TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

Tuukka Nieminen

A Non-recursive Solution Method for Fixed-interval Smoothing Problems Applied to Short-term Inertial Navigation

Julkaisu 1125 • Publication 1125

Tampere 2013

Tampereen teknillinen yliopisto. Julkaisu 1125
Tampere University of Technology. Publication 1125

Tuukka Nieminen

A Non-recursive Solution Method for Fixed-interval Smoothing Problems Applied to Short-term Inertial Navigation

Thesis for the degree of Doctor of Science in Technology to be presented with due permission for public examination and criticism in Sähkötalo Building, Auditorium S4, at Tampere University of Technology, on the 10th of May 2013, at 12 noon.

Tampereen teknillinen yliopisto - Tampere University of Technology
Tampere 2013

ISBN 978-952-15-3053-1 (printed)
ISBN 978-952-15-3055-5 (PDF)
ISSN 1459-2045

Abstract

This study focuses on *short-term inertial navigation* performed within a fixed time interval; one which is already over before the gathered data is processed. This yields a *fixed-interval smoothing problem*. The time interval is assumed to be short in order to simplify the equations related to inertial navigation without causing excessive errors to the *estimates of attitude, velocity, and position*, these values being the solutions to the problem. The aim is to develop a new solution method for applications of inertial navigation, particularly in sports, where the objective is often to ensure that the hardware can be integrated with the relevant equipment. This obviously imposes serious constraints on the size and mass of the used navigation system. Therefore, this study focuses on the use of *consumer-grade sensors*, and *new calibration methods* are also presented to improve the performance of such sensors.

The traditional approach to fixed-interval smoothing problems is based on the principle of combining two *recursive* filters, which are run forwards and backwards in time. This study, however, uses a *non-recursive solution method*. The advantages of this approach are best described with a single word: flexibility. Firstly, with this solution method there is no need to decide whether the fixed-interval smoothing problem is based on initial or boundary values, i.e. whether the *ordinary differential equation* describing the time evolution of the system is posed as an *initial value problem* or a *boundary value problem*. Secondly, it allows many forms of additional information to be used, which can be related to an arbitrary number of time instances. And thirdly, this solution method produces accurate results in the absence of any detailed knowledge of the involved errors.

The proposed non-recursive solution method uses a specific combination of the constructed state and observation equations in order to find a solution to the problem. The problem itself is expressed as a *Tikhonov regularization problem*, which allows one to obtain accurate results *without detailed knowledge of the involved errors*. When the problem is linear and the errors fulfill certain assumptions, the resulting solution is known to be the *best linear unbiased estimator*.

The main objective of this study is to construct a new solution method for fixed-interval smoothing problems; one which can be readily used in practical applications, where detailed knowledge of the involved errors is not available. The proposed solution method is presented in a detailed enough level to be implemented in a high-level environment such as Matlab®. Therefore, the thesis also presents a reference implementation of an algorithm designed to solve linear fixed-interval smoothing problems. This thesis concludes by applying the proposed solution method to two sports in which such technology has not been used before.

Preface

The research leading to this doctoral thesis was carried out in the unit of Electromagnetics at the Department of Electrical Engineering in Tampere University of Technology (TUT). It was year 2005 when my adviser, Prof. Lauri Kettunen asked me – then a 3rd year undergraduate student of electromagnetic field theory – to "check up on accelerometers". That was the job assignment which eventually led to this thesis. It has been a long and interesting journey, every single day of which has provided me a chance to learn something new. Back at year 2005, I did not know what I signed up for; I barely knew what is an accelerometer, let alone the whole concept of inertial navigation. By the time I received my Master's degree at year 2006, I thought that I knew something about inertial navigation. Today, I know enough about the topic to recognize the limitations of my knowledge about it. Moreover, I know that it was the right choice to sign up for the job.

My background in the electromagnetic field theory is supposedly quite different from the nominal background of people working in the field of inertial navigation. This has served both as an advantage and a disadvantage for this thesis. The advantage is that this kind of unorthodox background provides better than average possibilities to exercise "out of the box" -thinking, which is often required for creativity. I believe that the thesis indicates a fair amount of creativity and that is why I am quite pleased with the result.

The publishing-related disadvantage is that a combination of somewhat limited initial knowledge about the subject and a certain incapability to present the findings in the expected form can lead into extreme conservatism and less constructive feedback. I observed this in more than one occasion, and this is one of the two reasons why this thesis took longer to finish than I originally thought. The other reason is simply my incapability to do any better. Obviously, this thesis merely represents the best I could do at the time. Fortunately, a natural and ongoing phenomenon called progress has been present during all these years.

This thesis not only concludes my postgraduate studies, but nearly 24 years of formal education. During all these years, I have had the privilege to study and learn under the supervision of numerous exemplary instructors and teachers. At this point, I wish to express my gratitude to all of them. The following presents a concise educational biography of mine with references to people who have had a particularly memorable contribution to my path towards this point.

During years 1989 – 1998 at Kuorevesi comprehensive school, I am especially grateful to *Jarmo Kangas* and *Jussi Eloranta* for inspiring mathematics, physics, and chemistry lessons. Jarmo taught me the meaning of Fool's mate in the hard way; I have not made the same mistake twice. Jussi presented a very practical illustration of Occam's razor, although it admittedly took some time for me to recognize this.

Between years 1998 – 2001 at Jämsä high school, I am particularly grateful to *Martti Hartus* and *Pentti Isotalo* for numerous enthusiastic mathematics and physics courses. I have not encountered any problems in my physics studies since I made the decision to retake a course of classical mechanics, which I had trouble understanding the first time. It was during the courses taught by Martti and Pentti when my educational goals brightened up. Moreover, Martti's sarcastic comment about nobody understanding anything about Maxwell's equations did have a non-zero contribution to my career choices. During my years in high school, I am also indebted to *Raimo Viertola* from the Department of Physics at Jyväskylä University for challenging and highly educational exercise sets during an additional physics course.

During a year (2001 – 2002) of a bit different kind of education, I am especially grateful to company commander *Arro Jäntti* and reserve officers' school instructor *Riku Valtonen* for showing what it really means to be precise and well-organized. While this year did not provide too much knowledge about natural sciences, it was a period of determination and great spiritual growth. Patience (resulting from spiritual growth) and determination are both very necessary qualities in science and engineering.

TUT has provided me the rest of my formal education since year 2002. There, the introductory mathematics and physics courses taught by *Armo Pohjavirta* and *Jorma Keskinen* have been particularly memorable. Considering only the knowledge of prime importance for this thesis, I am indebted and grateful to *Robert Piché* and *Saku Suuriniemi*. Together with the time invested in resolving the assigned, highly non-trivial exercise problems, the inspiring lectures kept by Robert and Saku have had a major contribution to my current knowledge about (engineering) mathematics.

Lastly, an observation that all instructors and teachers mentioned by name

can be characterized by a single word: demanding. In my experience, the courses most useful in the end are the ones that require lots of determination, time, and hard work. As phrased by Erwin Rommel,

"Sweat saves blood, blood saves lives, but brains saves both."

Professionally, this thesis would certainly not have been possible without the contributions of Prof. *Lauri Kettunen*, lecturer *Jari Kangas*, and associate Prof. *Saku Suuriniemi*. Lauri, above all, provided me the opportunity to do this work. Jari and Saku have done hard and invaluable work in converting my initial manuscripts into concise and mathematically solid articles. Jari also took the tedious job of careful evaluation and commenting of the thesis in its various phases of progress. If there is a clear thread to be seen in the thesis, it is there because of him. Saku, on the other hand, pre-evaluated my more or less useful initial thoughts on countless occasions. Without a doubt, these discussions have had a major contribution to the thesis. Hopefully there have been also moments where the discussions were actually symbiotic rather than parasitic. I am greatly indebted to you all for your altruistic work for me. I would also like to thank Adrian Benfield for proofreading this thesis.

During my years in our unit, I have never encountered either administrative or economical problems, allowing me to concentrate fully on the job. This is anything but a coincidence; here, amongst Lauri, thanks are due to *Lasse Söderlund* and *Maija-Liisa Paasonen*. The unit of Electromagnetics has been a superb and an encouraging atmosphere to work in. For this, I would like to thank every current and former employee of our unit. Particularly, thanks are due especially to *Olli Särkkä*, but also to *Arttu Rasku* and *Matti Pellikka* for help in various tasks related to this thesis. Last but not least, I express my gratitude to *Pasi Raumonen* and *Janne Keränen* for their help as "senior practitioners of science".

Finally, I wish to express the greatest possible gratitude to my family (in the broad sense). Without your support, I would not have been able to complete this study. I am especially grateful to my parents *Lasse* and *Tuula* as well as to my greatest-of-all supporter *Nuusku* for their inexhaustible support.

Tampere, April 2013

Tuukka Nieminen

Contents

Abstract **i**

Preface **iii**

List of Publications **ix**

List of Figures **xi**

Lists of Symbols and Abbreviations **xiii**

1 Introduction **1**
1.1 Motivation 1
1.2 Literature review 4
1.3 Objective of the research 8
1.4 Author's contribution 10
1.5 The structure of the thesis 11

2 The essentials of inertial navigation **13**
2.1 Inertial navigation system 13
2.2 Frames of reference 14
2.3 Rotations between frames of reference 16
2.4 Motion equations 18
2.5 Simplified motion equations 19
2.6 Dynamics model 20
2.7 Summary 23

3 Calibration of IMUs **25**
3.1 The rate table 27
3.2 Reference measurements 29
3.3 Error analysis 32
3.4 Measurement model 35
3.5 Calibration procedure 37
3.6 Online calibration 38
3.7 Summary 40

4 State and observation equations **41**
4.1 State equation 43
4.1.1 Estimating attitude, velocity, and position 43
4.1.2 Estimating attitude, velocity, and position with online calibration 45
4.1.3 Estimating attitude 46
4.1.4 Estimating position and velocity 48
4.2 The observation equation 53
4.3 Summary 55

5 Solution methods **59**
5.1 Linear estimation problems 60
5.1.1 Problem formulation 63
5.1.2 The existence and uniqueness of the solution 64
5.1.3 Choosing the regularization parameter 66
5.1.4 Evaluating the curvature 68
5.1.5 A few remarks 71
5.2 Non-linear estimation problems 72
5.2.1 Problem formulation 73
5.2.2 Solution method 74
5.2.3 Evaluating the curvature 76
5.3 Summary 76

6 Examples **79**
6.1 Example 1: Curling 82
6.1.1 Observation equation 83
6.1.2 Results 86
6.2 Example 2: Ski jump 88
6.2.1 Observation equation 88
6.2.2 Results 91

7 Conclusions and open questions **95**

Bibliography **99**

A Reference implementation **107**

Publications **109**

List of Publications

This thesis consists of an introduction and the following publications, in chronological order:

[P1] Tuukka Nieminen, Jari Kangas, Saku Suuriniemi, and Lauri Kettunen
"An enhanced multi-position calibration method for consumer-grade inertial measurement units applied and tested"
Institute of Physics:
Measurement Science and Technology **21** (11pp), 2010
doi:10.1088/0957-0233/21/10/105204

[P2] Tuukka Nieminen, Jari Kangas, Saku Suuriniemi, and Lauri Kettunen
"Accuracy Improvements by Boundary Conditions for Inertial Navigation"
International Journal of Navigation and Observation (10pp), 2010
doi:10.1155/2010/869127

[P3] Tuukka Nieminen, Jari Kangas, and Lauri Kettunen
"Use of Tikhonov Regularization to Improve the Accuracy of Position Estimates in Inertial Navigation"
International Journal of Navigation and Observation (10pp), 2011
doi:10.1155/2011/450269

[P4] Tuukka Nieminen, Jari Kangas, Saku Suuriniemi, and Lauri Kettunen
"A non-recursive fixed-interval smoothing -based approach to attitude estimation"
Submitted to *IEEE Transactions on Aerospace and Electronic Systems*

List of Figures

1.1 Two possible realizations of measurement systems 4
1.2 Recursive and non-recursive solution schemes 8

2.1 A block diagram of an inertial navigation system 14
2.2 A graphical representation of the reference frames used 15
2.3 An illustration of the tangential velocity $\omega_{ib} \times p$ 17

3.1 An illustration of the rate table used to calibrate the sensors 28
3.2 Components of the gravitational acceleration 31
3.3 Error in the measured rotation rate of the rate table 34
3.4 Deterministic error sources of an IMU 35

4.1 Two different strategies to obtain estimates of C, v, and p . . 42
4.2 The chosen FEM basis functions 51

5.1 An example of the L-curve and its curvature 67

6.1 Allan deviations of the used sensors 81
6.2 Example 1: Used observations 83
6.3 Example 1: Measured specific forces and angular velocities . . 85
6.4 Example 1: Velocity as a function of traveled distance 87
6.5 Example 1: Resolved trajectories 88
6.6 Example 2: Used observations 89
6.7 Example 2: Measured specific forces and angular velocities . . 90
6.8 Example 2: Velocity as a function of traveled distance 92
6.9 Example 2: Resolved trajectories 93

Lists of Symbols and Abbreviations

Symbols

$\mathbb{C}$	Set of all complex numbers
$\mathbb{N}$	Set of all natural numbers
$\mathbb{R}$	Set of all real numbers
$\mathbb{R}_+$	Set of all non-negative real numbers
$\mathbb{C}^k$	k-dimensional complex-valued vector space
$\mathbb{R}^k$	k-dimensional real-valued vector space
$\mathbb{C}^{k\times k}$	$k \times k$-dimensional complex-valued vector space
$\mathbb{R}^{k\times k}$	$k \times k$-dimensional real-valued vector space
$0^{k\times k}$	Zero matrix with specified dimensions
a, v, p	Acceleration, velocity and position vectors: $a, v, p \in \mathbb{R}^3$
B_f, B_ω	Coordinate transformation matrices for the accelerometer and gyroscope triads from the respective s-frame to the b-frame: $B_f, B_\omega \in \mathbb{R}^{3\times 3}$
b_f, b_ω	Constant bias offsets of the accelerometer and gyroscope triads: $b_f, b_\omega \in \mathbb{R}^3$
$\hat{b}_f, \hat{b}_\omega$	Corrections to the bias offsets of the accelerometer and gyroscope triads: $\hat{b}_f, \hat{b}_\omega \in \mathbb{R}^3$
C	Direction cosine matrix (DCM): $C \in \mathbb{R}^{3\times 3}$
c_i	The ith column of a direction cosine matrix: $c_i \in \mathbb{R}^3$

