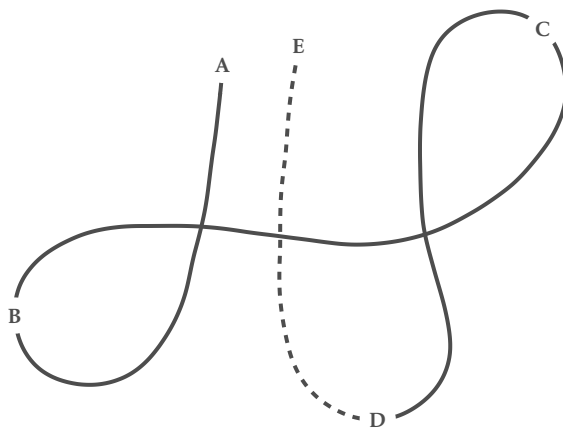
$$\begin{aligned} \mathbb{E}[f(\mathbf{x}_*)] &= \mathbf{k}_*^T (\mathbf{K} + \sigma_{\text{noise}}^2 \mathbf{I}_n)^{-1} \mathbf{y}, \\ \mathbb{V}[f(\mathbf{x}_*)] &= \kappa(\mathbf{x}_*, \mathbf{x}_*) - \mathbf{k}_*^T (\mathbf{K} + \sigma_{\text{noise}}^2 \mathbf{I}_n)^{-1} \mathbf{k}_*, \end{aligned} \quad (5)$$

where $\mathbf{K}_{i,j} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$, $\mathbf{k}_*$ is an n -dimensional vector with the i th entry being $\kappa(\mathbf{x}_*, \mathbf{x}_i)$, and $\mathbf{y}$ is a vector of the n observations. Furthermore, due to Gaussianity, the marginal likelihood (evidence) of the covariance function and noise parameters can also easily be computed, allowing for Bayesian inference of the parameters as well (Rasmussen and Williams, 2006).

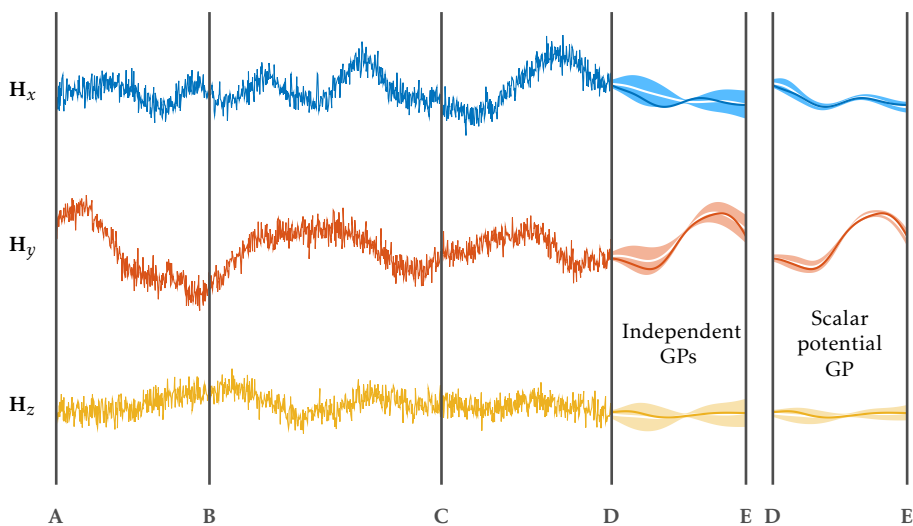
The choice of a specific covariance function encodes the *a priori* knowledge about the underlying process. One of the most commonly used covariance functions, which will also frequently be used in the next sections, is the stationary and isotropic squared exponential (also known as exponentiated quadratic, radial basis function, or Gaussian). Following the standard notation from Rasmussen and Williams (2006) it is parametrized as

$$\kappa_{\text{SE}}(\mathbf{x}, \mathbf{x}') = \sigma_{\text{SE}}^2 \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2 \ell_{\text{SE}}^2}\right), \quad (6)$$

where the hyperparameters σ_{SE}^2 and ℓ_{SE} represent the magnitude scale and the characteristic length-scale, respectively. These can be learned from data, for instance by maximizing the marginal likelihood.



(a) The route.



(b) The data and the GP prediction for D–E.

Figure 4: A simulated example of the interpolation problem. (a) Training data has been collected along the route A–D, but the magnetic field between D–E is unknown. (b) The noisy observations of the magnetic field between A–D, and GP predictions with 95% credibility intervals. Both the independent GP modeling approach (with shared hyperparameters) and the scalar potential based curl-free GP approach are visualized. The simulated ground truth is shown by the solid lines.

4.2 Interpolation of magnetic fields

In this work we tackle the problem of interpolating the magnetic field to spatial locations from where we do not have any measurements. In other words, we will tackle the problem of predicting the latent (unobservable noise-free) magnetic field $\mathbf{f}(\mathbf{x}_*)$ (such that $\mathbf{f} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$) at some arbitrary location $\mathbf{x}_*$ given a set of noise-corrupted measurements $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ of the magnetic field. Here, the measurements $\mathbf{y}$ correspond to the H-field corrupted by i.i.d. Gaussian noise.

Two important things need to be noted with regard to the interpolation of the magnetic field. First, note that the measurements of the magnetic field are *vector-valued* (contrary to the scalar observations in (4)). This raises the question of how to deal with the different magnetic field components. They can either be treated separately as will be done in Section 4.3, or a relation between the different components can be assumed, as is the case in the method we propose in Section 4.4. Secondly, note that the function describing the magnetic field is not zero-mean, contrary to the GP model in (4). Instead, its mean lies around a local Earth magnetic field. This depends on the location on the Earth but can also deviate from the Earth's magnetic field in indoor environments due to magnetic material in the structure of the building. The unknown mean can be modeled as an additional part of the covariance function $\kappa(\mathbf{x}, \mathbf{x}')$ (Rasmussen and Williams, 2006).

An illustration of GP regression for magnetic fields is provided in Figure 4, where noisy readings of a magnetic field have been collected along route A–D (comprising $\mathcal{D}$), and the magnetic field along route D–E (comprising the prediction locations $\mathbf{x}_*$) needs to be inferred from the measurements. Each component varies around a local magnetic field due to magnetic material in the vicinity of the sensor. Different interpolation techniques can be used based on different prior knowledge that can be incorporated in the GP. Two different interpolation results are shown: one based on independent modeling of each vector field components (with shared hyperparameters) and another based on associating the GP prior with the scalar potential of the (curl-free) vector field. These are based on the models we will introduce in the coming two sections.

4.3 Separate modeling of the magnetic field components

The most straightforward approach to GP modeling of vector-valued quantities is to model each of the field components as an independent GP. This approach has been widely applied in existing literature (see, e.g. Vallivaara et al., 2010, 2011; Kemppainen et al., 2011; Jung et al., 2015; Viseras Ruiz and Olariu, 2015). For each of the three magnetic field components $d \in \{1, 2, 3\}$, this model can be written as

$$\begin{aligned} f_d(\mathbf{x}) &\sim \mathcal{GP}(0, \kappa_{\text{const.}}(\mathbf{x}, \mathbf{x}') + \kappa_{\text{SE}}(\mathbf{x}, \mathbf{x}')), \\ y_{d,i} &= f_d(\mathbf{x}_i) + \varepsilon_{i,d}, \end{aligned} \tag{7}$$

where the observations $y_{d,i}$, $i = 1, 2, \dots, n$, are corrupted by independent Gaussian noise with variance σ_{noise}^2 . The non-zero mean of the magnetic field is handled by

a constant covariance function

$$\kappa_{\text{const.}}(\mathbf{x}, \mathbf{x}') = \sigma_{\text{const.}}^2, \quad (8)$$

where $\sigma_{\text{const.}}^2$ is a magnitude scale hyperparameter. The small-scale variation in the field is modeled by a squared exponential covariance function (6). Hence, the model (7) encodes the knowledge that the realizations are expected to be smooth functions in space with a constant shift from zero mean.

The model has four hyperparameters: two magnitude scale parameters ($\sigma_{\text{const.}}^2$ and σ_{SE}^2), a length-scale parameter (ℓ_{SE}), and a noise scale parameter (σ_{noise}^2). Assuming that the components are completely separate, each component has four hyperparameters to learn. The resulting model is flexible, as it does not encode any relation between the vector field components. In practice, this might lead to problems in hyperparameter estimation, with parameter estimates converging to local optima and magnetic field components behaving very differently with respect to each other. Therefore, the hyperparameters are often fixed to reasonable values—instead of learned from data.

A more sensible approach for separate, but not completely independent, modeling of the magnetic field measurements, models them as realizations of three independent GP priors with joint learning of the shared hyperparameters (see also Kempainen et al., 2011). Note that for this model, the covariance in the GP posterior is independent of the outputs $\mathbf{y}$ and only depends on the input locations $\mathbf{x}$ (which are shared for all components in $\mathbf{y}$). Hence, calculating the marginal likelihood only requires inverting a matrix of size n (not $3n$). For this model, the expression for evaluating the log marginal likelihood function for hyperparameter optimization can be written as

$$\begin{aligned} \mathcal{L}(\theta) = -\log p(\mathbf{y} \mid \theta, \mathcal{D}) = & \frac{3}{2} \log |\mathbf{K}_{\theta} + \sigma_{\text{noise}}^2 \mathbf{I}_n| \\ & + \frac{1}{2} \text{tr}[\mathbf{y}(\mathbf{K}_{\theta} + \sigma_{\text{noise}}^2 \mathbf{I}_n)^{-1} \mathbf{y}^{\top}] + \frac{3n}{2} \log(2\pi), \quad (9) \end{aligned}$$

where $\mathbf{y} \in \mathbb{R}^{3 \times n}$ and $\mathbf{K}_{\theta} \in \mathbb{R}^{n \times n}$.

Figure 4b shows the results of predicting the magnetic field behavior along the route D–E for the GP prior modeling the three magnetic field components separately but with joint learning of the shared hyperparameters. The colored patches show the 95% credibility intervals for the prediction with the mean estimate visualized by the white line. The simulated ground-truth (solid colored line) falls within the shown interval, and the model captures the general shape of the magnetic field variation along the path. The strengths of this model are that it is flexible and that the assumptions are conservative. The weaknesses on the other hand are evident: The model does not incorporate physical knowledge of the magnetic field characteristics. In the next section we will instead explore this knowledge by modeling the magnetic field as derivative measurements of a scalar potential.

4.4 Modeling the magnetic field as the gradient of a scalar potential

Following our choices in Section 3, we assume that the magnetic field $\mathbf{H}$ can be written as the gradient of a scalar potential $\varphi(\mathbf{x})$ according to (3). Here, $\varphi : \mathbb{R}^3 \rightarrow \mathbb{R}$ and $\mathbf{x} \in \mathbb{R}^3$ is the spatial coordinate. We assume $\varphi(\mathbf{x})$ to be a realization of a GP prior and the magnetic field measurements $\mathbf{y}_i \in \mathbb{R}^3$ to be its gradients corrupted by Gaussian noise. This leads to the following model

$$\begin{aligned}\varphi(\mathbf{x}) &\sim \mathcal{GP}(0, \kappa_{\text{lin.}}(\mathbf{x}, \mathbf{x}') + \kappa_{\text{SE}}(\mathbf{x}, \mathbf{x}')), \\ \mathbf{y}_i &= -\nabla \varphi(\mathbf{x}) \Big|_{\mathbf{x}=\mathbf{x}_i} + \varepsilon_i,\end{aligned}\tag{10}$$

where $\varepsilon_i \sim \mathcal{N}(\mathbf{0}, \sigma_{\text{noise}}^2 \mathbf{I}_3)$, for each observation $i = 1, 2, \dots, n$. Recall that nabla (∇) is a linear operator, and that Gaussianity is preserved under linear operations. Thus, we can still derive a closed-form solution for this GP model. For simplifying the notation in the next sections, we introduce the notation $\mathbf{f}(\mathbf{x})$ for the gradient field evaluated at $\mathbf{x}$.

The squared exponential covariance function (6) in (10) allows us to model the magnetic field anomalies induced by small-scale variations and building structures. The local Earth's magnetic field contributes linearly to the scalar potential as

$$\kappa_{\text{lin.}}(\mathbf{x}, \mathbf{x}') = \sigma_{\text{lin.}}^2 \mathbf{x}^\top \mathbf{x},\tag{11}$$

where $\sigma_{\text{lin.}}^2$ is the magnitude scale hyperparameter. The derivative of this linear component contributes to the model in a similar way as the constant covariance function (8) in the previous section.

Our model now has four hyperparameters: two magnitude scale parameters ($\sigma_{\text{lin.}}^2$ and σ_{SE}^2), a length-scale parameter (ℓ_{SE}), and a noise scale parameter (σ_{noise}^2). We learn these parameters from the data, by maximizing the marginal likelihood.

The second set of predictions for the route D–E in Figure 4b shows the interpolation outcome from using the above model. In comparison to the independent GP model, the scalar potential based GP prior provides additional information to the model by tying the vector field components to each other. This improves the estimates in terms of accuracy and makes the 95% credibility interval more narrow.

Incorporating physical knowledge about the magnetic field into a GP prior is not new. In fact, the assumptions in our model (10) are similar to the assumptions made by Wahlström et al. (2013) where the magnetic field is modeled as curl-free using a curl-free kernel (see, e.g. Fuselier, 2007; Baldassarre et al., 2010; Álvarez et al., 2012). The equivalence of the two models is shown in Wahlström (2015). In this work, however, the model formulation through the scalar potential is crucial, as it will enable us to easily extend the model to an approximate form for efficient GP inference in the next section.

5 Efficient GP modeling of the magnetic field

Gaussian processes are convenient tools for assigning flexible priors to data—as we saw in the previous section. However, the main problem with the model in the previous section is its high computational cost. The approach scales as $\mathcal{O}(n^3)$ (recall that each observation is three-dimensional, meaning $3n$ becomes large very quickly). This computational complexity renders the approach more or less useless in practice, when the number of observations becomes large (say $n > 1000$).

This is a fundamental restriction associated with the naive formulation of GP models involving the inversion of the covariance matrix. Using special structure of the problem and/or approximative methods, this high computational cost can often be circumvented. In this section we present an approach which both uses the special differential operator structure and projects the model on a set of basis functions characteristic to the covariance function. We first present the method for spatial batch estimation, and then extend it to a temporal dimension as well.