$\delta\cdot$	Error in variable $\cdot$
$\delta_\cdot$	Tolerance related to variable $\cdot$: $\delta_\cdot \in \mathbb{R}_+$
e	Basis vector: $e \in \mathbb{R}^3$, $\|e\|_2 = 1$
η	Error norm, $\|W_o\|_F$ or $\|w_o\|_2$: $\eta \in \mathbb{R}_+$
F_s	Sampling frequency: $F_s \in \mathbb{R}_+$
$f, \hat{f}$	Compensated and uncompensated measurement of the specific force: $f, \hat{f} \in \mathbb{R}^3$
$f_{i,j}$	The specific force at the jth rotation rate of the ith attitude in the calibration phase: $f_{i,j} \in \mathbb{R}^3$
$f_o(x)$	Model describing the dependencies between states (y_i) and the observations
$f_s(x)$	Model describing the mutual dependencies between states (y_i)
g, g_l	Gravitational acceleration and local gravitational acceleration: $g, g_l \in \mathbb{R}^3$
$g_\parallel, g_\perp$	Components of gravitational acceleration g in the plane of and orthogonal to the rate table: $g_\parallel, g_\perp \in \mathbb{R}$
H	Average height of the object with respect to the sea level: $H \in \mathbb{R}$
h	Time step $t_{i+1} - t_i$ (constant): $h \in \mathbb{R}_+$
I	Identity matrix (square) of arbitrary dimension
i, j	Indices: $i, j \in \mathbb{N}$
J_s, J_o	Matrices containing the partial derivatives of $f_o(x)$ and $f_s(x)$ with respect to y_i
k	Dimension of the state vector y: $k \in \mathbb{N}$
L	Latitude of the local geographic frame: $0° \leq L \leq 90°$
λ	Regularization parameter: $\lambda \in \mathbb{R}_+$
λ_ω	Eigenvalue: $\lambda_\omega \in \mathbb{C}$
Λ	A diagonal matrix containing the eigenvalues λ_ω: $\Lambda \in \mathbb{C}^{3\times3}$
m	Mass of the object: $m \in \mathbb{R}_+$
M, N	Number of observations (M) and time steps (N): $M, N \in \mathbb{N}$

N_a, N_r	Number of attitudes (N_a) and rotation rates (N_r) exploited during the calibration process: $N_a, N_r \in \mathbb{N}$
n	Counter for iterative algorithms: $n \in \mathbb{N}$
$\Omega_{i,j}$	Rotation rate of the rate table at the jth rotation rate of the ith attitude in the calibration phase: $\Omega_{i,j} \in \mathbb{R}$
Ω_E	Rotation rate of the Earth: $\Omega_E = 7.292115 \times 10^{-5}$ rad/s
$\omega, \hat{\omega}$	Compensated and uncompensated measurement of the angular velocity: $\omega, \hat{\omega} \in \mathbb{R}^3$
$\omega_{i,j}$	The angular velocity at the jth rotation rate of the ith attitude in the calibration phase: $\omega_{i,j} \in \mathbb{R}^3$
Q	A square matrix containing the eigenvectors x_ω: $Q \in \mathbb{C}^{3\times 3}$
R_0	Mean radius of the Earth: $R_0 = 6.371 \times 10^6$ m
r_i	Rotation radius of the IMU at the ith attitude during the calibration: $r \in \mathbb{R}_+$
ρ	Error norm, $\|W_s\|_F$ or $\|w_s\|_2$: $\rho \in \mathbb{R}_+$
S_f, S_ω	$S_f = B_f \mathrm{diag}(s_f)$, $S_\omega = B_\omega \mathrm{diag}(s_\omega)$
s_f, s_ω	Constant scale factors of the accelerometer and gyroscope triads: $s_f, s_\omega \in \mathbb{R}^3$
$\hat{s}_f, \hat{s}_\omega$	Corrections to the scale factors of the accelerometer and gyroscope triads: $\hat{s}_f, \hat{s}_\omega$
Σ_o, Σ_s	Symmetric positive definite matrices containing variances and covariances related to w_o and w_s
σ_s^2, σ_o^2	Variances related to the state (σ_s^2) and observation (σ_o^2) equations: $\sigma_s^2, \sigma_o^2 \in \mathbb{R}_+$
T_0, T_1	Initial (T_0) and final (T_1) times of the fixed time interval: $T_0, T_1 \in \mathbb{R}$, $T_1 \geq T_0$
T_Ω	Rotation time of a single constant rotation rate Ω during the calibration phase: $T_\Omega \in \mathbb{R}_+$
t, t_i	Time and the ith instance of time: $t, t_i \in \mathbb{R}$
θ, θ_0	Rotation angle (θ) and initial rotation angle (θ_0): $\theta, \theta_0 \in \mathbb{R}$
X, x	Matrix- and vector-valued variables: $X \in \mathbb{R}^{kN\times q}$, $x \in \mathbb{R}^{kN}$
x_ω	Eigenvector: $x_\omega \in \mathbb{C}^3$
y	State variable: $y \in \mathbb{R}^k$

$y_i, \dot{y}_i, \ddot{y}_i$	State (y_i) and its first ($\dot{y}_i$) and second ($\ddot{y}_i$) time derivatives at time instance t_i: $y_i, \dot{y}_i, \ddot{y}_i \in \mathbb{R}^k$
W_o, w_o	Observation noise of linear (W_o) and non-linear (w_o) observation equations
W_s, w_s	Process noise of linear (W_s) and non-linear (w_s) state equations
Z_o, z_o	Constant part of linear (Z_o) and non-linear (z_o) observation equations
Z_s, z_s	Constant part of linear (Z_s) and non-linear (z_s) state equations
$\|\cdot\|_p$	p-norm of a vector: $\|\cdot\|_p \in \mathbb{R}_+$
$\|\cdot\|_F$	Frobenius norm of a matrix: $\|\cdot\|_F \in \mathbb{R}_+$
$\|\cdot\|_{max}$	Max norm of a matrix: $\|\cdot\|_{max} \in \mathbb{R}_+$
$[p_i]_j$	Refers to the jth component of the vector p_i, for example: $[p_i]_j \in \mathbb{R}$ (the brackets are used only when necessary for clarity)
$[\omega]_\times$	Skew-symmetric matrix form of angular velocity ω: $[\omega]_\times \in \mathbb{R}^{3\times3}$, $[\omega]_\times^T = -[\omega]_\times$
$\overline{\omega}$	The average of e.g. ω over an interval specified by the context
$\mathbb{E}(\cdot)$	Expectation value of $\cdot$
$\ker(\cdot)$	Kernel of matrix $\cdot$
$\mathcal{N}(\cdot)$	Null space of matrix $\cdot$
$\mathcal{R}(\cdot)$	Range of matrix $\cdot$
$\mathrm{diag}(\cdot)$	Diagonal matrix formed by the elements of vector $\cdot$
$\mathrm{Tr}(\cdot)$	Trace, the sum of the diagonal terms of a square matrix: $\mathrm{Tr}(\cdot) \in \mathbb{R}$
$\mathbb{V}(\cdot)$	Variance of $\cdot$

Abbreviations

BLUE	Best linear unbiased estimator
BVP	Boundary value problem
DAE	Differential-algebraic equation
DCM	Direction cosine matrix
DOF	Degree(s) of freedom
ECEF	Earth-Centered Earth-Fixed frame of reference
ECI	Earth-Centered Inertial frame of reference
FDE	Finite difference equation
FEM	Finite element method
GLS	Generalized least squares
GNSS	Global navigation satellite system
IVP	Initial value problem
IMU	Inertial measurement unit
INS	Inertial navigation system
MEMS	Micro-electro-mechanical system
MMSE	Minimum mean squared error
ODE	Ordinary differential equation
OLS	Ordinary least squares
SDE	Stochastic differential equation
SO(3)	Special orthogonal group (3D)
TLS	Total least squares
TUT	Tampere University of Technology
WLS	Weighted least squares

Chapter 1

Introduction

This thesis is based on four articles (P1 – P4), already published in scientific journals (or about to be; P4). The main purpose of the thesis is to present a coherent background to the articles – or a storyline, if you will. Moreover, the theory on which the solution method is based is extended to cover situations leading to non-linear problems, as these are commonly encountered in practical engineering.

The thesis can be read as a stand-alone document in the sense that the reader is not forced to plough through all the referenced articles in order to follow the treatment of the problem. The reader is, however, expected to have a reasonable understanding of (numerical) mathematics. Nevertheless, the reader need not have any detailed knowledge of the field, i.e. – *inertial navigation*, as the key concepts are introduced along the way.

1.1 Motivation

In the literature, the term *navigation* is often seen together with the terms *guidance* and *control* [1–3]. Navigation is concerned with determining the position of an object relative to a known reference [3]. Guidance, on the other hand, provides directions to its destination and control keeps it on a predetermined trajectory [3]. Obviously, these terms are closely related and particularly, guidance and control are dependent on the navigation. Navigation is, however, independent of guidance and control: it is both possible and relevant to determine an object's position without a known trajectory or destination.

Taken together, navigation, guidance and control imply *real-time navigation*. All guidance and control systems require timely feedback to work properly. Here, real-time navigation is defined as navigation where the time

delay of determining object's navigational *state* is shorter than the time interval between two successive samples of the input data. When dealing with *inertial navigation* in particular, the demand for real-time navigation implies that the navigation problem is formally presented as an *initial value problem* (IVP). Given an *initial state*, $y_0 \in \mathbb{R}^k$ ($k \in \mathbb{N}$), and an *ordinary differential equation* (ODE)

$$\dot{y} = f(t, y), \tag{1.1}$$

find y such that

$$\begin{cases} \dot{y} & = & f(t, y) \\ y(T_0) & = & y_0 \end{cases} \tag{1.2}$$

holds for all times $t > T_0$ (or $t < T_0$ as well). In (1.1), "$\dot{}$" represents the time derivative and for future reference (sections 4.1.1 and 4.1.2), it is assumed that $f : \mathbb{R} \times \mathbb{R}^k \to \mathbb{R}^k$ is continuously differentiable with respect to each component of the state vector y. Here, unless otherwise noted, the state refers to the attitude[1], velocity, and position of the object. Hereafter, ODE (1.1) will be referred to as the *dynamics model*.

Throughout their development since the 1940s, *inertial navigation systems* (INS) have mostly been used as a prerequisite for *self-contained* navigation, guidance, and control of objects such as submarines, ships, air- and spacecrafts, and guided munition [4]. Therefore, the associated algorithms and solution methods are designed for the purposes of real-time navigation. However, it is generally accepted that real-time inertial navigation is extremely sensitive to: 1) errors in the initial state, and 2) imperfections in the *inertial sensors*. Together, these sources of error are known to create positional errors proportional[2] to t, t^2 and t^3 [2].

The development of *micro-electro-mechanical systems* (MEMS) has led to mass production, and therefore initiated the ongoing process of improving the *cost-efficiency* of inertial sensors [2]. Consequently, over the last few decades the market for commercial and *consumer* applications for inertial sensors has grown rapidly. While many of these applications do not involve inertial navigation per se, the quality of contemporary sensors is – under certain conditions – already sufficient for this purpose; especially so, if real-time navigation is not required:

> This study focuses on inertial navigation performed within a *fixed time interval* $[T_0, T_1]$ ($T_1 \geq T_0$), which is already over before the gathered data is processed.

[1] In this context, the term attitude refers to the angular displacement of the object with respect to a known reference. Hence, it also includes the heading, which is sometimes used to refer specifically to the horizontal alignment of the object.

[2] The referred error equations are based on the approximate error dynamics of an inertial navigation system. Thus, the presented proportionals cannot be treated as global facts. However, they do give a realistic idea of the challenges related to inertial navigation.

In other words, although the pre-processing (such as sensor-error compensation) may take place during or before the time interval in question, the state estimates are always produced afterwards.

Therefore, there is no compelling reason why the navigation problem should be posed as an IVP. It may be preferable to do so for the sake of convenience, but it must be stressed that an IVP is not the only possible way to proceed. In fact, one can just as well pose the navigation problem as a *boundary value problem* (BVP)[3]: ODE (1.1) defined in the domain $t \in [T_0, T_1]$ with *boundary values* that do not fit the definition of an IVP.

> This thesis considers inertial navigation problems which can be posed as either a BVP or an IVP. The point is that the proposed solution scheme does not compel the user to specify any initial or boundary values. Thus, the user does not have to choose between the two possibilities.

This means that those solution methods designed strictly for IVPs are no longer valid, since BVPs require their own, specific, solution methods [6, 7]. Therefore, in this thesis it is assumed that the concept of *initial values* is encompassed by the concept of *boundary values*. In other words, any reference to the term boundary values in this thesis assumes that initial values are one possible realization of the term.

In order to justify such a major leap from the mainstream school of thought, prospective applications are needed. The trigger for this study was the need to design an affordable, small, low-power navigation system capable of determining the trajectory of a *ski jumper* [8]. Ski jumping is an application where the dynamics model can conveniently be presented as a BVP because in ski jumping it is sufficient to perform the actual computations after the event. It should be emphasized that there are many other applications suitable for the proposed approach besides ski jumping, e.g. other (elite) sport events[4] and many forms of general physical exercise. In fact, there are many events in which real-time navigation is not essential, and for which the dynamics model can conveniently be presented either as an IVP or a BVP.

Typically, any practical navigation system intended for the described type of applications has the same basic requirements: it should be small, accurate, affordable, and energy-efficient. Furthermore, for the sake of the usability, the hardware used in the navigation system has to be integrated into the

[3] The majority of the BVPs arising from ODEs are so-called *two-point BVPs*, where the boundary values are posed only at the two end points of the domain [5]. The boundary values of a general BVP, however, may be freely located within the domain.

[4] *Javelin throw* [9] and *curling* [10] are both examples of tested applications.

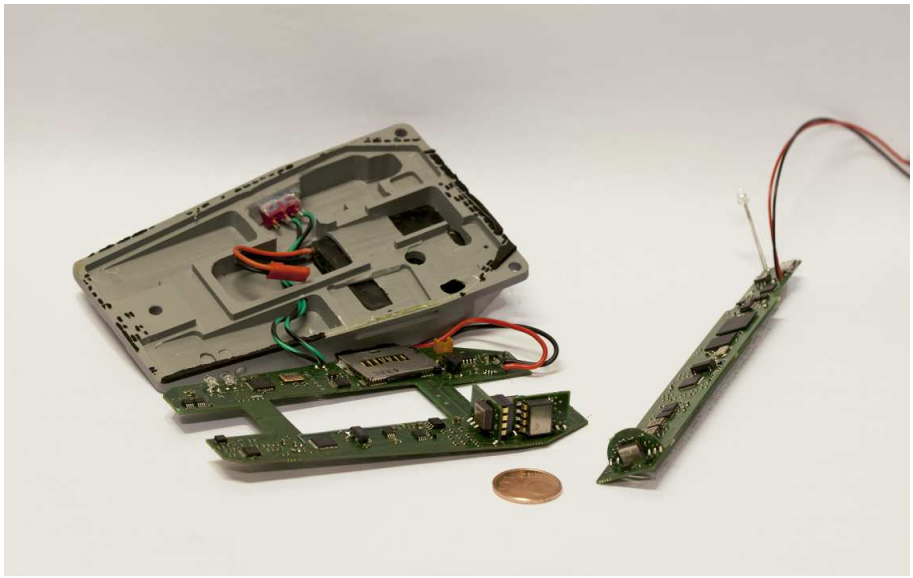

Figure 1.1: Two possible realizations of measurement systems suitable for these applications. The system on the left was used to collect and store the raw data shown in Example 1 (Chapter 6). The system on the right is a customized realization of a similar system which is integrated into an 800 gram javelin. For comparison, the diameter of the coin is 21.25 mm.