Existing GP methods for mapping and interpolation of the magnetic field have been considering only batch estimation, where the data is first acquired and then processed as a batch. In this section, we aim to extend this to an online method, enabling the GP regression estimate of the magnetic field to be updated when new data is acquired. We denote such a data set as $\mathcal{D}_n = \{(\mathbf{x}_i, \mathbf{y}_i) \mid i = 1, 2, \dots, n\}$, and thus $\mathcal{D}_i$ denotes all the data that has been observed up to time instance t_i .

Considering time as part of the data stream enables us to think of three distinctive setups for estimation of the magnetic field:

- **Batch estimation** of the magnetic field, where the data is first acquired and then the field is estimated at once.
- **Sequential updating** of the field estimate, where we assume all the measurements to be of the same static magnetic field.
- **Spatio-temporal estimation** of the time-dependent magnetic field, where we assume the field to change over time.

In the next sections we will present how these scenarios can be combined with the scalar potential based GP scheme without requiring to repeat the batch computations after each sample.

5.1 Reduced-rank GP modeling

A recent paper by Solin and Särkkä (2014) presents an approach that is based on a series expansion of stationary covariance functions. The approximation is based on the following truncated series:

$$\kappa(\mathbf{x}, \mathbf{x}') \approx \sum_{j=1}^m S(\lambda_j) \phi_j(\mathbf{x}) \phi_j(\mathbf{x}'), \quad (12)$$

where $S(\cdot)$ is the spectral density of the covariance function $\kappa(\cdot, \cdot)$, $\phi_j(\mathbf{x})$ is the j th eigenfunction of the negative Laplace operator and λ_j^2 is the corresponding eigenvalue. The efficiency of this approach is based on two properties: (i) the eigenfunctions are independent of the hyperparameters of the covariance function, and (ii) for many domains the eigenfunctions and eigenvalues can be solved beforehand in closed-form. Truncating this expansion at degree $m \ll n$ allows the GP regression problem to be solved with a $\mathcal{O}(nm^2)$ and the hyperparameters to be learned with a $\mathcal{O}(m^3)$ time complexity. The memory requirements scale as $\mathcal{O}(nm)$.

However, what makes this approach even better suited for this problem, is that the approximation is based on the eigendecomposition of the Laplace operator. This eigenbasis falls natural to the problem formulation in the previous section, where the latent potential field is observed through gradients. As for the choice of covariance structure, the squared exponential covariance function is stationary, so there is no problem applying this approach for that part of the above problem formulation. The linear covariance function is not stationary, but as we will see, that is not a problem in this case.

Our interest lies in modeling the magnetic field in compact subsets of $\mathbb{R}^3$, allowing us to restrict our interest to domains Ω comprising three-dimensional cuboids (rectangular boxes) such that $\mathbf{x} \in [-L_1, L_1] \times [-L_2, L_2] \times [-L_3, L_3] \subset \mathbb{R}^3$ (recall that a stationary covariance function is translation invariant). In this domain, we can solve the eigendecomposition of the Laplace operator subject to Dirichlet boundary conditions

$$\begin{cases} -\nabla^2 \phi_j(\mathbf{x}) = \lambda_j^2 \phi_j(\mathbf{x}), & \mathbf{x} \in \Omega, \\ \phi_j(\mathbf{x}) = 0, & \mathbf{x} \in \partial\Omega. \end{cases} \quad (13)$$

The choice of the domain and boundary conditions is arbitrary, but for regression problems with a stationary covariance function the model reverts back to the prior outside the region of observed data, so the Dirichlet boundary condition does not restrict the modeling if Ω is chosen suitably.

This particular choice of domain and boundary conditions yield the following analytic expression for the basis functions:

$$\phi_j(\mathbf{x}) = \prod_{d=1}^3 \frac{1}{\sqrt{L_d}} \sin\left(\frac{\pi n_{j,d}(x_d + L_d)}{2L_d}\right), \quad (14)$$

$$\lambda_j^2 = \sum_{d=1}^3 \left(\frac{\pi n_{j,d}}{2L_d}\right)^2, \quad (15)$$

where the matrix $\mathbf{n} \in \mathbb{R}^{m \times 3}$ consists of an index set of permutations of integers $\{1, 2, \dots, m\}$ (i.e., $\{(1, 1, 1), (1, 1, 2), \dots, (1, 2, 1), \dots, (2, 1, 1), \dots\}$). The basis functions are independent of the hyperparameters, and thus only need to be evaluated once. Adding the linear covariance function to the GP prior corresponds to inserting a set of three linear basis functions into the model.

The computational benefits come from the approximate eigendecomposition of the Gram (covariance) matrix, $\mathbf{K}_{i,j} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ (see Solin and Särkkä, 2014, for

derivations and discussion). It can now be written out in terms of the basis functions and spectral densities: $\mathbf{K} \approx \mathbf{\Phi} \mathbf{\Lambda} \mathbf{\Phi}^\top$. The basis functions which span the solution are collected in the matrix $\mathbf{\Phi} \in \mathbb{R}^{n \times (3+m)}$, with the following rows

$$\mathbf{\Phi}_i = (\mathbf{x}_i^\top, \phi_1(\mathbf{x}_i), \phi_2(\mathbf{x}_i), \dots, \phi_m(\mathbf{x}_i)), \quad (16)$$

for $i = 1, 2, \dots, n$. Accordingly, we define the corresponding measurement model matrix projecting the derivative observations onto the basis functions. Analogously, we define the matrix $\nabla \mathbf{\Phi} \in \mathbb{R}^{3n \times (3+m)}$ as the following block-row matrix:

$$\nabla \mathbf{\Phi}_i = (\nabla \mathbf{x}_i^\top, \nabla \phi_1(\mathbf{x}_i), \nabla \phi_2(\mathbf{x}_i), \dots, \nabla \phi_m(\mathbf{x}_i)), \quad (17)$$

for $i = 1, 2, \dots, n$. Similarly we define $\mathbf{\Phi}_*$ and $\nabla \mathbf{\Phi}_*$ as vectors evaluated at the prediction input location $\mathbf{x}_*$ defined analogously to Equations (16) and (17), respectively. The diagonal matrix $\mathbf{\Lambda}$ is defined by

$$\mathbf{\Lambda} = \text{diag}(\sigma_{\text{lin.}}^2, \sigma_{\text{lin.}}^2, \sigma_{\text{lin.}}^2, S_{\text{SE}}(\lambda_1), S_{\text{SE}}(\lambda_2), \dots, S_{\text{SE}}(\lambda_m)). \quad (18)$$

For three-dimensional inputs, the spectral density function of the squared exponential covariance function (6) is given by

$$S_{\text{SE}}(\omega) = \sigma_{\text{SE}}^2 (2\pi\ell_{\text{SE}}^2)^{3/2} \exp\left(-\frac{\omega^2 \ell_{\text{SE}}^2}{2}\right), \quad (19)$$

where the hyperparameters σ_{SE}^2 and ℓ_{SE} characterize the spectrum.

The Laplace operator eigenbasis approximation method can be combined with the independent GP approach, the independent GPs with shared hyperparameters, and the scalar potential GP approach. In the next sections the presentation will be specific to the scalar potential model, but a similar setup can be constructed for the other methods as well.

5.2 Batch estimation

We first tackle the batch estimation problem which provides the approximative solution to the GP regression problem in Equation (5) for the scalar potential GP.

Following the derivations of Solin and Särkkä (2014), predictions for interpolation and extrapolation of the magnetic field at yet unseen input locations $\mathbf{x}_*$ are given by:

$$\begin{aligned} \mathbb{E}[\mathbf{f}(\mathbf{x}_*)] &\approx \nabla \mathbf{\Phi}_* ([\nabla \mathbf{\Phi}]^\top \nabla \mathbf{\Phi} + \sigma_{\text{noise}}^2 \mathbf{\Lambda}^{-1})^{-1} [\nabla \mathbf{\Phi}]^\top \text{vec}(\mathbf{y}), \\ \mathbb{V}[\mathbf{f}(\mathbf{x}_*)] &\approx \sigma_{\text{noise}}^2 \nabla \mathbf{\Phi}_* ([\nabla \mathbf{\Phi}]^\top \nabla \mathbf{\Phi} + \sigma_{\text{noise}}^2 \mathbf{\Lambda}^{-1})^{-1} [\nabla \mathbf{\Phi}_*]^\top, \end{aligned} \quad (20)$$

where $\text{vec}(\cdot)$ is the vectorization operator which converts a matrix to a column vector by stacking its columns on top of each other, such that the $3 \times n$ matrix is converted into a vector of size $3n$. The basis functions $\nabla \mathbf{\Phi}$ and $\nabla \mathbf{\Phi}_*$ need to be evaluated by Equation (17), and $\mathbf{\Lambda}$ by (18).

Algorithm 1 Algorithm for batch estimation of the scalar potential GP magnetic field with the reduced-rank approach.

Input: $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n, \mathbf{x}_*, \Omega, m$.

Output: $\mathbb{E}[\mathbf{f}(\mathbf{x}_*)], \mathbb{V}[\mathbf{f}(\mathbf{x}_*)]$.

- 1: Use Eq. (17) to evaluate the basis functions $\nabla\Phi$ from $\mathbf{x}_i$ s and Ω .
 - 2: Use Eq. (21) to optimize hyperparameters $\theta = \{\sigma_{\text{lin}}^2, \sigma_{\text{SE}}^2, \ell_{\text{SE}}, \sigma_{\text{noise}}^2\}$.
 - 3: Use Eq. (17) to evaluate the basis functions $\nabla\Phi_*$ from $\mathbf{x}_*$ s and Ω .
 - 4: Solve the GP regression problem by Eq. (20).
-

For this model, the expression for evaluating the log marginal likelihood function for hyperparameter optimization can be written as

$$\mathcal{L}(\theta) = \frac{1}{2} \log |\mathbf{K}_\theta + \sigma_{\text{noise}}^2 \mathbf{I}_{3n}| + \frac{1}{2} \text{vec}(\mathbf{y})^\top (\mathbf{K}_\theta + \sigma_{\text{noise}}^2 \mathbf{I}_{3n})^{-1} \text{vec}(\mathbf{y}) + \frac{3n}{2} \log(2\pi), \quad (21)$$

where the quantities can be approximated by

$$\begin{aligned} \log |\mathbf{K}_\theta + \sigma_{\text{noise}}^2 \mathbf{I}_{3n}| &\approx (3n - m) \log \sigma_{\text{noise}}^2 \\ &+ \sum_{j=1}^m [\Lambda_\theta]_{j,j} + \log |\sigma_{\text{noise}}^2 \Lambda_\theta^{-1} + [\nabla\Phi]^\top \nabla\Phi|, \end{aligned} \quad (22)$$

$$\begin{aligned} \text{vec}(\mathbf{y})^\top (\mathbf{K}_\theta + \sigma_{\text{noise}}^2 \mathbf{I}_{3n})^{-1} \text{vec}(\mathbf{y}) &\approx \frac{1}{\sigma_{\text{noise}}^2} \left[\text{vec}(\mathbf{y})^\top \text{vec}(\mathbf{y}) - \text{vec}(\mathbf{y})^\top \nabla\Phi (\sigma_{\text{noise}}^2 \Lambda_\theta^{-1} \right. \\ &\quad \left. + [\nabla\Phi]^\top \nabla\Phi)^{-1} [\nabla\Phi]^\top \text{vec}(\mathbf{y}) \right], \end{aligned} \quad (23)$$

where the only remaining dependency on the covariance function hyperparameters are in the diagonal matrix Λ defined through the spectral density in Equation (18). In a software implementation, Cholesky decompositions can be employed for numerical stability in the calculation of determinants and matrix inverses. For optimizing the hyperparameters, gradient based optimizers can be employed (see Solin and Särkkä, 2014, for details on deriving the partial derivatives).

Algorithm 1 describes the step-by-step workflow for applying these equations in practice. The inputs for the method are the data $\mathcal{D}$ (spatial points and the magnetic field readings), the test points $\mathbf{x}_*$ to predict at, the domain boundaries Ω , and the approximation degree parameter m (controlling the accuracy of the Hilbert space approximation, see Eq. (12)). The algorithm returns the marginal mean and variance of the predicted magnetic field at $\mathbf{x}_*$. The scalar potential could be returned instead by using Equation (16) in step three.

5.3 Sequential estimation

Many applications require online (sequential) estimates of the magnetic field. The following formulation provides the *same* (within numerical precision) solution as the batch estimation solution (20) in the previous section. The inference scheme in the previous section is in practice the solution of a linear Gaussian estimation problem. Sequential solutions for this type of problems have been extensively studied, and this mathematical formulation is widely known as the *Kalman filter*. The connection between Kalman filtering and Gaussian process regression has recently been studied, for example, in Osborne (2010); Särkkä et al. (2013); Huber (2014).

Reformulation of a batch problem to a sequential algorithm is discussed (with examples) in the book Särkkä (2013). Following this formulation (and notation to large extent) we may write the following recursion: Initialize $\boldsymbol{\mu}_0 = \mathbf{0}$ and $\boldsymbol{\Sigma}_0 = \boldsymbol{\Lambda}_\theta$ (from the GP prior). For each new observation $i = 1, 2, \dots, n$ update the estimate according to

$$\begin{aligned} \mathbf{S}_i &= \nabla \boldsymbol{\Phi}_i \boldsymbol{\Sigma}_{i-1} [\nabla \boldsymbol{\Phi}_i]^\top + \sigma_{\text{noise}}^2 \mathbf{I}_3, \\ \mathbf{K}_i &= \boldsymbol{\Sigma}_{i-1} [\nabla \boldsymbol{\Phi}_i]^\top \mathbf{S}_i^{-1}, \\ \boldsymbol{\mu}_i &= \boldsymbol{\mu}_{i-1} + \mathbf{K}_i (\mathbf{y}_i - \nabla \boldsymbol{\Phi}_i \boldsymbol{\mu}_{i-1}), \\ \boldsymbol{\Sigma}_i &= \boldsymbol{\Sigma}_{i-1} - \mathbf{K}_i \mathbf{S}_i \mathbf{K}_i^\top. \end{aligned} \tag{24}$$

This means that for a test input location $\mathbf{x}_*$ we get predictions for the mean and the variance of the magnetic field which are given by

$$\begin{aligned} \mathbb{E}[\mathbf{f}(\mathbf{x}_*) \mid \mathcal{D}_i] &\approx \nabla \boldsymbol{\Phi}_* \boldsymbol{\mu}_i, \\ \mathbb{V}[\mathbf{f}(\mathbf{x}_*) \mid \mathcal{D}_i] &\approx \nabla \boldsymbol{\Phi}_* \boldsymbol{\Sigma}_i [\nabla \boldsymbol{\Phi}_*]^\top, \end{aligned} \tag{25}$$

and conditional on the data observed up to observation i . Writing out the conditioning on $\mathcal{D}$ was stripped in the earlier sections for brevity.