(sporting) equipment, and obviously this imposes serious constraints on the size and mass of the navigation system. Figure 1.1 shows two examples of such navigation systems. It must also be remembered that many sporting events take place in locations where it is not possible to consider solutions which rely on external references, such as global navigation satellite systems (GNSS)[5]. Yet another drawback with using non-INS methods is their operating speed, as relatively high sample rates are often required in order to successfully capture the finest details of the motion.

1.2 Literature review

This section covers relevant studies found in the literature and their shortcomings with regard to the objectives of this thesis. The hardware constraints are a very practical way to refine the topic. As stated above, the hardware must be small, affordable, and consume minimal power. In other words, the possibilities are effectively limited to *consumer-grade* sensors. Now, inertial navigation with consumer-grade sensors has long been con-

[5]Not to imply that the proposed solution method would be incapable of exploiting data from a GNSS: Timely position and/or velocity observations constitute excellent prerequisites for a successful estimation process.

sidered to be unstable or even impossible [2, 11], but it *is* possible as long as one accepts that the navigation period is limited by the quality of the sensors used. That is why the focus of this thesis is on *short-term inertial navigation* [2]. Obviously, the definition of "short" depends on the situation, but typically it refers to navigation periods of no more than 10 minutes (see section 2.5).

When using consumer-grade sensors in inertial navigation, the term *calibration* is bound to come up at some point. Short-term inertial navigation is no different; accurate results require precise measurements. Until now, the calibration of consumer-grade accelerometers has almost [12] exclusively been based on the use of gravitational acceleration as the only reference measurement [13–19]. The calibration of consumer-grade gyroscopes often requires special equipment, such as a rate table, to create reliable enough reference measurements [12, 13, 16]. A different approach is to calibrate the sensors on the move, without customized equipment [20, 21]. These are the *online calibration* methods referred to in this thesis. However, it must be remembered that these methods often come at the cost of a more elementary sensor-error model.

This thesis' main contribution to the subject of sensor calibration is to provide a range of accurate reference measurements for use in accelerometers and gyroscopes. This allows the user to focus on the performance of the sensors within a specific dynamic range. An online calibration method is also presented, whose basic concept is to apply a sensor-error model for the data gathered during an actual measurement event. Simply put, it utilises the gathered observations to establish the parameters of the error model. A detailed discussion of sensor calibration follows in Chapter 3.

Let us turn to the estimation of state. A classification of general estimation problems is a good starting point as it will aid in recognizing the type of problem at hand. According to [22], estimation problems can be divided into three classes. When estimating $y(t)$ by means of

- *filtering*, all measurements within the time interval $[T_0, t]$ are used
- *smoothing*, all measurements within the time interval $[T_0, T_1]$ are used and
- *prediction*, all measurements within the time interval $[T_0, t - d]$ $(d > 0)$ are used.

In the given circumstances, smoothing is the most relevant estimation technique, as all the measurements within the time interval $[T_0, T_1]$ need to be used. Smoothing can be further divided into three classes [22]:

- *fixed-interval smoothing*, where T_0 and T_1 are held fixed and $y(t)$ is estimated for all t within interval $[T_0, T_1]$

- *fixed-point smoothing*, where t and T_0 are held fixed and $y(t)$ is estimated as T_1 increases and
- *fixed-lag smoothing*, where $y(T_1 - d)$ is estimated as T_0 is held fixed and T_1 increases.

According to this classification, the problem at hand is a *fixed-interval smoothing* problem.

Currently, the most common solution methods for fixed-interval smoothing problems originate from the *(modified) Bryson-Frazier* [23–25], *Rauch-Tung-Striebel* [26] and *two-filter* [27] smoothers [28, 29]. All three methods – including their numerous extensions – consider the generally non-linear, dynamic, and discrete-time *state-space models* of the form

$$\begin{cases} y_{i+1} &= f_i(y_i) + w_i \\ z_i &= h_i(y_i) + v_i \end{cases} \quad \forall\, i \in [1, N], \tag{1.3}$$

where the state $y_{i+1} \in \mathbb{R}^k$ and *observation* $z_i \in \mathbb{R}^q$ ($q \in \mathbb{N}$). $N \in \mathbb{N}$ is the number of time steps and subscript i refers to a time instant t_i. In (1.3), w_i and v_i model the *process* and the *observation* noises. Moreover, the initial state y_1 is assumed to be known [22, 30]. Function $f_i : \mathbb{R}^k \to \mathbb{R}^k$ is obtained by approximating the dynamics model (1.1) with a suitable *finite difference equation* (FDE). This equation is referred to as the *state equation*. Function $h_i : \mathbb{R}^k \to \mathbb{R}^q$ describes the dependence relation between the state y_i and observation z_i; this equation is known as the *observation equation*. Note that in this thesis, the term *measurement* specifically refers to the outputs of the inertial sensors, while the term *observation* is used in a more general sense, and refers to many different types of available information. This distinction is necessary, as the state equation usually depends on the measurements whereas the observation equation depends on various individual observations.

The three above-mentioned fixed-interval smoothers can all be realized as two-pass algorithms consisting of a filtering pass followed by an additional filtering and/or smoothing pass in the reverse direction (see Figure 1.2). Typically, the first pass is carried out using the (extended) *Kalman filter* [22, 31, 32]. With regard to the second pass, all three solution methods are realized in different forms, but they are mathematically equivalent [33]. Particular applications of these three smoothers to inertial navigation, or more specifically, to *attitude estimation*[6] can be found from [29,32–43].

To model the underlying *stochastic process* with the *recursive* state-space model (1.3), it is assumed that y is a *Markov process* [22, 30]. A Markov process is a *causal* stochastic process whose state y_{i+1} depends only on

[6] Attitude estimation is a necessary part of inertial navigation.

the state y_i, not on the sequence of states that preceded it [44]. This, in turn, requires that the dynamics model is realized as an IVP [7]. If the dynamics model is realized as a BVP instead, y becomes a *non-causal*, non-Markovian stochastic process called a *reciprocal process* [7]. Therefore, the three cited fixed-interval smoothers are not applicable when the dynamics model is realized as a BVP.

Fixed-interval smoothing problems driven by BVPs have been discussed in the more recent literature [7, 45, 46]. This kind of problem can be solved using two-pass recursive algorithms similar to the ones mentioned above. However, all the existing solution methods assume that all the observations are given in the form depicted in (1.3). In contrast, the way the problem is approached in this thesis, the individual observations can be related to *an arbitrary number of time instants*. This is important when one wants to utilise all the available information about the situation. A simple example of an observation including more than one time instant is the average velocity of an object over a specified time. Another example could be where a part of the state of the object – itself unknown – is known to be the same at two or more time instants: i.e. the position of the object can be known to be the same at a number of time instants. Of course, there are also numerous types of "ordinary" observations which involve only a single time instant. Bearing this in mind, the observation equation provided in (1.3) is clearly not in the most flexible form. A substantially more flexible choice is to model the situation with a *static*, *non-recursive* observation equation of the form

$$f_o(x) = z_o + w_o. \tag{1.4}$$

In (1.4), $x = \begin{bmatrix} y_1^T & y_2^T & \cdots & y_N^T \end{bmatrix}^T$, $f_o : \mathbb{R}^{kN} \to \mathbb{R}^M$ ($M \in \mathbb{N}$), z_o is a constant, and w_o represents the observation noise.

Obviously, the state equation can also be written in the static, non-recursive form

$$f_s(x) = z_s + w_s, \tag{1.5}$$

where w_s represents the process noise. There are a number of methods capable of approximating the dynamics model in the form (1.5) [5]. This thesis uses either the *relaxation method* or the *finite element method* (FEM) to achieve this.

> While constructing the state equation, care has been taken *not* to include any information about the boundary values. Instead, the boundary values are *always* included in the observation equation. This is a crucial step towards a solution method in which the user does not need to choose specific boundary values, and can thus proceed without having to distinguish between an IVP and a BVP.

The details of this process – including the conditions for the existence and uniqueness of the solution – are discussed in sections 4 and 5.

Figure 1.2 represents an illustration of the key differences of the recursive and non-recursive solution schemes. The dashed ellipses illustrate individual observations, which can be related to an arbitrary number of time instances with the non-recursive solution scheme. The blue and red arrows depict the first and second passes of the two-phase solution scheme. The initial values depict the starting point. In the case of the non-recursive solution scheme, the solution is found within a single phase and no dedicated boundary values are required.

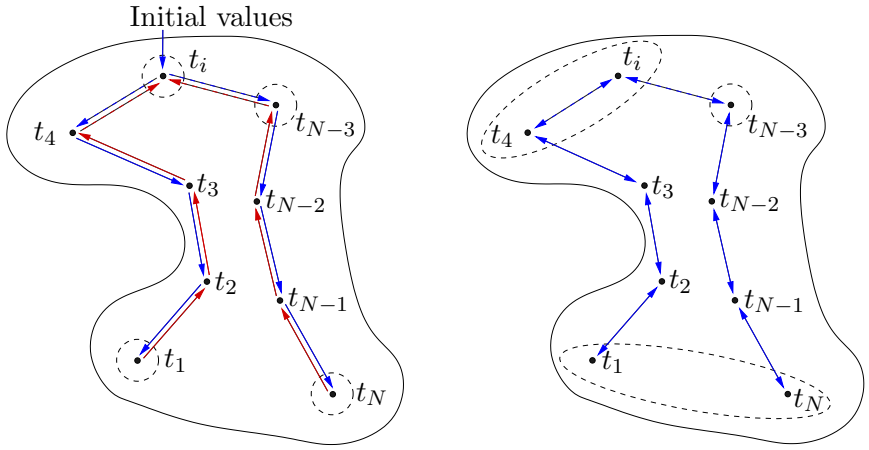

Figure 1.2: An illustration of the main principles of recursive (left-hand side) and non-recursive (right-hand side) solution schemes for fixed-interval smoothing problems.

1.3 Objective of the research

The objectives of this thesis are to construct:

(a) a *practical*[7] and *generic*[8] solution method for non-Markovian fixed-interval smoothing -based short term inertial navigation problems, which are stated in the non-recursive form

$$\begin{cases} f_s(x) &= z_s + w_s \\ f_o(x) &= z_o + w_o \end{cases} \tag{1.6}$$

[7] A practical solution method can be readily implemented and produces results in a reasonable time.

[8] A generic solution method is independent of the application and does not require specific information about the used equipment.

(b) a calibration method especially for consumer-grade inertial sensors.

Objective (a) is the main contribution of this thesis to the field of inertial navigation, i.e. the thesis proposes a new, non-recursive approach to the calculations. The point is that the treatment is not just limited to IVPs, which seem to have become hard-wired into any discussion of the topic. While the thesis does focus on fixed-interval smoothing problems, the treatment of the problem provides important new insights for people working in related fields. The proposed solution method has been extensively tested and used in a variety of applications. Finally, this thesis presents novel results for two specific sports applications.

Throughout the research we focus on the use of *consumer-grade* sensors, the performance of which is commonly thought of as being "not sufficient" for navigational purposes. The detailed treatment starts at the point at which the (raw) measurement data becomes available and ends when the requested state estimates have been obtained. It is beyond the scope of this thesis to discuss the significance of the state estimates obtained, or how they should be used. While it is our goal to design practical solution methods, no particular attention has been paid to optimizing the efficiency of the algorithms, which were implemented in a Matlab® environment (win32, version 7.12.0.635 / R2011a) [47].

The form of (1.6) is flexible enough to take advantage of many kinds of observations and – as it turns out – also to model situations where the properties of w_s and w_o are unknown. This is particularly interesting for those real-life situations where errors are always present, but seldom reliably characterized. The solution of the estimation problem (1.6) is obtained with a method similar to *Tikhonov regularization*, which has been extensively studied in the field of inverse problems [48–51]. In statistical literature, Tikhonov regularization is known as *ridge regression* [52–54]. Moreover, given certain assumptions, Tikhonov regularization and the *Wiener-Kolmogorov* filter have been shown to be equivalent [54–56]. The Wiener-Kolmogorov filter, in turn, is often referred to as the static counterpart of the prestigious Kalman filter [57].

In this thesis the focus is on how to adapt the proposed solution scheme to applications, where the readily-available fixed-interval smoothers are not applicable. The proposed solution scheme can also be applied to "classic" fixed-interval smoothing problems, which can, of course, also be solved with the existing solution methods. The comparison of the different approaches constitutes a starting point for further studies. Although this thesis focuses on new and emerging applications in which the established solution methods do not apply, it should be emphasized that the proposed solution scheme is

perfectly suitable for solving traditional IVPs and BVPs[9].

Now, let us further motivate the chosen, generic approach to inertial navigation. Namely, a particular application often comes with a specific

(a) goal with one or more quantities whose accuracy should be maximized

(b) set of sensors customized or carefully chosen to meet the application-dependent requirements and

(c) dynamics model (1.1) describing the time evolution of the system.

When available, these factors provide us with more information which should also be used to yield the best outcome. Then, depending on the quality of the available information, the resulting application-specific solution method may outperform the generic approach. With this in mind, it is important to notice that the non-recursive model (1.6) used in this thesis is general enough to exploit also the application-dependent information. In many situations, it is straightforward to do so, and this is pointed out by referring back to the mentioned items (a)–(c) at the appropriate parts of this thesis. However, the main focus of this thesis is on the generic approach, and the motivation for this is given below.

The proposed approach allows the resulting algorithm to be used with minimal costs to produce at least initial proof-of-concept results in a variety of applications. In many applications – like the ones considered in the examples – the results obtained with the generic algorithm are already rather accurate. If a given application requires more accurate results and shows potential to make application-specific development economically feasible, the proposed approach can be used as a starting point and a benchmark to improve on. However, it must be emphasized that there are a number of situations where it is crucial to have a single, generic algorithm suitable for a variety of applications. Elite sports and physiological analysis of human body are both examples of applications, where any given goal and situation would ultimately require one to develop a custom-made algorithm. Such an approach is hardly an economically feasible option, especially so when the market is small or the goal is to conduct academic research.

1.4 Author's contribution

The author is responsible for writing P1 – P4 and for the ideas and practical realizations outlined in them. Nevertheless, the publications would not have been possible without the contributions of Jari Kangas, Lauri Kettunen, and

[9] In fact, the BVP solver *bvp4c* implemented in Matlab® is based on a relaxation technique similar to the one employed here [58].

Saku Suuriniemi who provided invaluable help with finalizing the text and presenting the formal methodology behind the publications. The measurements referred to in publications P1 and P2 were performed by the author. The measurements in publications P3 and P4 were performed by the author in cooperation with Olli Särkkä.

The author has designed the specifications for the self-made hardware used in the measurements referenced in publications P1 – P4. The electronics were designed by Mika Oinonen and the embedded software of the electronics by Miika Pekkarinen. The required hardware installations were carried out by the technicians Hannu Nieminen and Pekka Nousiainen at the former Prototype Workshop of Tampere University of Technology (TUT).

1.5 The structure of the thesis

Chapter 2 presents the physical background to inertial navigation and explains the derivation of the mathematical models which are used to describe the connections between the measured quantities and the state estimates.

Chapter 3 is devoted to the calibration and compensation of inertial sensors, which is known to have a significant effect on the accuracy of the results. In practice, the calibration of consumer-grade sensors implies a balance between the resulting accuracy of the sensors and the costs involved in the calibration: a good calibration method should significantly increase the accuracy of the sensors, yet should be manageable with low-cost equipment. Therefore, in addition to the established calibration methods performed in a laboratory, a new online calibration method is also presented. The online calibration method allows us to model and compensate for a number of sensor-errors which can not (or have not) been removed in the laboratory calibration. Most of the contents of Chapter 3 can be verified by reference to publication P1.

Chapter 4 describes how to construct the state and observation equations (1.6). In order to give the reader a good understanding of the many possibilities of the proposed method a number of different formulations are introduced, each fit for use in different situations. These include linear formulations for solving the attitude and position estimates separately, as well as a generally non-linear formulation which can be used to solve estimates of attitude, velocity, and position together. The contents of this chapter relate to publications P2 – P4.

Once the reader knows how the state and the observation equations are constructed, the solution process is discussed in Chapter 5. While the con-

tents of this chapter are based on publications P3 and P4, the treatment is extended to cover non-linear models and some more practical issues related to the estimation process. The rationale behind the chosen solution scheme is also discussed in greater detail. This chapter also presents a Matlab® implementation of a linear fixed-interval smoothing problem solver.

Chapter 6 is devoted to practical examples, in which the methods described in the previous chapters are actually implemented in real-life situations. The first treats a three-degrees-of-freedom (3DOF) problem for determining the time-parametrized trajectory of a curling stone while the second example deals with a full 6DOF problem for determining the trajectory of a ski jumper. Conclusions and open questions are presented in Chapter 7, and the original publications are in the appendices.

Chapter 2

The essentials of inertial navigation

This chapter provides a solid mathematical framework for the following chapters to build on. Only those aspects of inertial navigation which are relevant to this thesis are covered. For any reasonably experienced engineer, this chapter serves as a short introduction to inertial navigation and defines the topic of this thesis. For an expert in the field, this chapter will introduce the notation and, more importantly, help to build a more precise picture of the subject. The treatment of this chapter is based on continuous-time models, the discretized versions of which are considered in Chapter 4.

2.1 Inertial navigation system

A good starting point is Figure 2.1 below, a block diagram showing some of the notation and building blocks of a *strapdown* INS. The *inertial measurement unit* (IMU) is an autonomous unit containing the sensors and their support electronics, as well as the built-in memory in which the measurements are stored for later use. Here, an IMU is assumed to contain a triad of accelerometers and gyroscopes. The outputs of the accelerometer and gyroscope triads are assumed to represent the momentary *specific force* $\hat{f} \in \mathbb{R}^3$, and the momentary *angular velocity* $\hat{\omega} \in \mathbb{R}^3$ respectively. The raw data from the sensors are assumed to be unprocessed, apart from the low-pass filters which are often integrated into the outputs of the sensor chips.

The error compensation block, where the unprocessed raw measurements $\hat{f}$ and $\hat{\omega}$ are converted to the respective measurements $f \in \mathbb{R}^3$ ($[f] = \mathrm{m/s^2}$) and $\omega \in \mathbb{R}^3$ ($[\omega] = \mathrm{rad/s}$), is also located within the IMU. While the actual conversion is often done as a part of the state estimation process, the

respective calibration parameters are unique for each particular IMU. The details of the error compensation block are presented in Chapter 3.

Given the observations and compensated measurements, the attitude $C \in \mathbb{R}^{3\times3}$ (see section 2.3), velocity $v \in \mathbb{R}^3$, and position $p \in \mathbb{R}^3$ of the object are estimated. In practice, the estimation process can take place on a mobile phone, computer or similar platform once a complete set of measurements and observations is available. The details of this process are presented in Chapters 4 and 5.

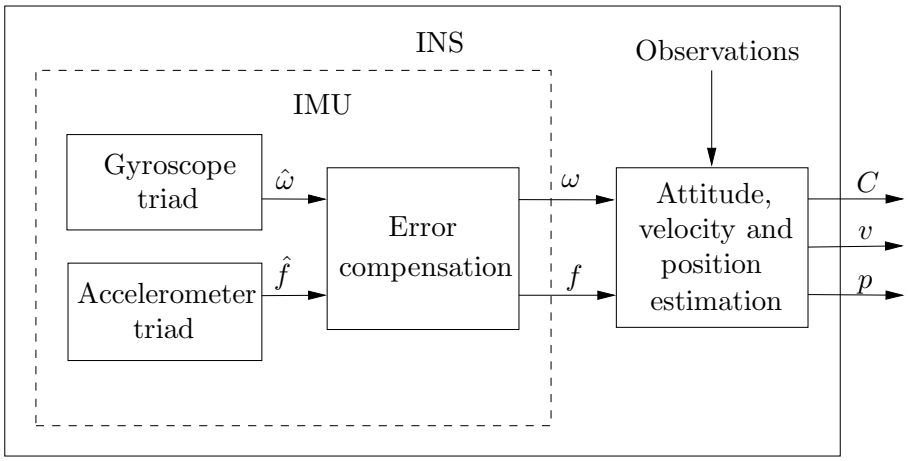

Figure 2.1: A block diagram of an inertial navigation system.

2.2 Frames of reference

Let us now specify the coordinate frames necessary for inertial navigation on Earth. The specifications are much the same as those presented in [2, 4] with the specification of the inertial frame adopted from [59]. It should be noted that the the specified frame fixed to the Earth agrees with the "Earth-Centered Inertial" (ECI) frame rather than the "Earth-Centered Earth-Fixed" (ECEF) frame [4]. Figure 2.2 presents an illustration of the specified coordinate frames.

Specification 1 *An inertial frame (i-frame) of reference is free of rotation and in free fall. In the i-frame, the laws regarding the preservation of linear and angular momentum take their simplest possible form.*

Specification 2 *An (e-frame) of reference is a Cartesian right-handed coordinate system, whose origin is located at the center of the Earth and the basis (e_{ex},e_{ey},e_{ez}) is fixed with respect to distant stars in such a way that e_{ez} coincides with the Earth's polar axis.*

Specification 3 *A local geographic frame (g-frame) of reference is a Cartesian right-handed coordinate system, whose origin and basis (e_{gx},e_{gy},e_{gz}) are fixed with respect to the Earth in such a way that e_{gz} coincides with the local vertical upward direction.*

Specification 4 *A body frame (b-frame) of reference is a Cartesian right-handed coordinate system, whose origin and basis (e_{bx},e_{by},e_{bz}) are fixed with respect to the body of the object.*

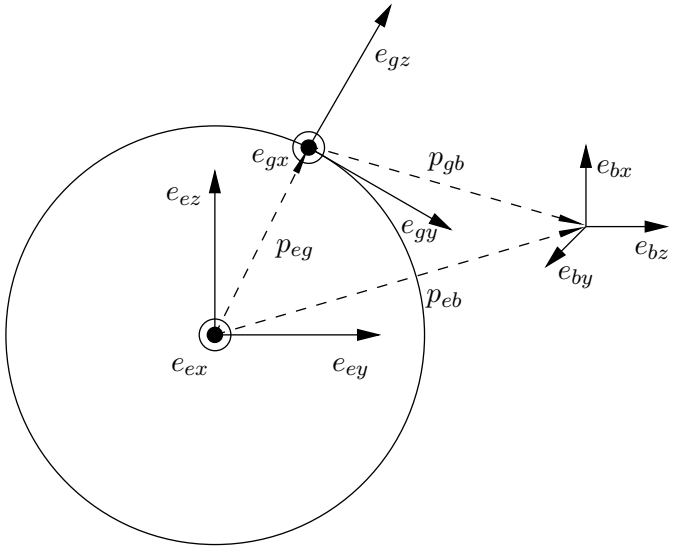

Figure 2.2: A graphical representation of the reference frames used. Dots represent vectors towards the reader.

The notation is here introduced with an example. Vector p_{eb} seen in Figure 2.2 represents the position of the b-frame with respect to the e-frame. As indicated by the figure, it holds that

$$p_{eb} = p_{eg} + p_{gb} \tag{2.1}$$

whenever the three vectors are all represented in the same coordinate frame. Moreover, the unit vector e_{ex}, for example, indicates the x-axis of the e-frame. When necessary, an additional upper index can be used to clarify the frame, where the vector in question is represented. Thus, it holds that

$$\begin{bmatrix} e_{jx}^j & e_{jy}^j & e_{jz}^j \end{bmatrix} = I \ \ \forall \ \ j \in \{i, e, g, b\}\,,$$

where $I \in \mathbb{R}^{3\times 3}$ is the identity matrix.

For a reader familiar with inertial navigation systems, it helps to think that the output of an ideal (6DOF) IMU fixed with respect to the i-frame is always zero. Or vice versa, a non-zero measurement of the same IMU

is always due to the acceleration and/or rotation of the IMU with respect to the i-frame. Correspondingly, the measurements of the body-mounted gyroscope and accelerometer triads are denoted by ω_{ib}^b and f_{ib}^b. Notice that the e-frame – whether rotating with the Earth (ECEF) or not (ECI) – is merely an approximation of the i-frame, which neglects many of the rotations and accelerations related to planetary motions [60,61]. However, within the accuracy of the majority of INSs, it is sufficient to assume that

$$a_{eb} = f_{ib} + g \tag{2.2}$$
$$\omega_{eb} = \omega_{ib}, \tag{2.3}$$

where a is the acceleration and g is the *gravitational acceleration* [2, 4]. The addition of the gravitational acceleration g in the *navigation equation* (2.2) [2] arises from the fact that the i-frame is in free fall while the e-frame is not.

2.3 Rotations between frames of reference

The rotations between the defined reference frames are carried out by the rotation matrices

$$C \in \mathbb{R}^{3\times 3} \ : \ CC^T = C^TC = I \ , \ \det(C) = 1, \tag{2.4}$$

where it hold for the inverse of C that $C^{-1} = C^T$, upper index T denotes the transpose and I is the identity matrix of the corresponding dimension. Hereafter, rotation matrices C will be called *direction cosine matrices* (DCM) owing to their geometrical interpretation [2]. The set of all DCMs form the *special orthogonal group* SO(3) [62]. In this thesis, DCMs are exclusively used to represent the attitude.

The lower index g of DCM C_g^e indicates the initial frame and the upper index e, the target frame of the rotation C_g^e. We can, for example, write (2.1) as

$$p_{eb}^e = C_g^e(p_{eg}^g + p_{gb}^g). \tag{2.5}$$

However, for clarity (2.5) will hereafter be written as

$$p_{eb} = C_g^e(p_{eg} + p_{gb}), \tag{2.6}$$

since it is clear from the context that p_{eg} and p_{gb} must be given in the g-frame, and as a result, the representation of p_{eb} in the e-frame is obtained.

Now, let us determine the connection between the attitude C_b^i and the measured angular velocity ω_{ib}^b. The idea behind the following derivation

is based on the treatment presented in [63]. Consider concentric i- and b-frames, where the b-frame rotates with angular velocity ω_{ib} with respect to the i-frame. Then, consider an arbitrary point p fixed with respect to the i-frame. An illustration of the situation is presented in Figure 2.3. With respect to the b-frame, the velocity of p is

$$v^b = \frac{d}{dt}\left[C_i^b p^i\right] = \dot{C}_i^b p^i + C_i^b \dot{p}^i = \dot{C}_i^b p^i. \tag{2.7}$$

On the other hand, since p^i is a constant, p^b must move on the surface of a sphere of radius $\|p\|_2$, where $\|\cdot\|_2$ represents the 2-norm of $\cdot$. Thus, its velocity v^b lies on the tangent plane of the surface and can be written as

$$v^b = \omega_{bi}^b \times p^b = -\omega_{ib}^b \times C_i^b p^i, \tag{2.8}$$

since $\omega_{bi}^b = -\omega_{ib}^b$ [64].

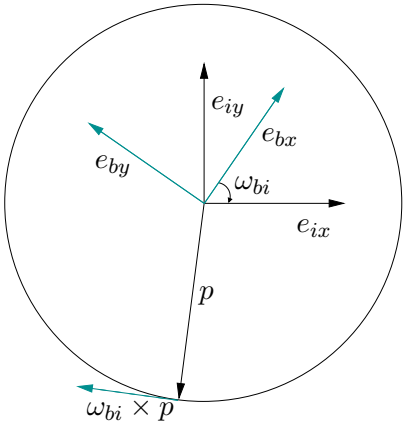

Figure 2.3: An illustration of the tangential velocity of an arbitrary point p caused by ω_{bi}.

By equating (2.7) and (2.8), we obtain

$$\dot{C}_i^b p^i = -\omega_{ib}^b \times C_i^b p^i. \tag{2.9}$$

Since (2.9) has to hold for any p^i, it follows that

$$\dot{C}_i^b = -[\omega_{ib}^b]_\times C_i^b \tag{2.10}$$

where

$$[\omega_{ib}^b]_\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{2.11}$$

is the skew-symmetric form of ω_{ib}^b [2]. Finally, by transposing (2.10), we obtain

$$\dot{C}_b^i = C_b^i [\omega_{ib}^b]_\times, \tag{2.12}$$

since $[\omega_{ib}^b]_\times^T = -[\omega_{ib}^b]_\times$.