Here we do not consider optimization of the hyperparameters θ . The marginal likelihood can be evaluated through the recursion, but in an online setting we suggest optimizing the hyperparameters with some initial batch early in the data collection and then re-optimizing them later on if necessary.

Algorithm 2 presents the scheme of how to apply the equations in practice. The inputs are virtually the same as for the batch algorithm, but now the hyperparameters θ are considered known. The current estimate can be returned after each iteration loop.

5.4 Spatio-temporal modeling

The sequential model allows for extending the modeling to also track dynamic changes in the magnetic field without virtually any additional computational burden. Let the data $\mathcal{D}_n = \{(t_i, \mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ now also comprise a temporal variable t which indicates the time when each observation was acquired.

Algorithm 2 Algorithm for sequential modeling of the scalar potential GP magnetic field estimate. Alternative (a) corresponds to the sequential model, and (b) to the spatio-temporal modeling approach.

Input: $\mathcal{D}_n, \mathbf{x}_*, \Omega, m, \theta$.

Output: Bar.

- 1: Initialize $\boldsymbol{\mu}_0 = \mathbf{0}$ and $\Sigma_0 = \mathbf{\Lambda}_\theta$ from Eq. (18).
 - 2: **for** $i = 1, 2, \dots, n$ **do**
 - 3: Evaluate $\nabla \Phi_i$ by Eq. (17) from $\mathbf{x}_i$.
 - 4: **(a)** Perform an update by Eq. (24).
 - 5: **(b)** Perform an update by Eqs. (29–30).
 - 6: Evaluate the current prediction at $\mathbf{x}_*$ by Eq. (25).
 - 7: **end for**
-

We present the following spatio-temporal model for tracking changes in the ambient magnetic field. The spatio-temporal GP prior assigned to the scalar potential $\varphi(\mathbf{x}, t)$, depending on both location and time, is defined as follows:

$$\varphi(\mathbf{x}, t) \sim \mathcal{GP}(0, \kappa_{\text{lin.}}(\mathbf{x}, \mathbf{x}') + \kappa_{\text{SE}}(\mathbf{x}, \mathbf{x}')\kappa_{\text{exp}}(t, t')), \quad (26)$$

where the additional covariance function $\kappa_{\text{exp}}(t, t')$ defines the prior assumptions of the temporal behavior. This covariance function is defined through

$$\kappa_{\text{exp}}(t, t') = \exp\left(-\frac{|t - t'|}{\ell_{\text{time}}}\right), \quad (27)$$

where ℓ_{time} is a hyperparameter controlling the length-scale of the temporal effects. In the temporal domain, this model is also known as the Ornstein–Uhlenbeck process (see, e.g. Rasmussen and Williams, 2006). The assumption encoded into it is that the phenomenon is continuous but not necessarily differentiable. Therefore it provides a very flexible means of modeling the changing ambient magnetic field. Also note that in Equation (26) the temporal effects are only associated with the anomaly component, the bias being tracked as a static component.

Following the derivations in Hartikainen and Särkkä (2010), we can write down the dynamic state space model associated with the time evolution of the spatio-temporal GP prior model (26)

$$\begin{aligned} \mathbf{A}_i &= \text{blkdiag}(\mathbf{I}_3, \mathbf{I}_m \exp(-\Delta t_i / \ell_{\text{time}})), \\ \mathbf{Q}_i &= \text{blkdiag}(\mathbf{0}_3, \mathbf{I}_m [1 - \exp(-2\Delta t_i / \ell_{\text{time}})]), \end{aligned} \quad (28)$$

where $\Delta t_i = t_{i+1} - t_i$ is the time difference between two consecutive samples and $\mathbf{0}_3$ denotes a 3×3 zero matrix.

For the time update (Kalman prediction step) we may thus write

$$\begin{aligned} \tilde{\boldsymbol{\mu}}_i &= \mathbf{A}_{i-1} \boldsymbol{\mu}_{i-1}, \\ \tilde{\Sigma}_i &= \mathbf{A}_{i-1} \Sigma_{i-1} \mathbf{A}_{i-1}^\top + \mathbf{Q}_{i-1}, \end{aligned} \quad (29)$$

and the modified measurement update (Kalman update step)

$$\begin{aligned}
 \mathbf{S}_i &= \nabla \Phi_i \tilde{\Sigma}_i [\nabla \Phi_i]^\top + \sigma_{\text{noise}}^2 \mathbf{I}_3, \\
 \mathbf{K}_i &= \tilde{\Sigma}_i [\nabla \Phi_i]^\top \mathbf{S}_i^{-1}, \\
 \boldsymbol{\mu}_i &= \tilde{\boldsymbol{\mu}}_i + \mathbf{K}_i (\mathbf{y}_i - \nabla \Phi_i \tilde{\boldsymbol{\mu}}_i), \\
 \Sigma_i &= \tilde{\Sigma}_i - \mathbf{K}_i \mathbf{S}_i \mathbf{K}_i^\top.
 \end{aligned} \tag{30}$$

Algorithm 2 features the workflow of applying the method (option (b)). In practice the only additional input is including the temporal length-scale in θ .

6 Experiments

The experiments in this paper are split into four parts. We first demonstrate the feasibility of the scalar potential approach with simulated data, where comparison to known ground truth is possible. After this we present a small-scale proof-of-concept demonstration of the approach.

The final two examples are closer to real-world use cases. The third experiment is concerned with mapping the magnetic field in a building using a handheld smartphone. The final experiment uses an inexpensive mobile robot for online mapping and real-time tracking of the changing magnetic field.

6.1 Simulated experiment

As a first part of our experimental validation of the model, we present a simulation study. This will be used to illustrate our method and to quantify its performance through Monte Carlo simulations. In the simulations, we assume that the magnetic field measurements can indeed be modeled using a scalar potential as argued in Section 3. Hence, we simulate the magnetometer data from the GP that models the magnetometer measurements as gradients of a scalar potential field as discussed in Section 4.4. To reduce the computational complexity, we simulate the data using the computationally efficient approach described in Section 5.2 with a large number of basis functions ($m = 4096$). This reduces computational complexity and is, as shown by Solin and Särkkä (2014), a good approximation of the true model.

The magnetometer is assumed to move in a three-dimensional volume with $x, y, z \in [-0.4, 0.4]$ such that $\mathbf{x} = (x, y, z)$. The training data used for training the GP is randomly uniformly distributed over this volume. A validation data set is used to assess the predictive power of the trained GP. This validation data is a three-dimensional meshgrid over the same volume and consists of $n_{\text{val.}} = 9261$ positions $\mathbf{x}_{\text{val.}}$ with a true magnetic field $\mathbf{f}_{\text{true}}(\mathbf{x}_{\text{val.}})$. The magnetic field is predicted at these points using the trained GP, leading to $\mathbf{f}_{\text{train.}}(\mathbf{x}_{\text{val.}})$, after which the quality of the GP solution can be assessed in terms of the root mean square error (RMSE).

We set the domain Ω in the GP model to $L_1 = L_2 = L_3 = 0.5$ and simulate using the following hyperparameters $\sigma_{\text{const.}}^2 = 0.3$, $\sigma_{\text{SE}}^2 = 1$, $\ell_{\text{SE}} = 0.1$, and $\sigma_{\text{noise}}^2 = 0.04$ (see Section 5.2 and 4.4 for more details on the notation).

We compare our proposed method with two other approaches. The first is the approach by Vallivaara et al. (2010, 2011), where the magnetic field is modeled using independent GPs for all three components. Each GP consists of a constant and a squared exponential kernel and has its own hyperparameters. The second approach, considered in Kempainen et al. (2011), models the magnetic field similarly but with shared hyperparameters. For details, see also Section 4.3.

In a first set of Monte Carlo simulations, we analyze the performance of the three different approaches depending on the number of (randomly distributed) training data points, that is in terms of the sparseness of the magnetometer data and the amount of interpolation that is needed for prediction. For all three approaches we use a large number of basis functions ($m = 4096$). They are hence expected to approach the performance of the full GP solution. To exclude problems with local minima in the hyperparameter optimization—which will be the topic of a second set of simulations—the hyperparameter optimization is started in the values used for simulating the data. The results from 30 Monte Carlo simulations are shown in Figure 5a. Naturally, the more data is used for training the GP, the smaller the RMSE becomes. The GP that models the magnetic field measurements as gradients of a scalar potential field, outperforms the other two approaches, independent of the amount of training data used. This can be understood from the fact that this approach incorporates most physical knowledge.

In a second set of Monte Carlo simulations, the sensitivity to the initialization of the hyperparameter optimization is analyzed for the three different methods. Only one simulated data set is used but the hyperparameter optimization is started in 30 randomly selected sets of hyperparameters θ_0^a for a varying length of the training data. The hyperparameters are assumed to lie around the estimates that are obtained using the same optimization strategy as above for 8000 data points. Hence, these sets of hyperparameters θ_{true}^a are known to result in small RMSE values as depicted in Figure 5a. The superscript ‘a’ on θ_{true}^a is used to explicitly denote that these sets of hyperparameters actually differ between the three different approaches. For the approach where the magnetic field components are modeled using independent GPs, each component results in a set of hyperparameters θ_{true}^a . For simplicity, in this approach, θ_{true}^a is chosen to be the mean of these three sets of hyperparameters.

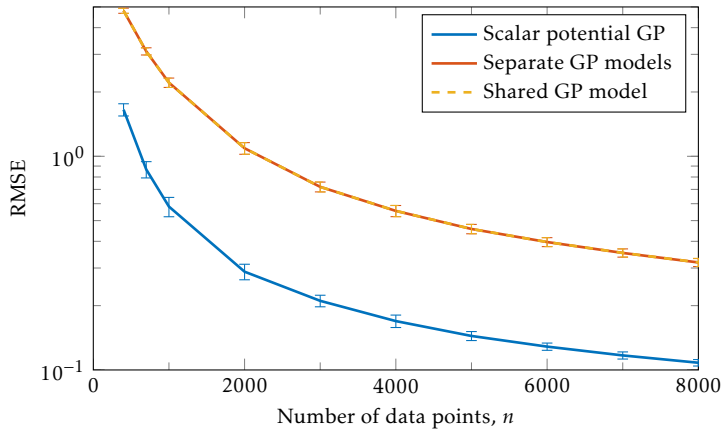
For each of the three approaches, the initial parameters θ_0 are then assumed to deviate from θ_{true}^a by at most 70% as

$$\theta_0 = \theta_{\text{true}}^a [1 + 0.7 U(-1, 1)]. \quad (31)$$

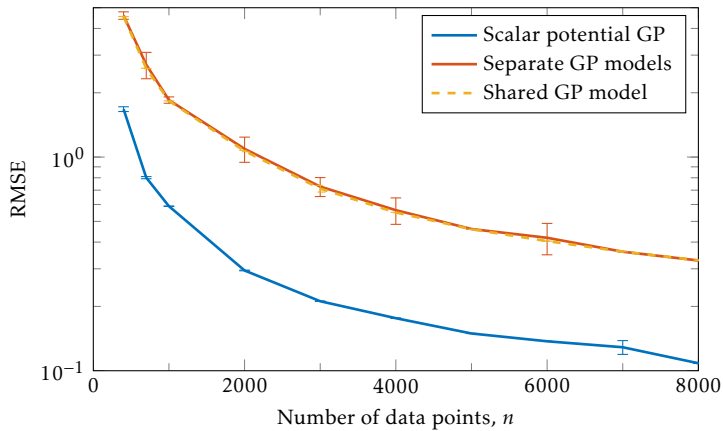
The Monte Carlo simulation results are depicted in Figure 5b. As can be seen, the approach which models the three magnetic field components using separate GPs suffers most from local minima. Our proposed model using a scalar potential still outperforms the other two.

6.2 Empirical proof-of-concept data

To illustrate our approach using real data, we have performed an experiment where a number of sensors have been moved around in a magnetic environment.



(a) Randomized over data sets.



(b) Randomized over initializations.

Figure 5: The average RMSE (with standard deviations) from 30 Monte Carlo simulations as a function of the number of simulated data points used for training the GP. The constantly lower error for the Hilbert space scalar potential GP in comparison to the separate GP models and shared hyperparameter GP model is explained by the additional prior physical knowledge encoded into the model. In (b), the shared hyperparameter GP shows a slight advantage over the fully independent models.

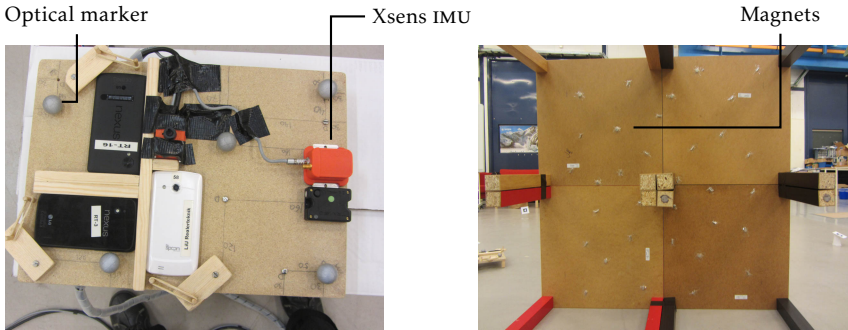
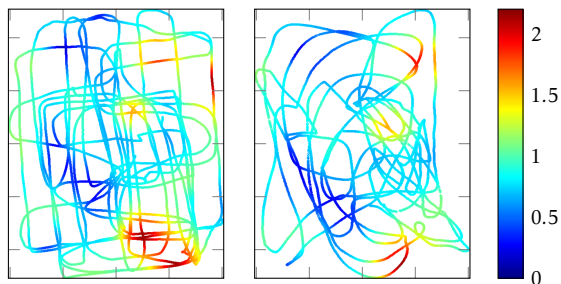


Figure 6: Setup of the experiment: On left the sensor board that was used for collecting the data. The right side figure shows the magnetic environment used in the experiment shown from below comprising small magnets to ensure sufficient excitation of the magnetic field.

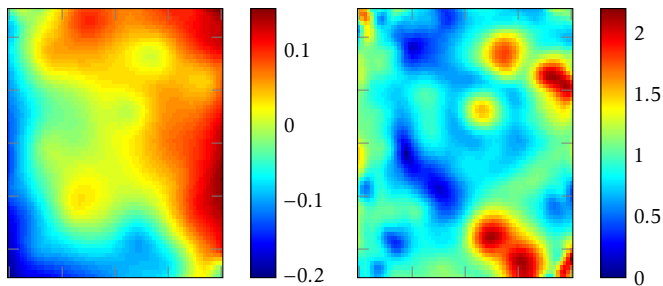
The sensor board used is shown in Figure 6. We use the magnetometer data from an Xsens MTi (Xsens Technologies B.V., <http://www.xsens.com>). Accurate position and orientation information is obtained using an optical system. These high-accuracy measurements were provided through the use of the Vicon real-time tracking system (Vicon Motion Systems Ltd., UK, <http://www.vicon.com>) courtesy of the UAS Technologies Lab, Artificial Intelligence and Integrated Computer Systems Division (AIICS) at the Department of Computer and Information Science (IDA), Linköping University, Sweden.

We obtain measurements while sliding the sensor board over a configuration of small tables. To ensure sufficient excitation, magnets have been placed in an irregular pattern underneath these tables as shown in Figure 6. Two different data sets have been collected. Both consist of approximately three minutes of data sampled at 100 Hz. One data set is used for training, while the second is for validation. The data of both the training and validation data sets are displayed in Figure 7a. To give an impression of the spatial variation of the magnetic field, the magnetic field intensity has been visualized through the colors of the data. Note that the magnetometer is calibrated such that it has a magnitude of one in a local undisturbed magnetic field.

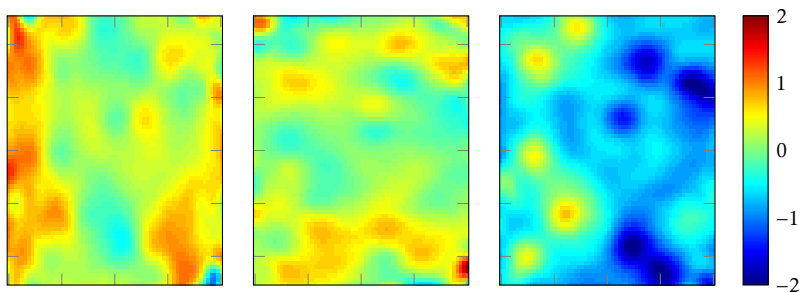
The magnetometer inside the IMU measures the magnetic field at the different locations. The optical measurements are used for two purposes. First, the positions from the optical system are used as known locations in the GP approach. This is a fairly reasonable assumption due to the high accuracy of the measurements of the optical system. Second, the orientations estimated by the optical system are used to rotate the magnetometer measurements from the magnetometer sensor frame to the lab frame. This rotation of the magnetic field measurements is needed for any of the GP methods discussed in this work. To use the optical and magnetometer data together, they need to be time synchronized. This synchronization is done in post-processing by correlating the angular velocities measured by the optical system and by the gyroscope in the IMU.



(a) Training data (left) and validation data (right).



(b) Learned scalar potential. (c) Norm of the magnetic field.



(d) The predicted magnetic field: H_x , H_y , and H_z .

Figure 7: Illustration of the magnetic field data and the results from the GP approach from Algorithm 1 for the experiment discussed in Section 6.2. In all figures, the y -axis is $-0.5, \dots, 0.5$ m. The x -axis is $-0.4, \dots, 0.4$ m. The units in the field surface plots are arbitrary due to normalization.

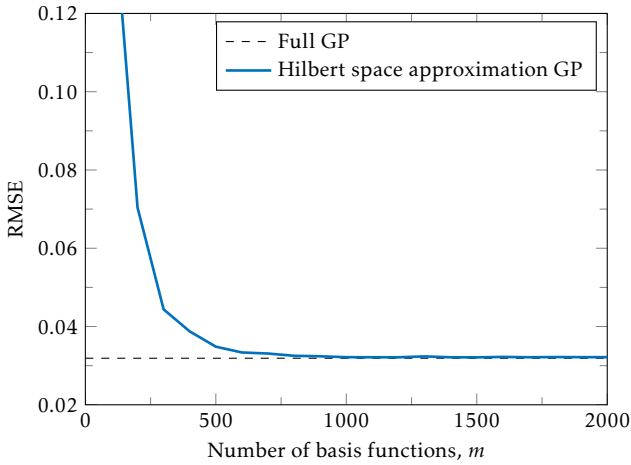


Figure 8: *RMSE as a function of the number of basis functions used in Algorithm 1 as compared to a validation data set. The approximative model approaches the full GP approach.*

We run Algorithm 1 for the training data set. The domain Ω in the GP model is set as $L_1 = L_2 = 0.6$ and $L_3 = 0.1$. The actual two-dimensional movement is performed in a rectangle of 80 cm $\times$ 100 cm and is hence well within the domain Ω . The computed scalar potential is shown in Figure 7b. From this, the predicted magnetic field measurements can be computed which are shown in Figure 7d. For completeness, the intensity of these predicted magnetic field measurements are shown in Figure 7c. Although this quantity is only indirectly related to the outcome of the GP approach, it is frequently used in the remaining sections because of its easy and intuitive visualization.

By predicting the measurements at the locations of the validation data set, it is also possible to compute the RMSE on the validation data. In Figure 8 we visualize the RMSE as a function of the number of basis functions used in Algorithm 1. We also compare to the RMSE from a full GP approach, that is using the same GP prior but without the Hilbert space approximation scheme to speed up the inference. To allow for comparison with a full GP approach—which suffers from a high computational complexity for large data sets—the data has been downsampled to 5 Hz. As can be seen, already for around $m = 1000$ basis functions, the quality of the estimates from Algorithm 1 approaches that of the full GP approach.

6.3 Mapping the magnetic field in a building

The third experiment was concerned with estimating a map of the magnetic environment inside a building by only using a smartphone for the data collection. The mapped venue is located on the Aalto University campus, and a floor plan sketch is shown in Figure 9. For practical reasons, we limited our interest to the lobby which is approximately 600 m² in size.

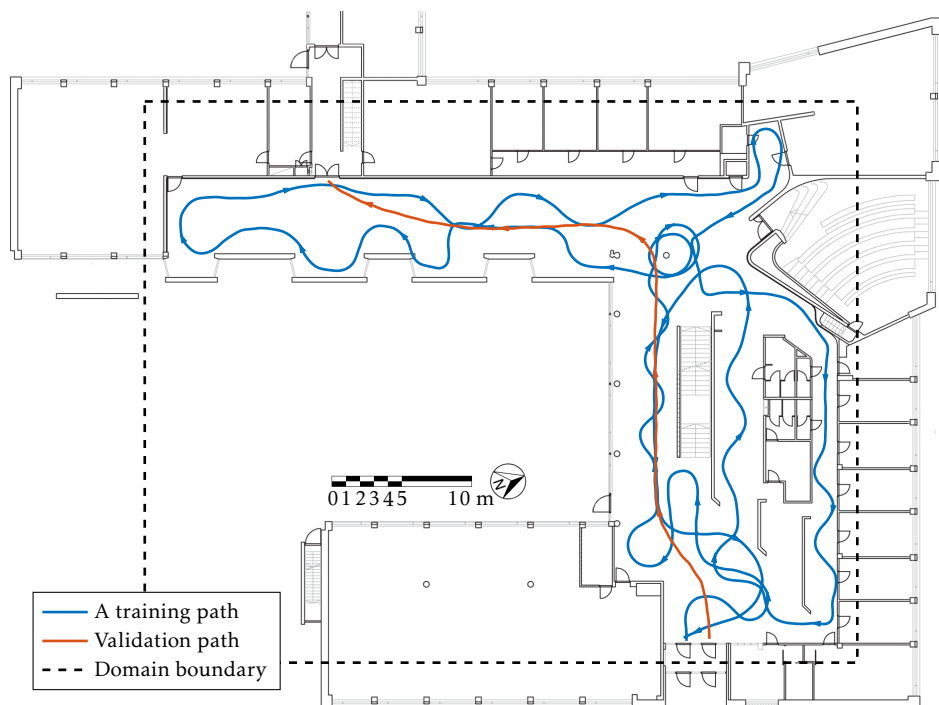


Figure 9: A training (red) and validation (blue) free-walking path that was used in the experiment. Trajectories were collected by a mobile phone, and the magnetometer data was corrected for gravitation direction and heading using the inertial sensors in the device. Walking direction markers are shown every 10 meters. The domain boundaries for the reduced-rank method are shown by the dashed line.

For the measurements, we used an Apple iPhone 4 and its built-in 9-dof IMU (3-axis AKM AK8975 magnetometer). All sensors were sampled at 50 Hz, and the data was streamed online to a laptop computer for processing and storing. The phone was held at waist-height and pointed towards the heading direction.

For reconstructing the walking path and phone orientation, we used a pedestrian dead-reckoning (PDR) approach developed at IndoorAtlas (IndoorAtlas Ltd., Finland, <http://www.indooratlas.com>), where only accelerometer and gyroscope readings were used—the path reconstruction thus being fully independent of the magnetometer readings. The alignment to the map and drift correction were inferred from a set of fixed points along the path during acquisition. Two sample paths are shown in Figure 9: one of the three training paths with similar routes and the validation path. The reconstructed paths were visually checked to match the ‘true’ walking paths.

We covered the walkable area in the lobby with three walking paths following the same route in each of them (see Figure 9). The reconstructed paths were

approximately 242, 253, and 302 meters long, respectively. The number of magnetometer data samples acquired along the paths were 9868, 10500, and 12335. Prior to each acquisition, the phone magnetometer was calibrated by a standard spherical calibration approach. The combined size of the training data set was $n = 32703$. For validation, we collected a walking path passing through the venue (length 54 m, $n = 2340$).

We considered a batch interpolation problem of creating a magnetic map of the lobby. The map was assumed static over time, and we applied Algorithm 1 to the training data with $m = 1024$ basis functions. The optimized hyperparameters were $\sigma_{\text{lin.}}^2 \approx 575 (\mu\text{T})^2$, $\sigma_{\text{SE}}^2/\ell_{\text{SE}}^2 \approx 373 (\mu\text{T})^2$, $\ell_{\text{SE}} \approx 1.87$ m, and $\sigma_{\text{noise}}^2 \approx 5.53 (\mu\text{T})^2$. The coefficients of the inferred linear (bias) model corresponding to the linear covariance function was $(-1.095, 12.995, -41.119)$.

Figure 1 shows the interpolated magnetic field magnitude ($\|f\|$) and the vector field components. The overall shape of the estimate agreed even when the model was trained separately with each of the training paths. To most part, the strong fluctuations in the magnetic field are located near walls or other structures in the building. The strong magnetic field in the open area in the lower right part of the floorplan was identified to most likely be due to a large supporting structure on the lower floor-level. We also used the model for predicting the measurements along the validation part (see Figure 9). The component-wise RMSEs were $(2.35 \mu\text{T}, 3.05 \mu\text{T}, 2.71 \mu\text{T})$ and mean absolute errors $(1.72 \mu\text{T}, 2.42 \mu\text{T}, 2.03 \mu\text{T})$. The measurement noise level of the magnetometer is in the magnitude of $1 \mu\text{T}$ and the uncertainty in the PDR estimate contributes to the remaining variance.

6.4 Online mapping

Finally, we demonstrate the power of sequential updating and time-dependent magnetic field estimation. The mapping was performed by a lightweight and inexpensive mobile robot equipped with a magnetometer, and the task was to obtain an estimate of the magnetic environment of an indoor space by re-calculating the estimate in an online fashion. In the second part of the experiment the magnetic environment was abruptly changed during the experiment, and the aim was to catch this phenomenon by spatio-temporal modeling.

We used a robot for collecting the data. The robot was built on a Diddy-Borg (PiBorg Inc., UK, <http://www.piborg.org>) robotics board, controlled by a Raspberry Pi 2 (model B) single-board computer (Raspberry Pi Foundation, UK, <http://www.raspberrypi.org>). For this example, we controlled the robot over Bluetooth with a joystick.

The robot was equipped with a 9-dof MPU-9150 Invensense IMU unit that was sampled at 50 Hz. The data were collected and stored internally on the Raspberry Pi. For additional validation, a Trivisio Colibri wireless IMU (TRIVISIO Prototyping GmbH, <http://www.trivisio.com/>), sampled at 100 Hz, and a Google Nexus 5 smartphone (AKM AK8963 3-axis magnetometer), sampled at 50 Hz, were also mounted on the robot for checking the quality of the Invensense IMU data. To reduce disturbances caused by the robot, the sensors were mounted on an ap-

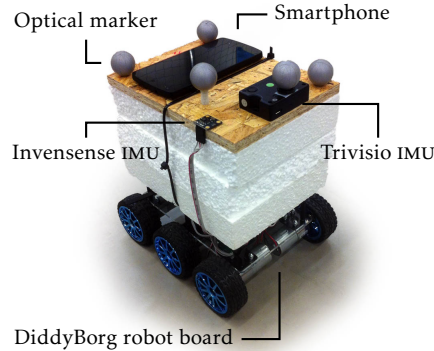


Figure 10: The robot was built on a DiddyBorg robotics board and controlled by a Raspberry Pi single-board computer. The three sensors (Invensense, Trivision, and smartphone) providing magnetometer readings were mounted on the top. The reference locations were provided by a Vicon optical tracking system.

proximately 20 cm thick layer of Styrofoam. During post-processing the data, the sensor positions and alignments on the robot were corrected for.

High-accuracy location and orientation reference measurements were provided through the use of the Vicon (Vicon Motion Systems Ltd., UK, <http://www.vicon.com>) real-time tracking system courtesy of the UAS Technologies Lab, Artificial Intelligence and Integrated Computer Systems Division (AIICS) at the Department of Computer and Information Science (IDA), Linköping University, Sweden. The location measurements could alternatively be recovered by odometry and heading information provided by the robot, but the interest in this experiment was rather to focus on the interpolation of the magnetic field, not the path estimation.

The task was to map the magnetic field inside a marked region roughly 6 m $\times$ 6 m in size. The size of the region was limited by the field of view of the Vicon system. The magnetometers were calibrated in the beginning of the measurement session by rotating the robot around all of its axes. A standard spherical calibration approach was used. Due to limits in acquisition length of the Vicon system, we captured the data in parts, each roughly three minutes in length. The magnetic environment remained unchanged for the first five data sets (paths shown in Figures 11c–11g), and later on changes in the field were initiated by bringing in large metallic toolbox shelves.

In the first part of the experiment, for interpolating the magnetic field we used the sequential reduced-rank scalar-potential approach presented in Algorithm 2. We assumed the magnetic environment stationary, and performed sequential updates in an online fashion. For practical reasons the calculations were done off-line, but the algorithm is fast enough for running in real-time. The rank of the approximation was fixed to $m = 1024$.