2.4 Motion equations

We now turn to the equations which relate the quantities of interest, p_{gb} and v_{gb}, (i.e. the position and velocity of the b-frame with respect to the g-frame) to the measured quantities f_{ib} and ω_{ib}. It should be noted that the derivation does not follow any particular source. However, the obtained results are obviously in agreement with the references, such as [2, 4]. Let us start by differentiating (2.6) with respect to time, which yields

$$\begin{aligned} \dot{p}_{eb} &= \dot{C}_g^e(p_{eg} + p_{gb}) + C_g^e(\dot{p}_{eg} + \dot{p}_{gb}) \\ v_{eb} &= C_g^e[\omega_{eg} \times (p_{eg} + p_{gb}) + v_{gb}]. \end{aligned} \tag{2.13}$$

In (2.13) we have applied (2.12) and the fact that $\dot{p}_{eg}^g = 0$. Now, by differentiating (2.13) with respect to time, we obtain

$$\begin{aligned} \dot{v}_{eb} &= \dot{C}_g^e[\omega_{eg} \times (p_{eg} + p_{gb}) + v_{gb}] \\ &+ C_g^e[\dot{\omega}_{eg} \times (p_{eg} + p_{gb}) + \omega_{eg} \times (\dot{p}_{eg} + \dot{p}_{gb}) + \dot{v}_{gb}] \\ a_{eb} &= C_g^e[\omega_{eg} \times \omega_{eg} \times (p_{eg} + p_{gb}) + 2\omega_{eg} \times v_{gb} + a_{gb}] \end{aligned} \tag{2.14}$$

since $\dot{\omega}_{ie}$ (the angular acceleration of the Earth) is zero [2]. By multiplying (2.14) from the left by C_e^g and substituting the navigation equation

$$a_{eb} = C_b^g f_{ib} + g,$$

we obtain the equation

$$a_{gb} = C_b^g f_{ib} - 2\omega_{eg} \times v_{gb} + g - \omega_{eg} \times \omega_{eg} \times (p_{eg} + p_{gb}). \tag{2.15}$$

The last term of (2.15) is recognized as the centripetal acceleration caused by the rotation of the Earth. Like the gravitational acceleration, it is also a function of $p_{eg} + p_{gb} = p_{eb}$. Hence, it is often convenient to pack them into a single term

$$g_l = g - \omega_{eg} \times \omega_{eg} \times p_{eb}, \tag{2.16}$$

denoted as the *local gravity vector* [2]. Then, (2.15) becomes

$$a_{gb} = C_b^g f_{ib} - 2\omega_{eg} \times v_{gb} + g_l, \tag{2.17}$$

where the DCM C_b^g can be resolved from the equation

$$\dot{C}_b^g = C_b^g[\omega_{gb}^b]_\times = C_b^g[\omega_{ib}^b - C_g^b \omega_{eg}^g]_\times. \tag{2.18}$$

2.5 Simplified motion equations

The problem does not necessarily have to be posed as it is in (2.17) and (2.18). In particular, when the time period is sufficiently short, it might sometimes be justifiable to treat the local geographic frame as an inertial frame. Then, it holds that $\omega_{eg} = 0$ leaving the equations

$$\begin{aligned} a_{g\hat{b}} &= C^g_{\hat{b}} f^b_{ib} + \hat{g}_l && (2.19) \\ \dot{C}^g_{\hat{b}} &= C^g_{\hat{b}} [\omega^b_{ib}]_\times, && (2.20) \end{aligned}$$

where $\hat{b}$ denotes the perturbed b-frame introduced by the simplification. Notice that the term $\hat{g}_l$ only accounts for the error in the value of g_l due to the position error δp caused by the simplification. As can be verified from (2.2), the effect of the gravity is built into the measurement f^b_{ib}. It is also important to realize that this simplification has no effect on the measurements f^b_{ib} and ω^b_{ib}.

The simplified equations (2.19) and (2.20) define the concept of "short-term inertial navigation" introduced in Chapter 1. For the defined concept to be useful in practice, it must still be possible to determine when it is feasible to use the simplified motion equations. Obviously, the answer depends on the "error budget", i.e. how much accuracy can one afford to lose because of the simplification [2]. For this purpose, we shall now derive the equations characterizing the error caused by the simplification. Let us first investigate the dynamics of the attitude error $\delta C = C^g_{\hat{b}} C^b_g$:

$$\begin{aligned} \delta\dot{C} &= \dot{C}^g_{\hat{b}} C^b_g + C^g_{\hat{b}} \dot{C}^b_g \\ &= C^g_{\hat{b}} [\omega^b_{ib}]_\times C^b_g - C^g_{\hat{b}} [\omega^b_{ib} - C^b_g \omega^g_{eg}]_\times C^b_g \\ &= C^g_{\hat{b}} [\omega^b_{ib}]_\times C^b_g - C^g_{\hat{b}} [\omega^b_{ib}]_\times C^b_g + C^g_{\hat{b}} [C^b_g \omega^g_{eg}]_\times C^b_g \\ &= C^g_{\hat{b}} [C^{\hat{b}}_g \delta C \omega^g_{eg}]_\times C^{\hat{b}}_g \delta C && (2.21) \end{aligned}$$

In (2.21), we have used equation (2.18) and written $\delta\dot{C}$ in terms of $C^g_{\hat{b}}$ (cf. the Psi- and Phi-angle error models [32]). The position error $\delta p = p_{g\hat{b}} - p_{gb}$ is determined by ODE

$$\begin{aligned} \ddot{\delta p} &= \ddot{p}_{g\hat{b}} - \ddot{p}_{gb} \\ &= \left(C^g_{\hat{b}} - C^g_b \right) f_{ib} + 2[\omega^g_{eg}]_\times \dot{p}_{gb} + \delta g_l \\ &= \left(I - \delta C^T \right) C^g_{\hat{b}} f_{ib} + 2[\omega^g_{eg}]_\times (v_{g\hat{b}} - \dot{\delta p}) + \delta g_l, && (2.22) \end{aligned}$$

where $(\ddot{})$ denotes the second time derivative and the error in the local gravity vector $\delta g_l = \hat{g}_l - g_l$ is caused by non-zero position error δp. Notice that

the error equations depend on the true motion of the object. Consequently, in the absence of more accurate information, we only have access to the simplification errors with respect to the estimated motion of the object. Also, an a priori estimation of the simplification error always requires an a priori knowledge of the motion.

Obviously, the behavior of the simplification error depends greatly on the provided boundary conditions. Let us emphasize that the "traditional" error analysis based on the evolution of the errors with zero initial values only applies when the motion equations are solved as an IVP without additional observations. In general, this result can only be used as a rough – usually rather pessimistic – estimate of the resulting errors. The error estimation should be based on the same set of observations as were used to resolve the state estimates in the first place. So, the error analysis requires us to solve two fixed-interval smoothing problems: one for the state estimates and one for the simplification errors. This thesis will provide the tools necessary to fulfill both tasks.

Simplifications of (2.21) and (2.22) are possible when the duration of the navigation period is particularly short. In these cases, with suitable boundary conditions, $\delta C \approx I$. If the change in the position of the object is also known to be small, we can further assume that $\delta g_l \approx 0$. In such a case, it holds that

$$\ddot{\delta p} \approx 2[\omega^g_{eg}]_\times (v_{g\hat{b}} - \dot{\delta p}). \tag{2.23}$$

2.6 Dynamics model

To a great extent, this thesis utilises simplified motion equations, where it is also assumed that g_l is a constant. The only exception to this is the calibration method presented in Chapter 3, where it is beneficial to work with the detailed motion equations. Otherwise, as the simplified motion equations concern only two reference frames, the notation for the equations can be more elegantly written in the form

$$\ddot{p} = Cf + g \tag{2.24}$$
$$\dot{C} = C[\omega]_\times, \tag{2.25}$$

where C always represents rotation from the body frame to the local geographic frame. By default, v and p represent the velocity and position of the body resolved in the local geographic frame. Obviously, we can also write (2.24) and (2.25) in the form expected by the dynamics model, which

yields

$$\begin{bmatrix} \dot{c}_1 \\ \dot{c}_2 \\ \dot{c}_3 \\ \dot{v} \\ \dot{p} \end{bmatrix} = \begin{bmatrix} 0 & \omega_z I & -\omega_y I & 0 & 0 \\ -\omega_z I & 0 & \omega_x I & 0 & 0 \\ \omega_y I & -\omega_x I & 0 & 0 & 0 \\ f_x I & f_y I & f_z I & 0 & 0 \\ 0 & 0 & 0 & I & 0 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ c_3 \\ v \\ p \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ g \\ 0 \end{bmatrix}, \tag{2.26}$$

where $C = \begin{bmatrix} c_1 & c_2 & c_3 \end{bmatrix}$.

The value of the gravitational acceleration g in (2.24) and (2.26) depends on the latitude L of the local geographic frame and the average height H of the object with respect to the sea-level. Then, in units of $\mathrm{m/s^2}$,

$$g = -\frac{9.780318 \times (1 + 5.3024 \times 10^{-3} \sin^2 L - 5.9 \times 10^{-6} \sin^2 2L)}{(1 + H/R_0)^2} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \tag{2.27}$$

where $R_0 = 6.371 \times 10^6$ m is the mean radius of the Earth [2]. While it might seem unnecessary to use such a specific value for g, it should be noted that there is no harm in doing so whenever the latitude and height are known. It should also be noted that it is not always necessary to provide a value for g. With the proposed solution methods, the gravitational acceleration can easily be treated as an unknown.

The orthogonality constraint

While the attitude equation is often presented in a deceptively simple form (2.25), one should bear in mind that the result is a DCM. The attitude of the object is the solution of the following problem. Given $\omega : \mathbb{R} \to \mathbb{R}^3$, find $C : \mathbb{R} \to \mathbb{R}^{3\times 3}$ such that

$$\dot{C} = C[\omega]_\times \ \ \forall \ \ t \in [T_0, T_1] \tag{2.28}$$

holds, where

$$[\omega]_\times = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} \tag{2.29}$$

and

$$C^T C = I \ \ \forall \ \ t \in [T_0, T_1]\,. \tag{2.30}$$

The *orthogonality constraint* (2.30) generates (2.28) a *differential-algebraic equation* (DAE) [6]. According to [6,65], there are four commonly-employed strategies to solve DAEs of the form (2.28):

(i) Rephrase the DAE as an ODE whose solution satisfies the orthogonality constraint.

(ii) Solve the DAE using a suitable numerical method. Then, apply a post-stabilization step, which brings the solution closer to satisfying the orthogonality constraint.

(iii) Find a numerical solution method for the DAE, which automatically satisfies the orthogonality constraint.

(iv) Rephrase the problem as an *orthogonal Procrustes* (also known as *Wahba*'s) problem [40, 65, 66].

Let us point out that the condition $\|f^n\|_2 = \left\|f^b\right\|_2$ is satisfied for all f only if the DCM used to carry out the coordinate transformation is orthogonal. As seen from (2.24), an *orthogonality error* will inevitably lead to a reduction in accuracy. In this thesis, orthogonality errors are measured with

$$\delta_C = \left\|CC^T - I\right\|_F, \tag{2.31}$$

where $\|\cdot\|_F$ refers to the Frobenius norm, which can be evaluated as

$$\|A\|_F = \sqrt{\mathrm{Tr}(AA^T)} = \sqrt{\mathrm{Tr}(A^TA)} \tag{2.32}$$

for an arbitrary real-valued matrix A. In (2.32), 'Tr' refers to the *trace*, the sum of the diagonal elements of a matrix [66]. Because a typical inertial navigation system requires at least one coordinate transformation it is, to some extent, subject to orthogonality errors. Of course, the method chosen for attitude representation affects the behavior of the errors, but within any realistic INS, δ_C is always non-zero. However, in a properly designed inertial navigation system, the accuracy is not limited by the orthogonality error.

With the traditional (recursive) solution methods, relatively little effort is usually needed to obtain sufficiently small orthogonality errors. In inertial navigation, a traditional approach is based on method (ii), where the post-stabilization step is applied from time to time [2]. This way, the orthogonality of the computed DCMs can be easily controlled and the post-stabilization step can take place before any problems occur. A number of iterative orthogonalization techniques suitable for the post-stabilization step are presented in [67]. The nth iterate ($C_{i,n}$) of one such technique is

$$C_{i,n} = \frac{1}{2}(C_{i,n-1}^{-T} + C_{i,n-1}) \tag{2.33}$$

and the iteration is continued until δ_C is below a predetermined tolerance. The tolerance is application-dependent and simulations are usually used to obtain a suitable value for it.

This thesis focuses on non-recursive solution methods, for which method (ii) is not sufficient as such. The basic reason for this is simple: with non-recursive solution methods, intolerable orthogonality errors may occur before

the post-stabilization step can be applied. Obviously, a chosen orthogonalization technique can then be separately applied to each obtained DCM, but the damage has already been done. In the case of (2.26), for example, the velocity and position estimates will be affected by the orthogonality error regardless of whether the post-stabilization step takes place or not. Therefore, in this thesis, the orthogonality constraint is taken into account with method (iii) and/or with so-called pseudo-observations, which results in an orthogonalization strategy similar to method (ii). With these solution methods, the orthogonality errors can be kept low enough for their effects to be negligible on the obtained results.

2.7 Summary

The motion equations and the dynamics model (2.26) described in this chapter are the basis on which the following chapters are built. In the next chapter, the detailed motion equations (2.15 and 2.18) are used to derive the required reference measurements. Otherwise, the analysis will be based on on the simplified motion equations (2.24 and 2.25). On a broader scale, the contents of the subsequent chapters are linked – whenever applicable – to Figure 2.1, which will aid the reader to grasp the bigger picture.

A reader with a physics background might think that a model referred to as a dynamics model should be realized as an equation describing the connections between the forces and torques exerted on a (rigid) body and the respective kinemates ω, v. Such equations are often referred to as the Newton-Euler or Euler-Lagrange equations of motion [62]. However, since the sensors used in inertial navigation provides the user with access to specific forces and angular velocities, there is no need to write the equation in terms of the actual causes of the motion. Therefore, equations with the same purpose, i.e. the ability to determine the motion of an object, can be written in a form such as (2.26). Hence the name, dynamics model.

When necessary, similar motion equations can be written in terms of other attitude representations such as quaternions or Euler angles. With quaternions, the orthogonality constraint will be replaced by a simpler one requiring the length of the quaternion to be one. With Euler angles, the main limitation is that special treatment is required whenever the "pitch" angle approaches $\pm\pi/2$ radians [2]. However, both of these choices come with the cost of a somewhat more complicated realization of the necessary coordinate transformations. Strictly in this sense, the method chosen here for attitude representation is hard to beat.

Chapter 3

Calibration of IMUs

This chapter discusses the calibration of IMUs. With reference to Figure 2.1 (page 14) this chapter is about the error compensation block between the unprocessed raw measurement data $(\hat{\omega}, \hat{f})$ and the actual measurement data (ω, f). While the main ideas of this chapter are based on the publication P1, this chapter presents a more rigorous derivation of the reference signals. Moreover, the treatment will focus on *small-sized*[1] consumer-grade IMUs, and will also present a new *online calibration* method. The online calibration method presented in section 3.6 exploits the measurements obtained during the actual event, and enables the modelling of the effects of operating conditions – such as temperature – on the individual sensors within the IMU.

By *calibration*, we mean the process of comparing the outputs of the gyroscope and accelerometer triads against a set of *reference measurements*. The calibration process yields a *measurement model*, which is defined as a *bijective function* between the set of unprocessed raw measurements $(\hat{\omega}, \hat{f})$ and the actual measurements (ω, f). An integral part of this measurement model is the sensor-error model, which is used to *compensate* for a number of deterministic errors within the raw measurements of an IMU. In the case of an uncalibrated IMU, a generic measurement model (usually without error compensation capabilities) can be constructed based on the data sheets of the particular sensors. Otherwise, the instance of the measurement model which is obtained is unique for each calibrated sensor triad.

[1] A "small-sized IMU" refers to an IMU, where it is assumed that all three accelerometers measure the specific force exerted on a single point. The lack of this property is commonly referred to as the "size effect", which is a reality for any IMU containing more than a single proof mass sensing the specific force [2]. In practical terms, the premise behind a small-sized IMU is that the size effect does not significantly limit the performance of the IMU.

The starting point for this chapter is the classic multi-position calibration method [2]. It is based on the simple yet elegant idea of observing the output of the IMU while it is kept stationary at different attitudes with respect to the gravitational acceleration and the angular velocity of the Earth [2–4, 13]. Given the relatively large magnitude of the gravitational acceleration ($\|g\|_2 \approx 9.80$ m/s^2), this method can be applied to almost any practical accelerometer. However, the angular velocity of the Earth, $\Omega_E = 7.292115 \times 10^{-5}$ rad/s, limits the use of this method to only high-end gyroscopes which are sensitive and stable enough to reliably capture the distinct components of such a faint angular velocity. In addition to this, the basic problem of the classical multi-position calibration method is that it relies on *extrapolation* whenever the dynamic range of the calibrated sensors exceeds the magnitudes of the reference measurements used.

This thesis augments the classic multi-position calibration method with a rate table (see Figure 3.1), which provides us with reference measurements for the whole dynamic range of the selected gyroscopes and accelerometers. Thus, unlike the classic, extrapolatory, multi-position calibration method, the method proposed here relies on *interpolation*. The basic principle of this method of calibration is to rotate the IMU with $N_r \in \mathbb{N}$ known rotation rates in $N_a \in \mathbb{N}$ known attitudes. For ease of notation it is assumed that the number of rotation rates N_r is the same for all N_a attitudes. The reference specific force at the jth rotation rate of the ith attitude is denoted by $f_{i,j} \in \mathbb{R}^3$, the reference angular velocity by $\omega_{i,j} \in \mathbb{R}^3$, and the rotation rate of the rate table by $\Omega_{i,j} \in \mathbb{R}$, where $i \in [1, \ldots, N_a]$ and $j \in [1, \ldots, N_r]$. The reference signals for the accelerometers are built on the *centripetal accelerations* caused by the rate table. The respective rotation radii $r_i \in \mathbb{R}_+$ ($i \in [1, \ldots, N_a]$) will be treated as unknowns, which obviates the need to determine them manually.

One of the key points of this method of calibration is the ability to customize how the reference measurements ($f_{i,j}$, $\omega_{i,j}$) are selected when seeking out the measurement model. If the data collected during the calibration process can be accessed once the data related to the actual event is processed, the measurement model can be resolved using the information gained about the actual range of specific forces and angular rates. The reason for this is that it allows the user to enhance the performance of the sensors within a specified dynamic range. The goal of the online calibration method is essentially the same, and it aims to reach it by exploiting the measurements obtained during the actual event.

3.1 The rate table

This section discusses the properties and premises related to the rate table, which is used to carry out the calibration procedure. Attention is paid to ensure that the specifications for the rate table are in balance with the need for a calibration method which is economically feasible[2] .

The key property of the rate table is that the *driving motor used is able to maintain a constant angular rate* $\Omega_{i,j}$. It must be emphasized that $\Omega_{i,j}$ need not be particularly repeatable, but the variations of $\Omega_{i,j}$ should be small once the motor has achieved the desired speed. This is because the reference measurements are based on the assumption that $\dot{\Omega}_{i,j} = 0$. *Only the averages of the observed quantities over* $N_\theta \in \mathbb{N}$ *full revolutions of the rate table are used during calibration.* There are several important reasons for this (see sections 3.2 – 3.3 for details):

- the average angular velocity $\overline{\Omega}_{i,j}$ of the rate table can be determined with high accuracy [68]
- the effects of the vibration and noise present in the system are significantly reduced [68]
- the accuracy of the resulting measurement model is not dependent on the dynamic performance of the sensors
- the reference measurements are more accurate.

It is important to understand that *no measurements obtained while the rate table is stationary are used.* This ensures that only those reference measurements which are averaged over full revolutions of the rate table (and have therefore reached their maximum accuracy) are used

Now, let us specify a few additional reference frames, which will help clarify the contents of this chapter.

Specification 5 *A rotation frame (r-frame) of reference is a Cartesian right-handed coordinate system, whose origin coincides with that of the body frame. The basis* (e_{rx},e_{ry},e_{rz}) *of the r-frame is fixed with respect to the rate table as follows:* $e_{rz} \parallel e_{gz}$ *are parallel with the rotation axis of the rate table, and* e_{rx} *points towards the rotation axis.*

Specification 6 *A sensor frame (s-frame) of reference is a coordinate system*[3]*, whose origin coincides with that of the body frame. The basis* (e_{sx},e_{sy}, e_{sz}) *of the s-frame is determined by the physical alignment of the sensors*

[2] Automation of the calibration procedure – which is a necessity for any economically feasible calibration method – is a question, which is not addressed here.

[3] The *s*-frame is generally *not* a Cartesian coordinate system.

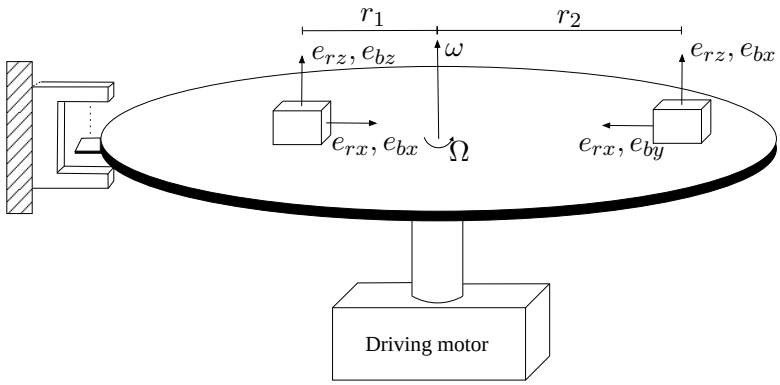

Figure 3.1: A schematic drawing of the rate table used to calibrate the sensors.

within the sensor triad in question. The axes of the s-frame are assumed to be linearly independent.

Figure 3.1 presents a schematic drawing of the rate table with attitudes b_1 and b_2 (see below) and the principle behind the optical fork sensor used to determine $\Omega_{i,j}$. The radii r_1 and r_2 denote the shortest distances of their respective b-frames from the rotation axis. The radius is assumed to be different for each of the attitudes used, but any possible dependency between r_i and $\Omega_{i,j}$ is ignored[4]. Note that the optical fork sensor provides a simple way of determining precisely when the IMU has completed a full revolution.

Based on these premises, and assuming that e_{gx} indicates north, it follows that

$$p^g_{eg} = \begin{bmatrix} 0 & 0 & R_0 + H \end{bmatrix}^T \tag{3.1}$$

$$p^r_{gb} = \begin{bmatrix} -r_i & 0 & 0 \end{bmatrix}^T \tag{3.2}$$

$$v^r_{gb} = \begin{bmatrix} 0 & -\Omega_{i,j} r_i & 0 \end{bmatrix}^T \tag{3.3}$$

$$a^r_{gb} = \begin{bmatrix} \Omega^2_{i,j} r_i & 0 & 0 \end{bmatrix}^T \tag{3.4}$$

$$\omega^r_{gb} = \begin{bmatrix} 0 & 0 & \Omega_{i,j} \end{bmatrix}^T \tag{3.5}$$

$$\omega^g_{eg} = \Omega_E \begin{bmatrix} \cos L & 0 & \sin L \end{bmatrix}^T, \tag{3.6}$$

where $\Omega_E = 7.292115 \times 10^{-5}$ rad/s is the rotation rate of the Earth [2].

[4]In an *open-loop accelerometer*, the measured acceleration is proportional to the displacement of the proof mass. If the proof mass moves significantly in the direction of the radius, one should change the rotation radius accordingly. However, the displacements of the proof mass within small-sized open-loop accelerometers are often negligible and *closed-loop accelerometers* in particular, are virtually insensitive to this error. [2,68]

Moreover, we can write

$$C_g^r = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.7}$$

where θ is the rotation angle of the rate table. Note that the assumption about the direction of e_{gx} was only made to simplify the appearance of equation (3.6); in the end – as the symmetry of the situation suggests – the only requirement is that $e_{rz} \parallel e_{gz}$.

Of course, in order to cover all six degrees of freedom, the IMU has to be rotated in more than one attitude. In order to characterize different attitudes of the IMU, constant DCMs of the form $C_r^{b_i}$ (where b_i denotes the ith attitude of the body frame) are used, thus rotating the reference measurements from the r-frame into the b-frame in question. The minimum amount of different attitudes needed for a 6 DOF IMU is three, which allows the axes of the b-frame to be parallel to each axis of the r-frame once. The corresponding DCMs are

$$C_r^{b_1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad C_r^{b_2} = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \quad C_r^{b_3} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

In P1, N_a is equal to 6 with each axis of the b-frame pointing once to each positive and negative axis of the r-frame. The remaining three DCMs are

$$C_r^{b_4} = -\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad C_r^{b_5} = -\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad C_r^{b_6} = -\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

This method of calibration in no way imposes any limit on the number of attitudes N_a. Nevertheless, the above six attitudes already cover the most extreme cases. Therefore, any potential benefits of additional attitudes are mostly due to the inherent errors in $C_r^{b_i}$ as compared to the realized attitudes. The remaining analysis is based on the assumption that the DCMs for the attitudes are accurate.

3.2 Reference measurements

In an ideal measurement model, the calibrated performance of the IMU depends solely on the accuracy of the reference measurements. Despite the inherent shortcomings of the real-world measurement model, the accuracy of the reference measurements still determines an upper limit for the calibrated

performance of the IMU. Therefore, the reference measurements are derived from the detailed motion equations (2.15 and 2.18) even though the overall quality of the sensors may not quite be up to the job of reliably sensing all the included terms. The advantage is that this approach allows the errors caused by the optional simplification of the reference measurements to be easily and accurately determined. As will be explained, the looser constraints regarding the alignment of the rate table are the main reason why the simplified motion equations are also relevant.

Let us now turn to the derivation for the reference specific force. The first step is to resolve the momentary value of the specific force f_{ib}^r from (2.15), which yields

$$f_{ib}^r = a_{gb} - g + C_g^r \left[2[\omega_{eg}]_\times C_r^g v_{gb} + [\omega_{eg}]_\times [\omega_{eg}]_\times (p_{eg} + C_r^g p_{gb})\right]. \tag{3.8}$$

Then, by substituting (3.1) – (3.7) into (3.8), we obtain

$$\begin{aligned} f_{i,j}^r &= \begin{bmatrix} \Omega_{i,j}^2 r_i \\ 0 \\ g_\perp \end{bmatrix} + 2\Omega_E \Omega_{i,j} r_i \begin{bmatrix} \sin L \\ 0 \\ -\cos L \cos\theta \end{bmatrix} + \Omega_E^2 r_i \begin{bmatrix} \sin^2 L \cos^2\theta + \sin^2\theta \\ \cos^2 L \sin\theta \cos\theta \\ \sin L \cos L \cos\theta \end{bmatrix} \\ &+ \Omega_E^2 (R_0 + H) \begin{bmatrix} \sin L \cos L \cos\theta \\ -\sin L \cos L \sin\theta \\ -\cos^2 L \end{bmatrix}, \end{aligned} \tag{3.9}$$

where $f_{i,j}^r$ represents the reference specific force caused by $\Omega_{i,j}$ and $g_\perp = g_z$ refers to the component of the gravitational acceleration perpendicular to the rate table (see Figure 3.2). Based on(2.18), it holds for ω_{ib}^r that

$$\omega_{ib}^r = \omega_{gb} + C_g^r \omega_{eg}. \tag{3.10}$$

Then, the substitution of (3.5) – (3.7) into (3.10) yields

$$\omega_{i,j}^r = \begin{bmatrix} \Omega_E \cos L \cos\theta \\ -\Omega_E \cos L \sin\theta \\ \Omega_{i,j} + \Omega_E \sin L \end{bmatrix}, \tag{3.11}$$

where $\omega_{i,j}^r$ represents the reference angular velocity caused by $\Omega_{i,j}$.

Instead of the momentary values (3.9) and (3.11), the calibration is based on the average values of $f_{i,j}^r$ and $\omega_{i,j}^r$ over N_θ full revolutions of the rate table. Since a number of the terms included in (3.9) and (3.11) are proportional to $\sin\theta$, $\cos\theta$, or $\sin\theta\cos\theta$, the respective terms vanish when averaged over full revolutions of the rate table. Moreover, the average of $\sin^2\theta$ and $\cos^2\theta$ over full revolutions of the rate table is 1/2. Hence, for all initial angles $\theta_0 \in \mathbb{R}$, we obtain

$$\overline{f}_{i,j}^r = \begin{bmatrix} \overline{\Omega}_{i,j}^2 r_i \\ 0 \\ g_\perp \end{bmatrix} + 2\overline{\Omega}_{i,j} \Omega_E r_i \begin{bmatrix} \sin L \\ 0 \\ 0 \end{bmatrix} + \frac{1}{2}\Omega_E^2 \begin{bmatrix} r_i(1 + \sin^2 L) \\ 0 \\ -(R_0 + H) \end{bmatrix} \tag{3.12}$$

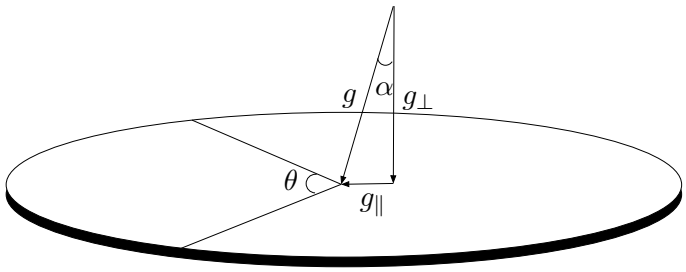

Figure 3.2: Components of the gravitational acceleration g perpendicular ($g_\perp$) and parallel ($g_\parallel$) to the plane of the rate table.

for the specific force. For the angular velocity, we obtain

$$\overline{\omega}^r_{i,j} = \begin{bmatrix} 0 \\ 0 \\ \overline{\Omega}_{i,j} + \Omega_E \sin L \end{bmatrix}. \tag{3.13}$$

Simplified reference measurements

The derived equations for $\overline{f}^r_{i,j}$ and $\overline{\omega}^r_{i,j}$ make it easy to determine when it is reasonable to simplify the reference measurements. In addition, Ω_E can easily be taken into account as long as the assumptions made in section 3.1 are met. However, this is not always so, especially when considering the alignment of the rotation frame with respect to the local geographic frame. Whenever there is a significant alignment error between the rate table and the g-frame, the presented equations for $\overline{f}^r_{i,j}$ and $\overline{\omega}^r_{i,j}$ no longer hold true, so it makes no sense to use them in detail. Nevertheless, the reference signals obtained when neglecting Ω_E are still useful, as they only depend on the alignment of the g-frame via the direction of gravitational acceleration.