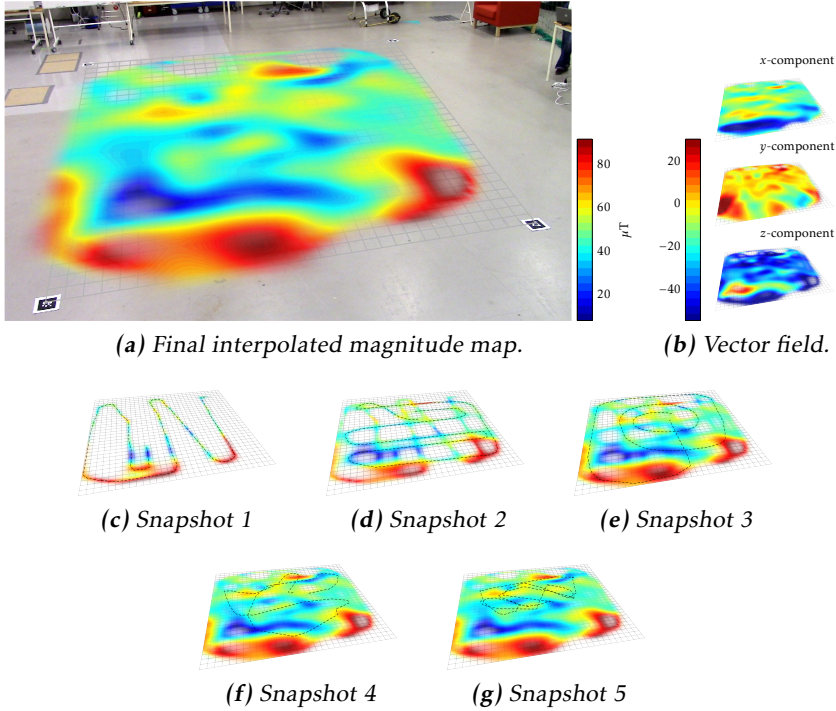


Figure 11: The GP interpolation task the robot was faced with. The final interpolation outcome of the magnitude field is shown in (a), and the different vector field components are shown in (b). Snapshots along the temporally updating field estimate are shown in (c–g) together with the path travelled since the previous update. The marginal variance (uncertainty) is visualized by the degree of transparency.

The length-scale, magnitude, and noise variance hyperparameters were learned from the first two data sets ($n = 17980$ vector valued observations) by maximizing with respect to marginal likelihood (see Algorithm 1). The obtained values were $\ell_{SE} \approx 0.32$ m, $\sigma_{SE}^2/\ell_{SE}^2 \approx 287 (\mu T)^2$ and $\sigma_{noise}^2 \approx 3.27 (\mu T)^2$. The linear model magnitude scale parameter was fixed to $\sigma_{lin}^2 = 500 (\mu T)^2$. The noise model is not only capturing the sensor measurement noise, but the entire mismatch between the data and the model. This explains the rather large noise variance. We also checked, that the hyperparameter estimates remained stable when optimized using the rest of the data.

Figure 11 shows the results for the static magnetic field experiment. The estimate was updated continuously five times a second, and we show five snapshots of the evolution of the magnetic field estimate in Figures 11c–11g (vector field magnitude shown in figures). These snapshots also show the path travelled since the previous snapshot. The alpha channel acts as a proxy for uncertainty; the marginal variance of the estimate is giving the degree of transparency. The final

magnitude estimate—after iterating through all the $n = 43029$ observations—is shown in Figure 11a together with the vector field components in Figure 11b.

The frontal part of the mapped region shows strong magnetic activity, whereas the parts further back do not show as strong fields. Inspection of the venue suggested metallic pipelines or structures in the floor to blame (or thank) for these features. In this particular case most parts of the effect is seen in the x -component. We repeated the reconstruction with data collected from the Trivisio and smart-phone sensors, and the results and conclusions remained unchanged. As a supplementary file to this paper, there is a video¹ demonstrating the online operation which has been sped-up 50 $\times$.

The last part of the experiment was dedicated to dynamical (time-dependent) modeling of the magnetic field. We used all the data from the first part of the experiment to train a sequential model and used that as the starting point for changing the field ($t = 0$). During acquisition of data while the robot was driving around, we brought in two metallic toolbox shelves: first a larger toolbox shelf on wheels (Figure 12c) and then a smaller box (Figure 12d). We acquired altogether some 300 s of data ($n = 15513$) of the changed environment.

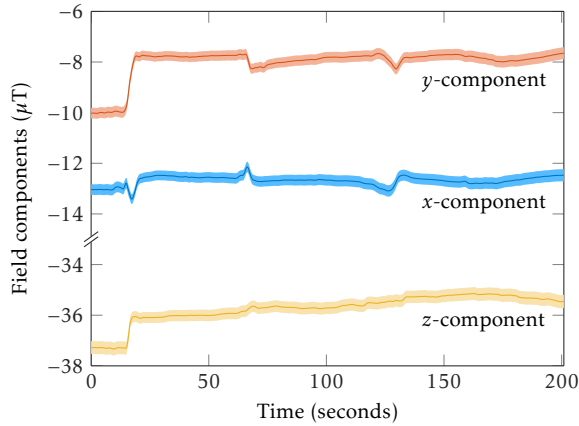
For encoding the assumptions of a changing magnetic field, we used Algorithm 2. The additional hyperparameter controlling the temporal scale was fixed to $\ell_{\text{time}} = 1$ hour, thus encoding an assumption of slow local changes. This choice is not restrictive, because the data is very informative about the abrupt changes.

Figure 12a shows the evolution of the magnetic field components for one fixed location (indicated by a red cross in the figures). The two toolboxes induce clear changes in the local anomaly field, but the effects are restricted to the immediate vicinity of the boxes. Thus the spatio-temporal model only gains information about the changed field, when the robot passes by the location of interest. This effect is clearly visible around $t = 20$ s, and later on around $t = 70$ s and $t = 130$ s. After this the estimate stabilizes and only drifts around for the remaining time. Even though the changes in the field components appear clear in Figure 12a, they are only around $2 \mu\text{T}$ and thus only account for a variation of about 2% in the scale of the entire field visualized in Figure 11b. Inducing more noticeable changes in the magnetic field would require moving around larger structures (say an elevator). Yet, even changes this small can be tracked by the modeling approach.

7 Discussion

In recent years, interest has emerged in mapping of the magnetic field by Gaussian processes for robot and pedestrian localization. This paper has aimed at presenting a new efficient method for mapping, but also at providing a study of best practices in using GPs in this context. Thus, we went through three different approaches for formulating GP priors for the magnetic field (independent, shared hyperparameters, and curl-free/scalar potential). These three models differ in

¹The supplementary video is available on YouTube: <https://www.youtube.com/watch?v=enlMiUqPVJo>



(a) Magnetic field estimate at a fixed location.

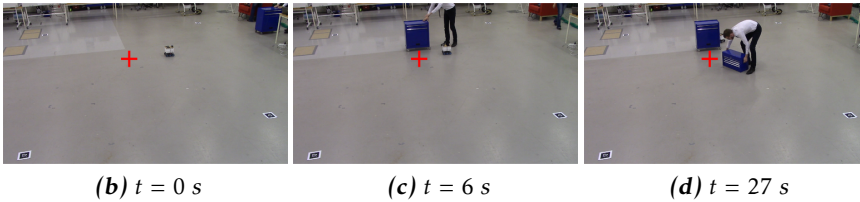


Figure 12: (a) Evolution of the magnetic field at one spatial location over a time-course of 200 seconds. (b–d) The blue metallic toolboxes are brought in at the beginning of the experiment. The abrupt changes in the field estimate corresponds to time instances when the robot has passed the toolboxes and gained information about the changed environment.

the amount of prior knowledge encoded in the model, and the more information available in the prior, the better the interpolation and extrapolation capabilities in the model—as long as the data agrees with the assumptions. This was also demonstrated in Figure 4 and in Section 6.1.

The methods presented in this paper are related to the use of Gaussian random field priors in inverse problems (Tarantola, 2004; Kaipio and Somersalo, 2005). This connection has been explored from various points of views (cf. Särkkä, 2011). However, the machine learning (Rasmussen and Williams, 2006) way of interpreting the GP priors indeed brings something new on the table—we are explicitly modeling the uncertainty in the field by using a stochastic model, which has interesting philosophical implications. The formulation of the prior through a GP covariance function provides both an intuitive and theoretically justified way of encoding the information.

Combining models from physics with GPs has also been studied under the name *Latent force models* by Álvarez and Lawrence (2009); Álvarez et al. (2013). The connection of these models with spatio-temporal Kalman filtering was stud-

ied, for example, in the work by Hartikainen and Särkkä (2011); Hartikainen et al. (2012); Särkkä and Hartikainen (2012); Särkkä et al. (2013). However, the Kalman filtering approach itself dates back to Curtain and Pritchard (1978) in the same way that GPs date back to O'Hagan (1978).

The scalar potential approach is not the only way to build the model. It would also be possible to include disturbance in the model, which would model the effect of free currents in the area or its boundaries. Furthermore, more complicated assumptions of the temporal time-changing behavior of the magnetic field could be included in the temporal covariance function. For example, various degrees of smoothness or periodicity could be included in the framework.

This paper has been considering the 'M' (mapping) part in SLAM. The 'L' (localization) part based on these maps is presented in Solin et al. (2016). The online mapping scheme presented in Algorithm 2 opens up for possibilities for simultaneously building the map and localization within the map. As seen in the experiments presented both in this work and in Solin et al. (2016), this appears feasible and provides an interesting direction for further research.

8 Conclusion

Small variations in the magnetic field can be used as inputs in various positioning and tracking applications. In this paper, we introduced an effective and practically feasible approach for mapping these anomalies. We encoded prior knowledge from Maxwell's equations for magnetostatics into a Bayesian non-parametric probabilistic model for interpolation and extrapolation of the magnetic field.

The magnetic vector field components were modeled jointly by a Gaussian process model, where the prior was associated directly with a latent scalar potential function. This ensures the field to be curl-free—a justified assumption in free spaces. This assumption couples the vector field components and additionally encodes the assumption of a baseline field with smooth small-scale variations. We also presented connections to existing formulations for vector-valued Gaussian process models.

In addition to constructing the model, we also presented a novel and computationally efficient inference scheme for interpolation and extrapolation using it. We built upon a Laplace operator eigenbasis approach, which falls natural to the formulation of the model. The inference scheme ensures a linear computational complexity with respect to the number of observations of the magnetic field. We also extended the method to an online approach with sequential updating of the estimate and time-dependent changes in the magnetic field.

We presented four experiments demonstrating the feasibility and practicality of the methods. A simulated experiment showed the benefit of including additional knowledge from physics into the model, and a simple proof-of-concept example demonstrated the strength of the approximation scheme in solving the model. Two real-world use cases were also considered: we mapped the magnetic field in a building on foot using a smartphone, and demonstrated online mapping using a wheeled robot.

Acknowledgements

This work was supported by grants from the Academy of Finland (266940, 273475) and by CADICS, a Linnaeus Center, by the project Probabilistic modelling of dynamical systems (contract number: 621-2013-5524), funded both by the Swedish Research Council (VR) and by the Swedish Foundation for Strategic Research under the project Cooperative Localization.

We are grateful for the help and equipment provided by the UAS Technologies Lab, Artificial Intelligence and Integrated Computer Systems Division (AIICS) at the Department of Computer and Information Science (IDA), Linköping University, Sweden. We also acknowledge the computational resources provided by the Aalto Science-IT project. Finally, we would like to thank IndoorAtlas Ltd. for providing expertise and for lending us equipment for the measurements.

Bibliography

- M. A. Álvarez and N. D. Lawrence. Latent force models. In *Proceedings of the 12th International Conference on Artificial Intelligence and Statistics*, volume 5 of *JMLR Workshop and Conference Proceedings*, pages 9–16, Clearwater Beach, FL, USA, 2009.
- M. A. Álvarez, L. Rosasco, and N. D. Lawrence. Kernels for vector-valued functions: A review. *Foundations and Trends in Machine Learning*, 4(3):195–266, 2012.
- M. A. Álvarez, D. Luengo, and N. D. Lawrence. Linear latent force models using Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2693–2705, 2013.
- S. Anderson, T. D. Barfoot, C. H. Tong, and S. Särkkä. Batch nonlinear continuous-time trajectory estimation as exactly sparse Gaussian process regression. *Autonomous Robots*, 39(3):221–238, 2015.
- M. Angermann, M. Frassl, M. Doniec, B. J. Julian, and P. Robertson. Characterization of the indoor magnetic field for applications in localization and mapping. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–9, Sydney, Australia, November 2012.
- L. Baldassarre, L. Rosasco, A. Barla, and A. Verri. Vector field learning via spectral filtering. In *Machine Learning and Knowledge Discovery in Databases*, volume 6321 of *Lecture Notes in Computer Science*, pages 56–71. Springer, 2010.
- T. D. Barfoot, C. H. Tong, and S. Särkkä. Batch continuous-time trajectory estimation as exactly sparse Gaussian process regression. In *Proceedings of Robotics: Science and Systems*, Berkeley, CA, USA, July 2014.
- S. Barkby, S. B. Williams, O. Pizarro, and M. V. Jakuba. Bathymetric particle filter SLAM using trajectory maps. *The International Journal of Robotics Research*, 31(12):1409–1430, 2012.
- B. K. Bhattacharyya. Bicubic spline interpolation as a method for treatment of potential field data. *Geophysics*, 34(3):402–423, 1969.
- P. Calcagno, J.-P. Chilès, G. Courrioux, and A. Guillen. Geological modelling from field data and geological knowledge: Part I. Modelling method coupling 3D potential-field interpolation and geological rules. *Physics of the Earth and Planetary Interiors*, 171(1):147–157, 2008.
- N. Cressie and C. K. Wikle. *Statistics for Spatio-Temporal Data*. John Wiley & Sons, Hoboken, NJ, 2011.
- N. A. C. Cressie. *Statistics for Spatial Data*. Wiley series in probability and mathematical statistics. Wiley-Interscience, New York, revised edition, 1993.