In order to simplify the treatment, let us model the alignment error by keeping C^r_g unchanged and decomposing the gravitational acceleration into $g_\perp \in \mathbb{R}$, for the component perpendicular to the rate table, and $g_\parallel \in \mathbb{R}$ for the component in the plane of the rate table. This corresponds to the situation, where $e_{rz} \parallel e_{gz}$, but the gravitational acceleration is no longer parallel to e_{gz}. Figure 3.2 illustrates this. In this case, the average of $g_\parallel$ over N_θ full revolutions is

$$\overline{g}_\parallel = \frac{\|g\|_2 \sin\alpha}{2\pi N_\theta} \int_{\theta_0}^{\theta_0 + 2\pi N_\theta} \sin\theta \, d\theta = 0. \tag{3.14}$$

Therefore, the error in the gravitational acceleration is determined solely by the accuracy of $g_\perp$. On the other hand, alignment error $|\alpha| < \pi/2$ results

in

$$\delta g_{\perp} = \|g\|_2 (1 - \cos\alpha). \tag{3.15}$$

In such a case, even an easily-observable alignment error of 0.1° will only cause an acceptable 1.5 μg error in the value of $g_{\perp}$. The simplified reference measurements appear in the form

$$\overline{f}^r_{i,j} = \begin{bmatrix} \overline{\Omega}^2_{i,j} r_i & 0 & g_{\perp} \end{bmatrix}^T \tag{3.16}$$

$$\overline{\omega}^r_{i,j} = \begin{bmatrix} 0 & 0 & \overline{\Omega}_{i,j} \end{bmatrix}^T, \tag{3.17}$$

as used in P1.

3.3 Error analysis

This section offers an analysis of the effect of an erroneous rotation rate on the reference signals. Due to the symmetry of the situation, the analysis will be restricted to only positive rotation rates, which are denoted by $\Omega \in \mathbb{R}_+$. Let us define the measured rotation rate as

$$\hat{\Omega} = \overline{\Omega} + \delta\Omega, \tag{3.18}$$

where $\hat{\Omega}$ and $\delta\Omega$ are functions of θ. For the average of $\hat{\Omega}$, we obtain the error bounds

$$\hat{\overline{\Omega}} = \overline{\Omega} \pm \left|\overline{\delta\Omega}\right|, \tag{3.19}$$

and the next task is to estimate $\overline{\delta\Omega}$.

In the rate table, $\hat{\overline{\Omega}}$ is obtained with an analog optical fork sensor, as shown in Figure 3.1. The output of the optical fork sensor is recorded with a sampling frequency F_s and a plate is fixed to the rate table cutting off the optical signal once per revolution. For error analysis, we make the following assumptions regarding the optical sensor system:

- the internal delay of the optical fork sensor is a constant; and
- the clock drift of the pulse-sensing circuitry is negligible.

Based on these assumptions, it can be concluded that $\overline{\delta\Omega}$ results only from the finite operating frequency of the pulse-sensing circuitry. This can be written as

$$\overline{\delta\Omega} = \frac{2\pi N_\theta}{T_\Omega + \delta T_\Omega} - \frac{2\pi N_\theta}{T_\Omega} = -\frac{2\pi N_\theta}{T_\Omega(T_\Omega/\delta T_\Omega + 1)}, \tag{3.20}$$

where T_Ω represents the rotation time and δT_Ω the respective error resulting from the finite operating frequency F_s. As stated above, the rotation

angle $2\pi N_\theta$ is free of error. Based on the assumptions made above, it holds that

$$T_\Omega = \frac{2\pi N_\theta}{\overline{\Omega}} \text{ and } \delta T_\Omega = \frac{t_\epsilon}{F_s}, \text{ where } t_\epsilon \in (-1, 1).$$

The latter equation states that the error δT_Ω in the measured rotation time is limited by the time between two successive samples in the pulse-sensing circuitry. Given this, by substituting the equations for T_Ω and δT_Ω into (3.20) and taking the absolute values, we obtain

$$\left|\overline{\delta\Omega}\right| = \frac{\overline{\Omega}^2}{\left|2\pi N_\theta \frac{F_s}{t_\epsilon} + \overline{\Omega}\right|}. \tag{3.21}$$

Clearly, the maximum $\left|\overline{\delta\Omega}\right|$ is approached as $t_\epsilon \to -1$. Then, assuming further that $2\pi N_\theta F_s > \overline{\Omega}$, (3.21) reduces to

$$\left|\overline{\delta\Omega}\right| < \frac{\overline{\Omega}^2}{2\pi N_\theta F_s - \overline{\Omega}}. \tag{3.22}$$

Figure 3.3 presents a plot of $\left|\overline{\delta\Omega}\right|$ as a function of $\overline{\Omega}$ while keeping the rotation time T_Ω a constant. This represents a situation where the velocity of the rate table is set according to a simple piece-wise constant, i.e. a "staircase" function with equally spaced ramps. The irregular shape of the graphs are a consequence of the fact that $N_\theta = \overline{\Omega} T_\Omega/(2\pi)$ can only have integer values. Errors below the rotation rate of the Earth are obtained for rotation rates up to 10π rad/s (1800 °/s), even with a rather conservative sample rate of 100 kHz. Naturally, longer rotation times yield more accurate results, but at the expense of the time it takes to complete the calibration procedure. Notice that the graph only considers the errors in the reference measurements; obviously the temporal instability of the sensors also contributes to the choice, but this aspect is not considered in this thesis.

Errors in the measured specific forces

With the specific force measurements, non-zero $\delta\Omega$ causes errors in two components of $f^r_{i,j}$. In the direction of e_{rx}, the total error is governed by the centripetal acceleration $\overline{a}_x = \overline{\Omega}^2 r$. Since non-zero $\delta\Omega$ implies that $\dot{\Omega}$ necessarily deviates from zero, the tangential acceleration $\overline{a}_y = -\overline{\dot{\Omega}} r$ also results in the direction e_{ry}. The error caused by both of these terms will be investigated assuming that the error in the radius r is negligible. In this

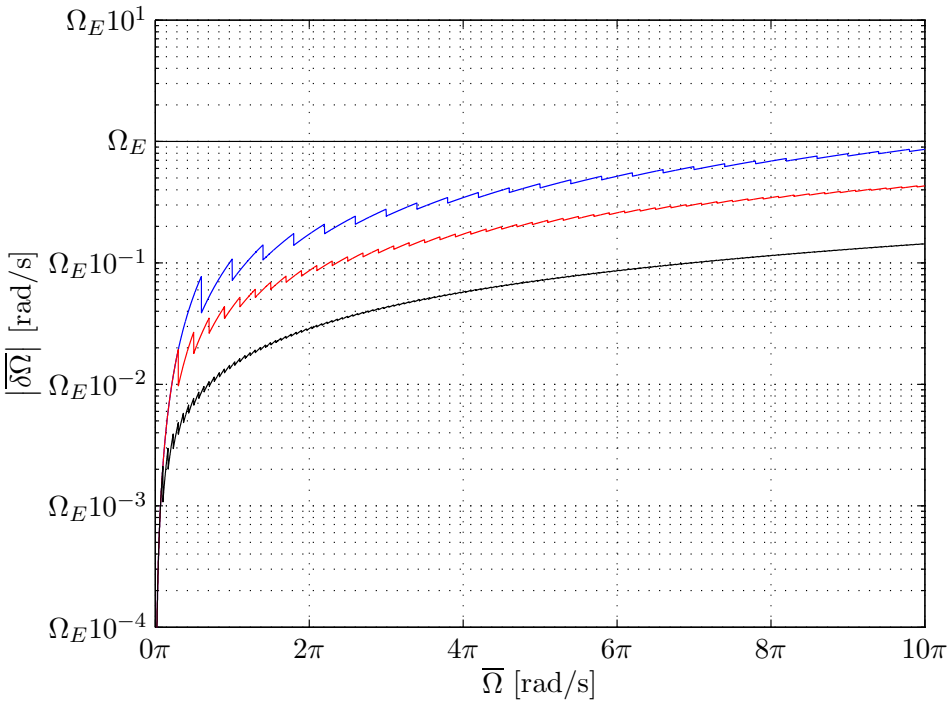

Figure 3.3: Absolute error $|\overline{\delta\Omega}|$ as a function of rotation rate $\overline{\Omega}$ for sampling rate $F_s = 100$ kHz. The blue, red and black lines represent the respective rotation times of $T_\Omega = 5$, $T_\Omega = 10$, and $T_\Omega = 30$ seconds.

case, the average of the centripetal acceleration $\overline{a}_x$ yields

$$\begin{aligned}
\overline{\delta a_x} &= \frac{1}{2\pi N_\theta} \int_{\theta_0}^{\theta_0 + 2\pi N_\theta} (\overline{\Omega} + \delta\Omega)^2 r - \overline{\Omega}^2 r \; d\theta \\
&= \frac{r}{2\pi N_\theta} \int_{\theta_0}^{\theta_0 + 2\pi N_\theta} 2\overline{\Omega}\delta\Omega + \delta\Omega^2 \; d\theta \\
&= \frac{r}{2\pi N_\theta} \left(2\overline{\Omega} \int_{\theta_0}^{\theta_0 + 2\pi N_\theta} \delta\Omega \; d\theta + \int_{\theta_0}^{\theta_0 + 2\pi N_\theta} \delta\Omega^2 \; d\theta \right). \qquad (3.23)
\end{aligned}$$

Hence, $\overline{\delta a_x}$ is bounded by

$$|\overline{\delta a_x}| \leq r \left(2\overline{\Omega} \, |\overline{\delta\Omega}| + |\delta\Omega|^2_{max} \right), \qquad (3.24)$$

where $|\delta\Omega|^2_{max}$ represents the square of the maximum absolute momentary error in Ω. Similarly for the average of the tangential acceleration $\overline{a}_y$, where it holds that

$$\overline{\delta a_y} = -\frac{1}{2\pi N_\theta} \int_{\theta_0}^{\theta_0 + 2\pi N_\theta} \dot{\hat{\Omega}} r \; d\theta = -\frac{r}{2\pi N_\theta} \int_{\theta_0}^{\theta_0 + 2\pi N_\theta} \delta\dot{\Omega} \; d\theta. \qquad (3.25)$$

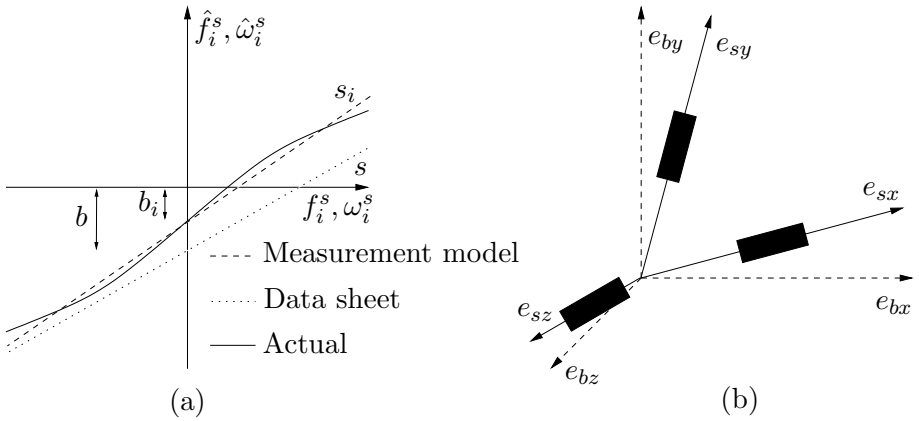

Figure 3.4: An illustration of the deterministic error sources of an IMU.

Then,

$$\left|\overline{\delta a}_y\right| \leq r|\delta\dot{\Omega}|_{max}, \tag{3.26}$$

where the maximum absolute momentary acceleration $|\delta\dot{\Omega}|_{max}$ is most easily estimated according to the gyroscopes which are used. The same holds for $|\delta\Omega|^2_{max}$, which is related to the error in the centripetal acceleration.

3.4 Measurement model

This section discusses the properties of the measurement model, which determine the non-stochastic relations between $(\hat{\omega},\hat{f})$ and (ω,f). In this thesis, simple affinity will be employed as a model of the measurements. This section will also show that, despite their simplicity, these measurement models can compensate for most of the major sources of error arising from typical consumer-grade IMUs.

As discussed in P1, the errors of an IMU can be divided into two categories. These are the errors caused by:

(a) the individual sensors, and

(b) the accelerometer and gyroscope triads within the IMU.

Figure 3.4 gives a graphical representation of these errors. The solid line in Figure 3.4 (a) depicts the actual input (f_i^s, ω_i^s) / output $(\hat{f}_i^s, \hat{\omega}_i^s)$ relation with regard to an individual sensor within the IMU. The dashed line depicts the input/output relation of an affine measurement model based on the calibrated scale factor (s_i) and bias offset (b_i) for the sensor in question, while the dotted line depicts the model derived using the corresponding parameters (s,b), as specified by the manufacturer. The remaining differences between

the modeled measurement and the actual input/output relation are due to the error sources not accounted for in the chosen measurement model. Non-linearity is a typical example of such an error [2]. Here, the measurement models concerning individual sensors are of the form

$$f^s = \mathrm{diag}(s_f)(\hat{f}^s + b_f) \tag{3.27}$$
$$\omega^s = \mathrm{diag}(s_\omega)(\hat{\omega}^s + b_\omega), \tag{3.28}$$

where $\mathrm{diag}(s_f), \mathrm{diag}(s_\omega) \in \mathbb{R}^{3\times 3}$ are the diagonal matrices consisting of the scale factors for each sensor, while $b_f, b_\omega \in \mathbb{R}^3$ are the vectors arising from the bias offsets of each sensor. For any useful sensor triad, the scale factors are non-zero, and thus, $\mathrm{diag}(s_f)$ and $\mathrm{diag}(s_\omega)$ are invertible.