- R. F. Curtain and A. J. Pritchard. *Infinite Dimensional Linear Systems Theory*. Springer-Verlag, 1978.
- M. P. Deisenroth, D. Fox, and C. E. Rasmussen. Gaussian processes for data-efficient learning in robotics and control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):408–423, 2015.
- H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: Part I. *IEEE Robotics & Automation Magazine*, 13(2):99–110, 2006.
- B. Ferris, D. Hähnel, and D. Fox. Gaussian processes for signal strength-based location estimation. In *Proceedings of Robotics: Science and Systems*, Philadelphia, PA, USA, 2006.
- M. Frassl, M. Angermann, M. Lichtenstern, P. Robertson, B. J. Julian, and M. Doniec. Magnetic maps of indoor environments for precise localization of legged and non-legged locomotion. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 913–920, Tokyo, Japan, November 2013.
- J. Fritz, I. Neuweiler, and W. Nowak. Application of FFT-based algorithms for large-scale universal kriging problems. *Mathematical Geosciences*, 41(5):509–533, 2009.
- E. J. Fuselier, Jr. *Refined Error Estimates for Matrix-Valued Radial Basis Functions*. PhD thesis, Texas A&M University, 2007.
- D. J. Griffiths. *Introduction to electrodynamics*, volume 3. Prentice Hall, New Jersey, 1999.
- A. Guillen, P. Calcagno, G. Courrioux, A. Joly, and P. Ledru. Geological modelling from field data and geological knowledge: Part II. Modelling validation using gravity and magnetic data inversion. *Physics of the Earth and Planetary Interiors*, 171(1):158–169, 2008.
- J. Hartikainen and S. Särkkä. Kalman filtering and smoothing solutions to temporal Gaussian process regression models. In *Proceedings of the International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 379–384, Kittilä, Finland, 2010. IEEE.
- J. Hartikainen and S. Särkkä. Sequential inference for latent force models. In *Proceedings of the International Conference on Uncertainty in Artificial Intelligence*, pages 311–318, Barcelona, Spain, 2011.
- J. Hartikainen, M. Seppänen, and S. Särkkä. State-space inference for non-linear latent force models with application to satellite orbit prediction. In *Proceedings of the 29th International Conference on Machine Learning*, pages 903–910, Edinburgh, Scotland, UK, 2012.

- J. Haverinen and A. Kemppainen. Global indoor self-localization based on the ambient magnetic field. *Robotics and Autonomous Systems*, 57(10):1028–1035, 2009.
- J. D. Hol. *Sensor fusion and calibration of inertial sensors, vision, ultra-wideband and GPS*. Dissertation no. 1368, Linköping University, Linköping, Sweden, June 2011.
- M. F. Huber. Recursive Gaussian process: On-line regression and learning. *Pattern Recognition Letters*, 45:85–91, August 2014.
- J. D. Jackson. *Classical Electrodynamics*. Wiley, New York, 3rd edition, 1999.
- J. Jung, T. Oh, and H. Myung. Magnetic field constraints and sequence-based matching for indoor pose graph SLAM. *Robotics and Autonomous Systems*, 70: 92–105, 2015.
- J. Kaipio and E. Somersalo. *Statistical and Computational Inverse Problems*, volume 160. Springer, 2005.
- A. Kemppainen, J. Haverinen, I. Vallivaara, and J. Röning. Near-optimal exploration in Gaussian process SLAM: Scalable optimality factor and model quality rating. In *Proceedings of the 5th European Conference on Mobile Robots (ECMR)*, pages 283–289, Örebro, Sweden, September 2011.
- M. Lázaro-Gredilla, J. Quiñonero-Candela, C. E. Rasmussen, and A. R. Figueiras-Vidal. Sparse spectrum Gaussian process regression. *Journal of Machine Learning Research*, 11:1865–1881, 2010.
- E. Le Grand and S. Thrun. 3-axis magnetic field mapping and fusion for indoor localization. In *IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 358–364, Hamburg, Germany, September 2012.
- J. J. Leonard and H. F. Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *Proceedings of the International Workshop of Intelligent Robots and Systems (IROS)*, volume 3, pages 1442–1447, Osaka, Japan, November 1991. IEEE.
- B. Li, T. Gallagher, A. G. Dempster, and C. Rizos. How feasible is the use of magnetic field alone for indoor positioning? In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–9, Sydney, Australia, November 2012.
- F. Mackay, R. Marchand, and K. Kabin. Divergence-free magnetic field interpolation and charged particle trajectory integration. *Journal of Geophysical Research: Space Physics*, 111(A6):A06208, June 2006.
- M. N. Nabighian, V. J. S. Grauch, R. O. Hansen, T. R. LaFehr, Y. Li, J. W. Peirce, J. D. Phillips, and M. E. Ruder. The historical development of the magnetic method in exploration. *Geophysics*, 70(6):33ND–61ND, 2005.

- S. T. O’Callaghan and F. T. Ramos. Gaussian process occupancy maps. *The International Journal of Robotics Research*, 31(1):42–62, 2012.
- A. O’Hagan. Curve fitting and optimal design for prediction (with discussion). *Journal of the Royal Statistical Society. Series B (Methodological)*, 40(1):1–42, 1978.
- M. Osborne. *Bayesian Gaussian Processes for Sequential Prediction, Optimisation and Quadrature*. PhD thesis, University of Oxford, Oxford, UK, 2010.
- C. J. Paciorek. Bayesian smoothing with Gaussian processes using Fourier basis functions in the spectralGP package. *Journal of Statistical Software*, 19(2):1–38, 2007.
- J. Quiñonero-Candela and C. E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6: 1939–1959, 2005.
- C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.
- P. Robertson, M. Frassl, M. Angermann, M. Doniec, B. J. Julian, M. Garcia Puyol, M. Khider, M. Lichtenstern, and L. Bruno. Simultaneous localization and mapping for pedestrians using distortions of the local magnetic field intensity in large indoor environments. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–10, 2013.
- S. Särkkä. Linear operators and stochastic partial differential equations in Gaussian process regression. In *Artificial Neural Networks and Machine Learning – ICANN 2011*, volume 6792 of *Lecture Notes in Computer Science*, pages 151–158. Springer, 2011.
- S. Särkkä. *Bayesian Filtering and Smoothing*. Cambridge University Press, 2013.
- S. Särkkä and J. Hartikainen. Infinite-dimensional Kalman filtering approach to spatio-temporal Gaussian process regression. In *Proceedings of the 15th International Conference on Artificial Intelligence and Statistics*, volume 22 of *JMLR Workshop and Conference Proceedings*, pages 993–1001, La Palma, Canary Island, 2012.
- S. Särkkä, A. Solin, and J. Hartikainen. Spatiotemporal learning via infinite-dimensional Bayesian filtering and smoothing. *IEEE Signal Processing Magazine*, 30(4):51–61, 2013.
- M. Smith, I. Posner, and P. Newman. Adaptive compression for 3D laser data. *The International Journal of Robotics Research*, 30(7):914–935, 2011.
- A. Solin and S. Särkkä. Hilbert space methods for reduced-rank Gaussian process regression. *ArXiv e-prints*, January 2014. arXiv:1401.5508.

- A. Solin, M. Kok, N. Wahlström, T. B. Schön, and S. Särkkä. Modeling and interpolation of the ambient magnetic field by Gaussian processes. *ArXiv e-prints*, September 2015. arXiv:1509.04634.
- A. Solin, S. Särkkä, J. Kannala, and E. Rahtu. Terrain navigation in the magnetic landscape: Particle filtering for indoor positioning. In *Proceedings of the European Navigation Conference*, Helsinki, Finland, May–June 2016.
- V. Springel. Smoothed particle hydrodynamics in astrophysics. *Annual Review of Astronomy and Astrophysics*, 48:391–430, 2010.
- A. Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. Other Titles in Applied Mathematics. SIAM, 2004.
- C. H. Tong, P. Furgale, and T. D. Barfoot. Gaussian process Gauss–Newton for non-parametric simultaneous localization and mapping. *The International Journal of Robotics Research*, 32(5):507–525, 2013.
- I. Vallivaara, J. Haverinen, A. Kemppainen, and J. Röning. Simultaneous localization and mapping using ambient magnetic field. In *Proceedings of the IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 14–19, Salt Lake City, UT, USA, September 2010.
- I. Vallivaara, J. Haverinen, A. Kemppainen, and J. Roning. Magnetic field-based SLAM method for solving the localization problem in mobile robot floor-cleaning task. In *Proceedings of the 15th International Conference on Advanced Robotics (ICAR)*, pages 198–203, Tallinn, Estonia, June 2011.
- J. Vanderlinde. *Classical Electromagnetic Theory*, volume 145 of *Fundamental Theories of Physics*. Springer, Dordrecht, 2004.
- A. Viseras Ruiz and C. Olariu. A general algorithm for exploration with Gaussian processes in complex, unknown environments. In *Proceeding of the IEEE International Conference on Robotics and Automation (ICRA)*, Seattle, WA, USA, May 2015.
- N. Wahlström. *Modeling of Magnetic Fields and Extended Objects for Localization Applications*. Dissertation no. 1723, Linköping University, Linköping, Sweden, 2015.
- N. Wahlström, M. Kok, T. B. Schön, and F. Gustafsson. Modeling magnetic fields using Gaussian processes. In *Proceedings of the 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 3522 – 3526, Vancouver, Canada, May 2013.
- O. J. Woodman. *Pedestrian localisation for indoor environments*. PhD thesis, University of Cambridge, United Kingdom, September 2010.

Paper G

MEMS-based inertial navigation based on a magnetic field map

Authors: Manon Kok, Niklas Wahlström, Thomas B. Schön, Fredrik Gustafsson

Edited version of the paper:

M. Kok, N. Wahlström, T. B. Schön, and F. Gustafsson. MEMS-based inertial navigation based on a magnetic field map. In *Proceedings of the 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 6466–6470, Vancouver, Canada, May 2013.

MEMS-based inertial navigation based on a magnetic field map

Manon Kok[★], Niklas Wahlström[†], Thomas B. Schön[†], Fredrik Gustafsson[★]

[★]Dept. of Electrical Engineering,
Linköping University,
SE-581 83 Linköping, Sweden
{manko, fredrik}@isy.liu.se

[†]Dept. of Information Technology
Uppsala University,
SE-751 05 Uppsala, Sweden
{firstname.lastname}@it.uu.se

Abstract

This paper presents an approach for 6D pose estimation where MEMS inertial measurements are complemented with magnetometer measurements assuming that a model (map) of the magnetic field is known. The resulting estimation problem is solved using a Rao-Blackwellized particle filter. In our experimental study the magnetic field is generated by a magnetic coil giving rise to a magnetic field that we can model using analytical expressions. The experimental results show that accurate position estimates can be obtained in the vicinity of the coil, where the magnetic field is strong.

1 Introduction

With the reducing cost of accelerometers and gyroscopes (inertial sensors) and magnetometers, these sensor are becoming increasingly available in day-to-day life. It is for instance common that these sensors are present in modern smartphones. Positioning based on inertial sensors alone suffers greatly from drift and does not give reliable estimates for any but the highest quality sensors. Because of this, sensors such as GPS and ultra-wideband are often used as an aiding source (Hol, 2011). While GPS solutions only work for outdoor applications, indoor solutions are often highly dependent on additional infrastructure.

Magnetometers are a reliable source of information due to their high sampling rates and reliable sensor readings. They measure the superposition of the local earth magnetic field and the magnetic field induced by magnetic structures in the vicinity. Magnetometers are widely used as a source of heading information, relying on the assumption that no magnetic disturbances are present. Especially

in indoor applications this assumption is often violated due to the presence of steel in the construction of buildings and objects like radiators, tables and chairs.

This paper presents a method to obtain accurate position and orientation estimates based on inertial and magnetometer data assuming a map of the magnetic field is known. This enables positioning with widely available sensors, without requirements on additional infrastructure.

In recent years, the idea of using the presence of magnetic disturbances as a source of position information has started appearing in the literature. Most interest is from the robot localization perspective where odometry information is available (Suksakulchai et al., 2000; Navarro and Benet, 2009; Vallivaara et al., 2011; Georgiou and Dai, 2010). Generally, in these applications localization is only considered in 2D, and the sensor is assumed to be rotating around only one axis. To the best of the authors' knowledge, little work has been done on combining inertial and magnetometer measurements, for example Vissière et al. (2007); Dorveaux et al. (2011). This is a more challenging problem compared to using odometry information, since low grade inertial measurement units (IMUs) generally have poor dead-reckoning performance. The approach presented in Vissière et al. (2007) is not based on magnetic field maps, but uses knowledge about the physical properties of the magnetic field and its gradient to aid localization using an extended Kalman filter approach. Other approaches focus on using sensors in smartphones for localization (Chung et al., 2011; IndoorAtlas, 2012; Gozick et al., 2011) and consider magnetometer data only or very limited information from the inertial sensors. The direction of the magnetic field can, however, only be derived from the magnetic field measurements when the sensor orientation is known. Not estimating the full orientation therefore poses constraints on the allowed sensor rotations. In our approach no constraints on the sensor rotations are required since the full 6D pose is estimated.

To isolate the problem of localization inside a known magnetic field map from the problem of obtaining the map, this work assumes that the magnetic field map is known and is generated by a magnetic coil. The reason for using a magnetic coil is that it is one of the few cases for which the magnetic field can be computed analytically. In other words, we have a perfect model describing the magnetic field produced by the magnetic coil. The magnetic field measurements can be described as a nonlinear function of the sensor position in this map and its orientation with respect to the map.

2 Models

Before introducing the dynamic and measurement equations, the relevant coordinate frames and the state vector will be introduced. All measurements are assumed to be obtained in the body coordinate frame denoted by b , which is the coordinate frame of the measurement unit with the origin in the center of the accelerometer triad. The position is tracked in the earth coordinate frame denoted by e , which is fixed in the world. The magnetic field map is represented in the map coordinate frame denoted by m whose orientation is assumed to be aligned with

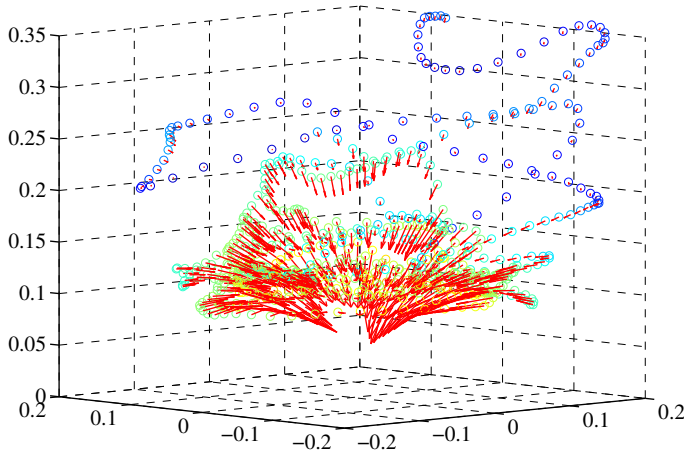


Figure 1: Magnetometer measurements represented in the earth coordinate frame. The measurements have been preprocessed by subtracting the earth magnetic field. The magnitude is indicated by the colors and the direction by the arrows.

that of the coil. The origin of the earth coordinate frame e is assumed to coincide with that of the map coordinate frame and with the center of the magnetic coil.

The relevant state vector consists of the sensor's position p^e and velocity v^e , its orientation with respect to the earth frame expressed as a unit quaternion $q^{eb} = (q_0 \ q_1 \ q_2 \ q_3)^T$ and the gyroscope bias b_ω^b . In our model we have used the inertial measurements as inputs to the dynamic equations in order to not increase the state dimension. For reasons that will become clear after the model has been provided, we split the state vector into two parts $x_t = ((x_t^n)^T \ (x_t^l)^T)^T$, where

$$x_t^n = ((p_t^e)^T \ (q_t^{eb})^T)^T, \quad x_t^l = ((v_t^e)^T \ (b_\omega^b)^T)^T. \quad (1)$$

2.1 Dynamical model

The dynamical equations can be derived by using the inertial measurements as inputs. A commonly used, slowly time-varying random walk model is used for the gyroscope bias (Hol, 2011). This leads to the following state update equations

for the linear and nonlinear states (Hol, 2011; Törnqvist, 2008)

$$x_{t+1}^n = \underbrace{\begin{pmatrix} \mathcal{I}_3 & 0 \\ 0 & \mathcal{I}_4 \end{pmatrix} x_t^n}_{A^{nn}} + \underbrace{\begin{pmatrix} T\mathcal{I}_3 & 0 \\ 0 & -\frac{T}{2}\tilde{S}(q_t^{\text{eb}}) \end{pmatrix} x_t^l}_{A_t^{nl}(x_t^n)} + \underbrace{\begin{pmatrix} \frac{T^2}{2}\mathcal{R}(q_t^{\text{eb}}) & \frac{T^2}{2}\mathcal{I}_3 & 0 \\ 0 & 0 & \frac{T}{2}\tilde{S}(q_t^{\text{eb}}) \end{pmatrix} u_t}_{B_t^n(x_t^n)} + \underbrace{\begin{pmatrix} \frac{T^2}{2}\mathcal{R}(q_t^{\text{eb}}) & 0 \\ 0 & \frac{T}{2}\tilde{S}(q_t^{\text{eb}}) \end{pmatrix} \begin{pmatrix} w_{a,t}^b \\ w_{\omega,t}^b \end{pmatrix}}_{G_t^n(x_t^n) w_t^n} \quad (2a)$$

$$x_{t+1}^l = \underbrace{\begin{pmatrix} \mathcal{I}_3 & 0 \\ 0 & \mathcal{I}_3 \end{pmatrix} x_t^l}_{A^{ll}} + \underbrace{\begin{pmatrix} T\mathcal{R}(q_t^{\text{eb}}) & T\mathcal{I}_3 & 0 \\ 0 & 0 & 0 \end{pmatrix} u_t}_{B_t^l(x_t^n)} + \underbrace{\begin{pmatrix} T\mathcal{R}(q_t^{\text{eb}}) & 0 \\ 0 & \mathcal{I}_3 \end{pmatrix} \begin{pmatrix} w_{a,t}^b \\ w_{b,\omega,t}^b \end{pmatrix}}_{G_t^l(x_t^n) w_t^l}. \quad (2b)$$

Here, $\mathcal{I}_k$ denotes the identity matrix of size $k \times k$, $\mathcal{R}(q_t^{\text{eb}}) \in SO(3)$ is the rotation matrix obtained from the unit quaternion q_t^{eb} and¹

$$\tilde{S}(q_t^{\text{eb}}) = \begin{pmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{pmatrix}. \quad (3)$$

The input vector u_t is given by

$$u_t = \left((y_{a,t}^b)^\top \quad (g^e)^\top \quad (y_{\omega,t}^b)^\top \right)^\top, \quad (4)$$

where g^e denotes the gravity vector and the accelerometer and the gyroscope measurements are denoted by y_a^b and y_ω^b , respectively. The latter are modeled as

$$y_{a,t}^b = R_t^{\text{be}}(a_t^e - g^e) + w_{a,t}^b, \quad (5a)$$

$$y_{\omega,t}^b = \omega_t^b + b_\omega^b + w_{\omega,t}^b, \quad (5b)$$

based on the fact that the accelerometer measures both the gravity vector and the body's free acceleration. The noise is modeled as

$$w_a^b \sim \mathcal{N}(0, Q_a), \quad Q_a = \sigma_a^2 \mathcal{I}_3, \quad (6a)$$

$$w_\omega^b \sim \mathcal{N}(0, Q_\omega), \quad Q_\omega = \sigma_\omega^2 \mathcal{I}_3, \quad (6b)$$

$$w_{b,\omega}^b \sim \mathcal{N}(0, Q_{b,\omega}), \quad Q_{b,\omega} = \sigma_{b,\omega}^2 \mathcal{I}_3. \quad (6c)$$

¹Note that the propagation of the quaternion state in this way is an approximation, valid only for high sampling rates. The algorithm does not prevent use of the exact update equation and the approximation is only used to reduce computational complexity.

The state noise is assumed to be distributed according to

$$w_t = \begin{pmatrix} w_t^n \\ w_t^l \end{pmatrix} \sim \mathcal{N}(0, Q), \quad (7a)$$

$$Q = \begin{pmatrix} Q^{nn} & Q^{nl} \\ (Q^{nl})^\top & Q^{ll} \end{pmatrix} = \begin{pmatrix} Q_a & 0 & Q_a & 0 \\ 0 & Q_\omega & 0 & 0 \\ Q_a^\top & 0 & Q_a & 0 \\ 0 & 0 & 0 & Q_{b_\omega} \end{pmatrix}. \quad (7b)$$

Note that the linear and nonlinear state noise is highly correlated since the accelerometer noise acts on both the position and velocity states. This needs to be taken into account in the implementation.

2.2 Magnetometer measurement model

The magnetometer measurements are modeled as

$$y_{m,t}^b = h(x_t^n) + e_{m,t}^b, \quad (8)$$

where $e_{m,t}^b \sim \mathcal{N}(0, R)$ and $h(x_t^n)$ is a function of the position p_t^e and orientation q_t^{eb} states. In practice this will be a superposition of the local earth magnetic field and all magnetic disturbances present.

As discussed in the introduction, to isolate the problem of positioning inside a map from the problem of making the map, we chose an experimental setup where the magnetic field is generated by a magnetic coil. In this case a magnetic field map is analytically known assuming the coil's position and orientation are known. The function $h(x_t^n)$ is given by

$$h(x_t^n) = \mathcal{R}(q^{be}) R^{\text{em}} B(R^{\text{me}} p_t^e). \quad (9)$$

The function $B(R^{\text{me}} p_t^e)$ gives the magnetic field in the map coordinate frame at a position p^m . The expression for the magnetic field from the coil is given by (Schepers, 2009)

$$B(p^m) = \frac{\mu_0 N_w I}{2\pi \sqrt{(\sqrt{p_x^2 + p_y^2} + a)^2 + p_z^2}} \begin{pmatrix} \frac{p_x p_z}{p_x^2 + p_y^2} \left[-K(k) + \frac{a^2 + p_x^2 + p_y^2 + p_z^2}{(\sqrt{p_x^2 + p_y^2} - a)^2 + p_z^2} E(k) \right] \\ \frac{p_y p_z}{p_x^2 + p_y^2} \left[-K(k) + \frac{a^2 + p_x^2 + p_y^2 + p_z^2}{(\sqrt{p_x^2 + p_y^2} - a)^2 + p_z^2} E(k) \right] \\ \left[K(k) + \frac{a^2 - p_x^2 - p_y^2 - p_z^2}{(\sqrt{p_x^2 + p_y^2} - a)^2 + p_z^2} E(k) \right] \end{pmatrix}, \quad (10)$$

where $p^m = (p_x \ p_y \ p_z)$, μ_0 is the magnetic permeability in vacuum, a is the coil radius, N_w is the number of windings, I is the current through the coil and

$E(k)$ and $K(k)$ are given by the following elliptic integrals

$$E(k) = \int_0^{\pi/2} \sqrt{1 - k^2 \sin^2 \theta} d\theta, \quad (11a)$$

$$K(k) = \int_0^{\pi/2} \frac{1}{\sqrt{1 - k^2 \sin^2 \theta}} d\theta, \quad (11b)$$

where

$$k = \sqrt{\frac{4a\sqrt{p_x^2 + p_y^2}}{(\sqrt{p_x^2 + p_y^2} + a)^2 + p_z^2}}. \quad (12)$$

These equations implicitly assume that the origin of the earth coordinate frame coincides with that of the map coordinate frame. Note that our measurement model assumes that no background field is present.

2.3 Some additional words about the magnetic field model

The magnetic field of a coil is generally described as a function of the perpendicular distance p_z towards the coil and the radial distance $r = \sqrt{p_x^2 + p_y^2}$ towards the center of the coil (Schepers, 2009; Griffiths, 1999). However, in tracking we are interested in absolute position rather than just the distance to a source. Parametrizing the magnetic field in terms of a position p_x, p_y, p_z introduces unobservability. Assuming the coil is placed horizontally, this results in two horizontal circles, one above and one below the coil, where the horizontal position is coupled to the heading as an unobservable manifold. We assume that the sensor can only be positioned above the coil and therefore have an entire circle of solutions at each time step. Note that in the more general case where multiple magnetic sources are present and possibly rotated with respect to each other, the unobservable manifold will be differently shaped or in some cases non-existent. To make our dynamic model applicable to any magnetic field map, we have not adapted the parametrization of our state vector to this specific structure.

3 Computing the estimate

As can be seen from the dynamical and measurement model presented in Section 2, the state dynamics is assumed to be linear while the measurement model is a nonlinear function of the sensor's position and orientation. A nonlinear filtering technique is therefore needed to compute a state estimate. A linear substructure can, however, be recognized, which can be exploited using a Rao-Blackwellized particle filter (RBPF) in which the state is split into a state x^1 that enters linearly in

both the dynamic and measurement model and a state x^n that enters non-linearly, where x^l and x^n are defined by (1). An RBPF solves the nonlinear filtering problem by using a Kalman filter (KF) for the linear states and a particle filter (PF) for the nonlinear states.

The RBPF in this paper has been derived from Törnqvist (2008) and Lindsten (2011) and is summarized in Algorithm 1. It applies the model structure (2), (8), the noise assumptions (6) and their correlations given in (7). In (13), $\bar{x}_t^i$ and $\bar{P}_t^i$ are computed, which are a stacked version of the nonlinear and linear states and covariances. Based on these, the nonlinear and linear time update are given by (14), (15) respectively. Note that in (15) the pseudo-inverse, denoted by $\dagger$, of $\bar{P}_t^{nn,i}$ needs to be taken because this matrix is rank deficient due to the presence of quaternion states.

Since the measurement model (9) only depends on the nonlinear states, measurement information about the linear states is in our problem only available through the nonlinear states. Algorithm 1 does therefore not contain an explicit KF measurement update. However, measurement information implicitly present in the nonlinear states is taken into account in the linear states in (15).

3.1 RBPF-MAP

To compare particle filter estimates to reference data, a point estimate needs to be computed at each time step. The most commonly used approach for this is to take the conditional mean estimate. Due to the unobservability in our model (see Section 2.3), however, all particles on a horizontal circle are equally likely, which can lead to an uninformative point estimate in center of the circle.

In Driessen and Boers (2008); Saha et al. (2009) a maximum a posteriori estimate for the particle filter (PF-MAP) has been derived, which is argued to give a better point estimate in multi-modal applications. The PF-MAP estimate is an approximation of the MAP estimate given by

$$\hat{x}_{t|t}^{\text{MAP}} = \arg \max_{x_t^i} p(y_t | x_t^i) \sum_j p(x_t^i | x_{1:t-1}^j) w_{t-1}^j. \quad (16)$$

Following a similar reasoning, the RBPF-MAP estimate, can be shown to be

$$\hat{x}_{t|t}^{\text{MAP}} = \arg \max_{x_t^{n,i}, x_t^{l,i}} p(y_t | x_t^{n,i}, x_t^{l,i}) \sum_j w_{t-1}^j \mathcal{N}(x_t^i; \bar{x}_{t|t-1}^j, \bar{P}_{t|t-1}^j), \quad (17)$$

where $\bar{x}_{t|t-1}^j$ and $\bar{P}_{t|t-1}^j$ can be obtained from (13). Note that since our problem does not have a KF measurement update, instead of the commonly used double subscript denoting the time for the linear states, Algorithm 1 only uses a single subscript.

When implementing this in Step 2 of the Algorithm 1, it needs to be taking into account that the covariance matrix $\bar{P}_t^j$ is rank deficient due to the presence of quaternion states. Because computation of (17) is computationally heavy, it could also be considered to use the most probable particle of the posterior. This would lead to similar results in Section 4.

Algorithm 1 Rao-Blackwellized particle filter

1. Initialization: For $i = 1, \dots, N$ generate $x_0^{n,i} \sim p_{x_0^n}$, set $\{x_0^{l,i}, P_0^i\} = \{x_0^l, P_0\}$, $\gamma_{-1}^i = \frac{1}{N}$, and set $t = 0$.
2. Measurement update: For $i = 1, \dots, N$ evaluate the particle importance weights $\gamma_t^i = \frac{1}{c_t} \gamma_{t-1}^i p(y_t | x_{0:t}^{n,i}, y_{0:t-1})$ based on (8) where $c_t = \sum_{i=1}^N \gamma_{t-1}^i p(y_t | x_{0:t}^{n,i}, y_{0:t-1})$.
3. If $t > 0$, compute the estimate $\hat{x}_t$ based on (17).
4. Resampling: If $\hat{N}_{\text{eff}} = \frac{1}{\sum_{i=1}^N (\gamma_t^i)^2} < \frac{2}{3}N$, resample N particles with replacement from the set $\{x_t^{n,i}, x_t^{l,i}\}_{i=1}^N$ where the probability to take sample i is γ_t^i , and reset the weights to $\gamma_t^i = \frac{1}{N}$.
5. Time update: Determine the Gaussian mixture

$$\bar{x}_{t+1}^i = A_t^i x_t^i + B_t^i u_t, \quad (13a)$$

$$\bar{P}_{t+1}^i = A_t^{l,i} P_t^i (A_t^{l,i})^\top + G_t^i Q (G_t^i)^\top, \quad (13b)$$

where

$$\begin{aligned} \bar{x}_t^i &= \begin{pmatrix} \bar{x}_t^{n,i} \\ \bar{x}_t^{l,i} \end{pmatrix}, & \bar{P}_t^i &= \begin{pmatrix} \bar{P}_t^{nn,i} & \bar{P}_t^{nl,i} \\ (\bar{P}_t^{nl,i})^\top & \bar{P}_t^{ll,i} \end{pmatrix}, \\ A_t^{l,i} &= \begin{pmatrix} A_t^{nl,i}(x_t^{n,i}) \\ A_t^{ll,i} \end{pmatrix}, & A_t^i &= \begin{pmatrix} A_t^{nn} & A_t^{nl,i}(x_t^{n,i}) \\ 0 & A_t^{ll,i} \end{pmatrix}, \\ B_t^i &= \begin{pmatrix} B_t^{n,i}(x_t^{n,i}) \\ B_t^{l,i}(x_t^{n,i}) \end{pmatrix}, & G_t^i &= \begin{pmatrix} G_t^{n,i}(x_t^{n,i}) & 0 \\ 0 & G_t^{l,i}(x_t^{n,i}) \end{pmatrix}. \end{aligned}$$

The nonlinear states can now sampled according to

$$x_{t+1}^{n,i} \sim \mathcal{N}(\bar{x}_{t+1}^{n,i}, \bar{P}_{t+1}^{nn,i}), \quad (14)$$

and the linear states can be updated according to

$$x_{t+1}^{l,i} = \bar{x}_{t+1}^{l,i} + (\bar{P}_{t+1}^{nl,i})^\top (\bar{P}_{t+1}^{nn,i})^\dagger (x_{t+1}^{n,i} - \bar{x}_{t+1}^{n,i}), \quad (15a)$$

$$P_{t+1}^i = \bar{P}_{t+1}^{ll,i} - (\bar{P}_{t+1}^{nl,i})^\top (\bar{P}_{t+1}^{nn,i})^\dagger \bar{P}_{t+1}^{nl,i}. \quad (15b)$$

6. Set $t := t + 1$ and iterate from Step 2.
-

4 Experimental results

4.1 Experimental setup

An experiment has been performed in which the magnetic field is generated by a magnetic coil where the number of windings N_w is equal to 50, the current I through the coil is 1 A and the radius a of the coil is 6 cm. A MEMS IMU (Xsens MTi) providing synchronized inertial and magnetometer measurements at a sampling frequency of 100 Hz is used. A picture of the experimental setup can be found in Figure 2. Ground truth data is collected from an optical reference system (Vicon system) and is used for validation of the estimates as well as for determining the position and orientation R^{em} of the coil.

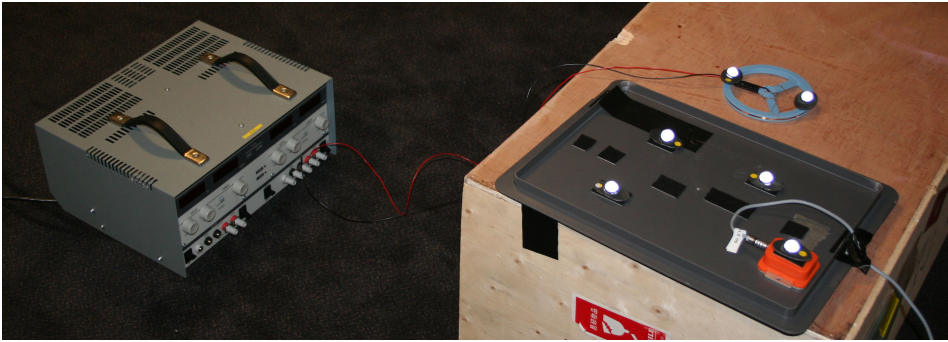


Figure 2: The experimental setup consisting of an IMU (orange box), a coil and a power supply. Optical markers are present, used for obtaining ground truth data, via an optical reference system.

Before the magnetometer measurements can be used in Algorithm 1, they need to be preprocessed for two reasons. First, the model (9) assumes that the magnetometer only measures the magnetic field due to a coil. A constant term representing the local earth magnetic field therefore needs to be determined and subtracted from all measurements. Second, the IMU used outputs magnetometer measurements in arbitrary units, while the model (9) determines the magnetic field in Tesla. A constant multiplication on all axes is therefore needed. Both constants are obtained by determining a best estimate from a part of the data where the magnetic disturbance is (approximately) zero. The preprocessed data is illustrated in Figure 1. The circles represent the preprocessed magnetometer measurements, downsampled to 4 Hz. The color of the circles represents the magnitude of the magnetic field. The magnetic field falls off cubically with distance which explains why the magnitude of the magnetic field is reduced quickly with distance from the coil. Each preprocessed measurement also gives rise to a red arrow indicating the direction of the magnetic field. The length of the arrows illustrates the magnitude.

4.2 Results

Using the collected inertial and magnetometer data, Algorithm 1 can be applied to obtain state estimates. Due to the fact that the magnitude of the magnetic field falls off cubically with distance, all results in this section are based on data no further away from the coil's origin than 40 cm. These have been compared to the ground truth data from the reference system. This section focuses on analysis of the position estimates. Due to the unobservability discussed in Section 2.3 we do not expect exact matches between the RBPF estimates and the ground truth data. A good comparison of the quality of the estimates, however, are the radial position and height estimates. The error plots can be found in Figure 3. The RBPF is initialized around the true estimate using the reference data, but any other (reasonable) initialization will give comparable results.

As can be seen in Figure 3, very good position estimates are obtained. However, at approximately 42 s, there is a big peak in both the radial position and the height errors. This can be explained by the fact that at this time instant, the sensor is the furthest away from the coil, almost 40 cm. The approach presented in this work is thus able to obtain high accurate position estimates for longer times, only when the sensor remains close to the coil. This is a major limitation in using the magnetic field as a source of position information in the way presented in this paper. The further away from the magnetic disturbance the less informative the measurements become. Even though at 40 cm from the coil the signal to noise ratio is still good, tracking problems occur due to model errors. It is therefore important to have a good model of the magnetic field (Wahlström et al., 2013).

5 Conclusions and future work

This paper has shown that close to a magnetic distortion generated by a magnetic coil, good position and orientation estimates can be obtained from inertial and magnetometer data only. Ideas for future work include extending the magnetometer model to a more realistic measurement model. First trials show that we can probably deal with including the local earth magnetic field. We also aim at combining this work with Wahlström et al. (2013) into an approach where simultaneous localization and mapping (SLAM) is possible. Another future line of research aims at studying the unobservability manifolds from the magnetic field in different cases.

Acknowledgements

This work is supported by MC Impulse, a European Commission, FP7 research project and CADICS, a Linneaus Center funded by the Swedish Research Council (VR). The authors would like to thank Xsens Technologies for their support in starting this work as well as in collecting the data sets and Dr. Slawomir Grzonka for pointing out this interesting field of research.

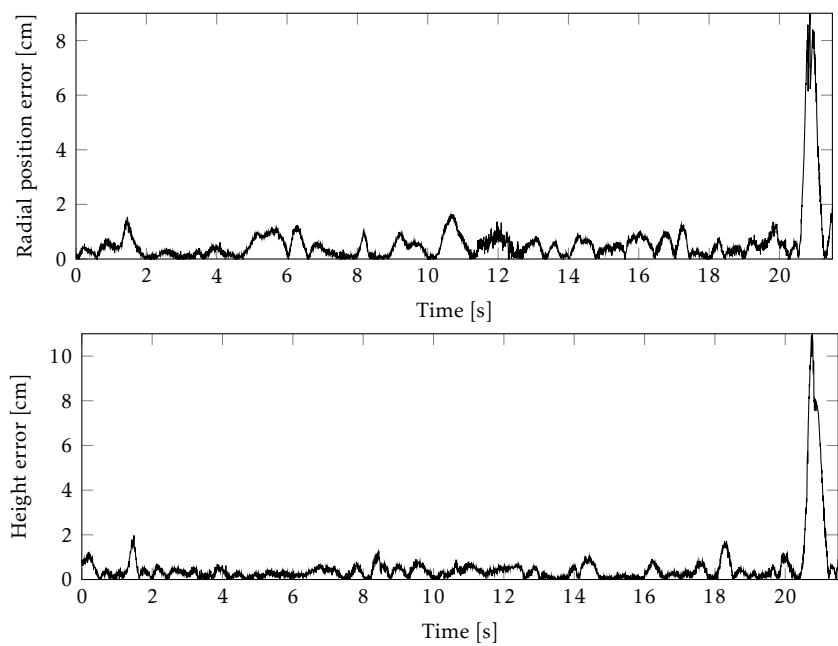


Figure 3: Error plots comparing the RBPF position estimates with the ground truth data from the optical reference system.

Bibliography

- J. Chung, M. Donahoe, C. Schmandt, I. J. Kim, P. Razavai, and M. Wiseman. Indoor location sensing using geo-magnetism. In *Proceedings of the 9th international conference on Mobile systems, applications, and services (MobiSys)*, pages 141–154, Bethesda, USA, June 2011.
- E. Dorveaux, T. Boudot, M. Hillion, and N. Petit. Combining inertial measurements and distributed magnetometry for motion estimation. In *Proceedings of the American Control Conference (ACC)*, pages 4249–4256, San Francisco, USA, June 2011.
- H. Driessen and Y. Boers. MAP estimation in particle filter tracking. In *IET Seminar on Target Tracking and Data Fusion: Algorithms and Applications*, pages 41–45, 2008.
- E. Georgiou and J. Dai. Self-localization of an autonomous maneuverable nonholonomic mobile robot using a hybrid double-compass configuration. In *Proceedings of the 7th International Symposium on Mechatronics and its Applications (ISMA)*, pages 1–8, Sharjah, United Arab Emirates, April 2010.
- B. Gozick, K. P. Subbu, R. Dantu, and T. Maeshiro. Magnetic maps for indoor navigation. *IEEE Transactions on Instrumentation and Measurement*, 60(12): 3883–3891, 2011.
- D. J. Griffiths. *Introduction to electrodynamics*, volume 3. Prentice Hall, New Jersey, 1999.
- J. D. Hol. *Sensor fusion and calibration of inertial sensors, vision, ultra-wideband and GPS*. Dissertation no. 1368, Linköping University, Linköping, Sweden, June 2011.
- IndoorAtlas. <http://www.indooratlas.com>, 2012. Accessed on November 30.
- M. Kok, N. Wahlström, T. B. Schön, and F. Gustafsson. MEMS-based inertial navigation based on a magnetic field map. In *Proceedings of the 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 6466–6470, Vancouver, Canada, May 2013.
- F. Lindsten. *Rao-Blackwellised particle methods for inference and identification*. Licentiate's thesis no. 1216, Linköping University, Linköping, Sweden, June 2011.
- D. Navarro and G. Benet. Magnetic map building for mobile robot localization purpose. In *Proceedings of the IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, pages 1–4, Mallorca, Spain, September 2009.
- S. Saha, Y. Boers, H. Driessen, P. K. Mandal, and A. Bagchi. Particle based MAP state estimation: A comparison. In *Proceedings of the 12th International Conference on Information Fusion*, pages 278–283, July 2009.

- H. M. Schepers. *Ambulatory assessment of human body kinematics and kinetics*. PhD thesis, University of Twente, Enschede, June 2009.
- S. Suksakulchai, S. Thongchai, D. M. Wilkes, and K. Kawamura. Mobile robot localization using an electronic compass for corridor environment. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, volume 5, pages 3354–3359, Nashville, USA, October 2000.
- D. Törnqvist. *Estimation and Detection with Applications to Navigation*. PhD thesis, Linköping University, November 2008. Thesis No. 1216.
- I. Vallivaara, J. Haverinen, A. Kemppainen, and J. Roning. Magnetic field-based SLAM method for solving the localization problem in mobile robot floor-cleaning task. In *Proceedings of the 15th International Conference on Advanced Robotics (ICAR)*, pages 198–203, Tallinn, Estonia, June 2011.
- D. Vissière, A.P. Martin, and N. Petit. Using magnetic disturbances to improve IMU-based position estimation. In *Proceedings of the European Control Conference (ECC)*, pages 2853–2858, Kos, Greece, July 2007.
- N. Wahlström, M. Kok, T. B. Schön, and F. Gustafsson. Modeling magnetic fields using Gaussian processes. In *Proceedings of the 38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 3522 – 3526, Vancouver, Canada, May 2013.

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.

E. 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. Tidfelt: 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.

J. D. Hol: Sensor Fusion and Calibration of Inertial Sensors, Vision, Ultra-Wideband and GPS. Thesis No. 1368, 2011. ISBN 978-91-7393-197-7.

D. Ankelhed: On the Design of Low Order H-infinity Controllers. Thesis No. 1371, 2011. ISBN 978-91-7393-157-1.

C. Lundquist: Sensor Fusion for Automotive Applications. Thesis No. 1409, 2011. ISBN 978-91-7393-023-9.

P. Skoglar: Tracking and Planning for Surveillance Applications. Thesis No. 1432, 2012. ISBN 978-91-7519-941-2.

K. Granström: Extended target tracking using PHD filters. Thesis No. 1476, 2012. ISBN 978-91-7519-796-8.

C. Lyzell: Structural Reformulations in System Identification. Thesis No. 1475, 2012. ISBN 978-91-7519-800-2.

J. Callmer: Autonomous Localization in Unknown Environments. Thesis No. 1520, 2013. ISBN 978-91-7519-620-6.

D. Petersson: A Nonlinear Optimization Approach to H2-Optimal Modeling and Control. Thesis No. 1528, 2013. ISBN 978-91-7519-567-4.

Z. Sjanic: Navigation and Mapping for Aerial Vehicles Based on Inertial and Imaging Sensors. Thesis No. 1533, 2013. ISBN 978-91-7519-553-7.

F. Lindsten: Particle Filters and Markov Chains for Learning of Dynamical Systems. Thesis No. 1530, 2013. ISBN 978-91-7519-559-9.

P. Axelsson: Sensor Fusion and Control Applied to Industrial Manipulators. Thesis No. 1585, 2014. ISBN 978-91-7519-368-7.

A. Carvalho Bittencourt: Modeling and Diagnosis of Friction and Wear in Industrial Robots. Thesis No. 1617, 2014. ISBN 978-91-7519-251-2.

M. Skoglund: Inertial Navigation and Mapping for Autonomous Vehicles. Thesis No. 1623, 2014. ISBN 978-91-7519-233-8.

S. Khoshfetrat Pakazad: Divide and Conquer: Distributed Optimization and Robustness Analysis. Thesis No. 1676, 2015. ISBN 978-91-7519-050-1.

T. Ardeshiri: Analytical Approximations for Bayesian Inference. Thesis No. 1710, 2015. ISBN 978-91-7685-930-8.

N. Wahlström: Modeling of Magnetic Fields and Extended Objects for Localization Applications. Thesis No. 1723, 2015. ISBN 978-91-7685-903-2.

J. Dahlin: Accelerating Monte Carlo methods for Bayesian inference in dynamical models. Thesis No. 1754, 2016. ISBN 978-91-7685-797-7.