Type (b) errors are depicted in Figure 3.4 (b). Due to the inevitable errors in the manufacturing process of the IMU, the sensitive axes of the sensors are not orthogonal to each other and do not agree with the basis of the b-frame. So, a change of basis is required in order to convert the calibrated readings of the individual sensors (f^s, ω^s) to the desired basis (f^b, ω^b). This conversion is of the form

$$f^b = B_f f^s \tag{3.29}$$
$$\omega^b = B_\omega \omega^s, \tag{3.30}$$

where $B_f, B_\omega \in \mathbb{R}^{3\times 3}$ are of the form

$$\begin{bmatrix} {e^b_{bx}}^T e^b_{sx} & {e^b_{bx}}^T e^b_{sy} & {e^b_{bx}}^T e^b_{sz} \\ {e^b_{by}}^T e^b_{sx} & {e^b_{by}}^T e^b_{sy} & {e^b_{by}}^T e^b_{sz} \\ {e^b_{bz}}^T e^b_{sx} & {e^b_{bz}}^T e^b_{sy} & {e^b_{bz}}^T e^b_{sz} \end{bmatrix} \tag{3.31}$$

for the respective s-frame. Since the b-frame has an orthonormal basis, it holds for each column vector of B_f and B_ω that

$$\begin{aligned} & ({e^b_{bx}}^T e^b_{si})^2 + ({e^b_{by}}^T e^b_{si})^2 + ({e^b_{bz}}^T e^b_{si})^2 \\ = \; & {e^b_{si}}^T (e^b_{bx} {e^b_{bx}}^T + e^b_{by} {e^b_{by}}^T + e^b_{bz} {e^b_{bz}}^T) e^b_{si} \\ = \; & {e^b_{si}}^T I e^b_{si} \\ = \; & 1 \;\; \forall \;\; i \in \{x, y, z\}\,. \end{aligned}$$

The matrices B_f and B_ω are invertible – which is the necessary and sufficient condition for the employed measurement models to be bijective – whenever the basis of the respective s-frame spans $\mathbb{R}^3$.

Now, let us combine (3.27) and (3.29) (respectively (3.28) and (3.30)). This yields

$$f^b = B_f \mathrm{diag}(s_f)(\hat{f}^s + b_f) = S_f(\hat{f}^s + b_f) \tag{3.32}$$
$$\omega^b = B_\omega \mathrm{diag}(s_\omega)(\hat{\omega}^s + b_\omega) = S_\omega(\hat{\omega}^s + b_\omega), \tag{3.33}$$

where $S_f = B_f \text{diag}(s_f)$ and $S_\omega = B_\omega \text{diag}(s_\omega)$. Here we utilise the measurement models presented in (3.32) and (3.33). The attached article P1 also considers the gyroscopes' sensitivity to linear accelerations. In this context, however, such sensitivity parameters are omitted as they would make the remaining analysis significantly more complicated, while barely improving the accuracy of the obtained results.

3.5 Calibration procedure

This section deals with the details of the calibration procedure. Let us start by noting that, during the calibration process, we are primarily interested in the inverse relations of (3.32) and (3.33). These are

$$\hat{f}^s = S_f^{-1} f^b - b_f \tag{3.34}$$

$$\hat{\omega}^s = S_\omega^{-1} \omega^b - b_\omega. \tag{3.35}$$

The goal of the calibration procedure is to find numerical representations of the equations (3.32) and (3.33) for the IMU in question. The actual measurement models can then be resolved simply by inverting the obtained numerical representations of S_f^{-1} and S_ω^{-1}. If necessary, the special structure of matrices (B_f,B_ω) and (diag(s_f),diag(s_ω)) allows them to be deduced separately.

For every pair of indices (i,j), it holds that

$$\overline{\hat{f}}^s_{i,j} = S_f^{-1} C_r^{b_i} \overline{f}^r_{i,j} - b_f \tag{3.36}$$

$$\overline{\hat{\omega}}^s_{i,j} = S_\omega^{-1} C_r^{b_i} \overline{\omega}^r_{i,j} - b_\omega. \tag{3.37}$$

These can be rewritten as

$$\left[[C_r^{b_i} \overline{f}^r_{i,j}]_x I \quad [C_r^{b_i} \overline{f}^r_{i,j}]_y I \quad [C_r^{b_i} \overline{f}^r_{i,j}]_z I \quad -I \right] x_f = \overline{\hat{f}}^s_{i,j} \tag{3.38}$$

$$\left[[C_r^{b_i} \overline{\omega}^r_{i,j}]_x I \quad [C_r^{b_i} \overline{\omega}^r_{i,j}]_y I \quad [C_r^{b_i} \overline{\omega}^r_{i,j}]_z I \quad -I \right] x_\omega = \overline{\hat{\omega}}^s_{i,j}, \tag{3.39}$$

where $x_f, x_\omega \in \mathbb{R}^{12}$ contain the elements of S_f^{-1}, b_f and S_ω^{-1}, b_ω respectively in column-wise order. Entries of the form $[\cdot]_i$ refer to the ith component of the vector inside the brackets and $I \in \mathbb{R}^{3\times 3}$. By stacking (3.38) and (3.39) for all pairs of indices i and j, we obtain overdetermined equations with a total of $3 \times N_a \times N_r$ measurements. As mentioned above, the rotation radii (usually different for each b_i) are also treated as unknowns. Then, the equation for the accelerometer triad becomes non-linear with additional N_a unknowns. The Jacobian $J_{i,j}$ of (3.38) for the particular case where the radius is the same for all b_i is

$$J_{i,j} = \left[[C_r^{b_i} \overline{f}^r_{i,j}]_x I \quad [C_r^{b_i} \overline{f}^r_{i,j}]_y I \quad [C_r^{b_i} \overline{f}^r_{i,j}]_z I \quad -I \quad S_f^{-1} C_r^{b_i} \frac{\partial \overline{f}^r_{i,j}}{\partial r} \right] \tag{3.40}$$

where the partial derivative yields

$$\frac{\partial \overline{f}^{r}_{i,j}}{\partial r} = \begin{bmatrix} \overline{\Omega}^{2}_{i,j} + 2\overline{\Omega}_{i,j}\Omega_E \sin L + 1/2\Omega^{2}_{E}(1+\sin^2 L) \\ 0 \\ 0 \end{bmatrix}. \tag{3.41}$$

A detailed discussion of the solution techniques for these equations is presented in Chapter 5. In that chapter, the equations obtained here will be applied to the presented discussion by neglecting the state equation. In this section, x consists of a single state y, and thus x is determined only by the observation equation.

Once a set of reference measurements and observations has been obtained, a calibration specific to a given dynamic range is easily obtained by neglecting those observations which do not fit the given range. If all observations $(\overline{\hat{f}}^{s}_{i,j}, \overline{\hat{\omega}}^{s}_{i,j})$ and reference measurements $(\overline{f}^{b_i}_{i,j}, \overline{\omega}^{b_i}_{i,j})$ are accessible when the gathered data is processed, it is possible to resolve the parameters of the error models on demand.

3.6 Online calibration

One of the problems of working with consumer-grade inertial sensors is that no single set of parameters obtained with the presented calibration process can be expected to produce a measurement model that will last indefinitely. The bias levels b and the scale factors s of the individual sensors are typical examples of parameters which are dependent on the operating conditions, such as temperature. Moreover, regardless of the operating conditions, the bias levels and scale factors of the sensors tend to be different each time the measurement device is activated, in addition to which they fluctuate stochastically during the measurement process [2]. While the sensors' deterministic dependency on the operating conditions can be compensated for by means of more sophisticated calibration equipment, the stochastic errors can not.

This thesis focuses on those unidealities in the sensors which can be compensated for using a set of scale factors (s_f, s_ω) and bias levels (b_f, b_ω), which are *constants* during the time interval $[T_0, T_1]$. That is, the proposed online calibration scheme is able to compensate for the scale factor and bias errors caused by, for example, operational conditions, provided that they do not change between $[T_0, T_1]$. This is a reasonable assumption to make, given that the time frame is short enough: the correlation times of the stochastic processes describing, for example, the change of the bias with time are often measured in minutes [69].

The idea of the online calibration scheme is simply to incorporate the measurement model into the dynamics model and estimate the unknown parameters along with the actual states; just as was done above with the rotation radii. Many times, the available observations are only indirect measures of the unknown parameters. When combined with the numerical solution of the dynamics model, the online calibration scheme is predictably less robust than the static calibration method presented above. This is especially true if the observations are particularly unreliable and/or there are very few observations. Consequently, the online calibration is only regarded as an ancillary method for the already-completed calibration; its objective is only to find *corrections* to the already-found scale factors and bias levels. If the respective corrections are denoted as $\hat{s}_f$, $\hat{s}_\omega$, $\hat{b}_f$ and $\hat{b}_\omega$, the models suitable for estimating them are

$$f^b = S_f(\mathrm{diag}(\hat{s}_f)\hat{f}^s + \hat{b}_f) \tag{3.42}$$

$$\omega^b = S_\omega(\mathrm{diag}(\hat{s}_\omega)\hat{\omega}^s + \hat{b}_\omega). \tag{3.43}$$

In order to simplify the notation, the bias levels b_f and b_ω are included in the corrections. Note the slight difference between (3.42) and (3.32) (similarly for (3.43) and (3.33)), which was done just to separate the scale factor and bias corrections from each other. It is also worth mentioning that the estimated parameters must be incorporated in the s-frame. This way, the corrections apply directly to individual sensors, which is the key idea of the online calibration method. Otherwise, matrices (S_f,S_ω) "blend" the contributions of individual sensors and the measurement models used will not model the situation adequately.

A straightforward computation shows that the combination of the dynamics model (2.26) presented on page 21 with measurement models (3.42) and (3.43) yields

$$\begin{bmatrix} \dot{c}_1 \\ \dot{c}_2 \\ \dot{c}_3 \\ \dot{v} \\ \dot{p} \\ \dot{\hat{s}}_\omega \\ \dot{\hat{b}}_\omega \\ \dot{\hat{s}}_f \\ \dot{\hat{b}}_f \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} 0 & -c_3 & c_2 \end{bmatrix} S_\omega \left(\mathrm{diag}(\hat{s}_\omega)\hat{\omega} + \hat{b}_\omega \right) \\ \begin{bmatrix} c_3 & 0 & -c_1 \end{bmatrix} S_\omega \left(\mathrm{diag}(\hat{s}_\omega)\hat{\omega} + \hat{b}_\omega \right) \\ \begin{bmatrix} -c_2 & c_1 & 0 \end{bmatrix} S_\omega \left(\mathrm{diag}(\hat{s}_\omega)\hat{\omega} + \hat{b}_\omega \right) \\ \begin{bmatrix} c_1 & c_2 & c_3 \end{bmatrix} S_f \left(\mathrm{diag}(\hat{s}_f)\hat{f} + \hat{b}_f \right) \\ v \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ g \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}. \tag{3.44}$$

In (3.44), the zero rows indicate the fact that the correction terms are modeled as constants. Chapter 4 discusses how (3.44) can be written as a state equation.

3.7 Summary

This chapter presented the appropriate techniques for calibrating IMUs. The static calibration method performed on the rate table can be used to find the parameters related to individual sensors, and to the misalignment of the sensors within the IMU. The method of online calibration allows one to find the scale factors and bias levels for individual sensors. The measurement models presented here are merely examples of all the possible measurement models. The online calibration method, for example, can readily be extended to take into account the misalignment errors in the IMU.

The online calibration method is particularly useful where it is not possible to keep the IMU stationary, and therefore it is difficult to separate out the bias offsets. It is generally accepted that both the bias offsets and the scale factors of the individual sensors have a significant effect on the accuracy of the results [2]. It is not uncommon for temperature variations, for instance, to change the bias offsets and scale factors of consumer-grade sensors by several percent (see, for example, Table 6.1 on page 80). In such cases, accurate measurement models are often of key importance, and this method of online calibration is a vital tool for this. It can be used as long as there are enough observations to allow the additional correction terms to be estimated with reasonable accuracy.

It is hereafter assumed that the sensors have been calibrated. At the very least, generic measurement models can be employed to yield the factory calibration for each sensor triad. With the exception of the online calibration method, the next chapter constructs the state and observation equations using the compensated measurements ω and f.

Chapter 4

State and observation equations

This chapter deals with the question of how to systematically "convert" the starting point – a dynamics model together with a number of observations – into the form of state and observation equations. The underlying fixed-interval smoothing problem will be posed and solved in Chapter 5 by exploiting only the information provided by these two "standardized" equations. This chapter summarises and extrapolates the ideas and methods presented in publications P2 – P4.

It should be noted that there is basically nothing new in the way the state equation is constructed from the provided measurements. Both of the methods presented here are well-known and extensively used in the field of numerical mathematics [5]. The only difference is in the way the boundary values are treated. Whereas standard methods are expected to provide us with a unique solution (assuming that one exists), here the state equation itself is not expected to contain all the information required for a unique solution. As will be explained in the following chapter, it is the properties of the state and observation equations together that matters.

In this thesis, the focus is on generic situations, where the measurements ω and f are the only sources of information related to the state equation. In some situations, additional information about the dynamics model may be available. This could, for example, come from a model describing the behavior of the system throughout the fixed time interval $[T_0, T_1]$ (cf. item (c) on page 10). In that case, the observation equation – now including also ω and f (if available) – is used to specify and particularize the provided model. The reader is provided with sufficiently detailed information to construct the respective state and observation equations. After that, the process of resolving the situation follows the procedure outlined in Chapter 5.

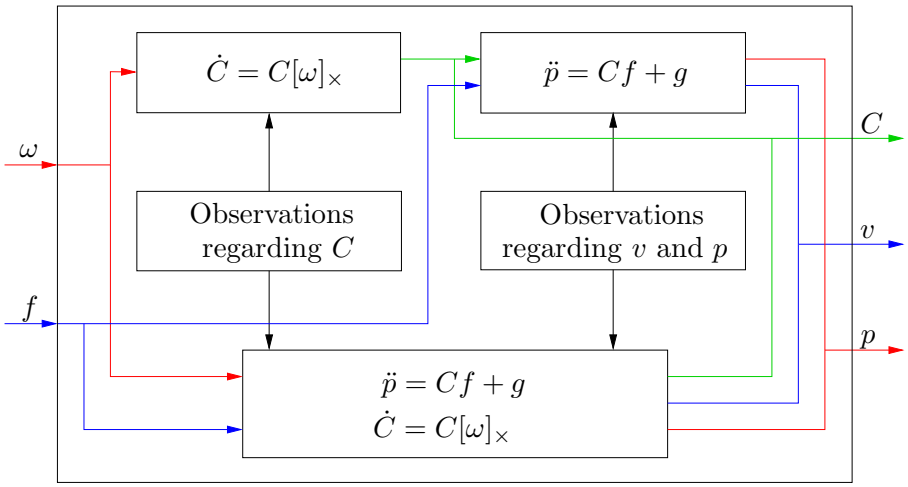

Figure 4.1: Two different strategies to obtain estimates of attitude (C), velocity (v), and position (p).

Referring back to the block diagram (Figure 2.1 on page 14), both this chapter and the next one are about the block in which the attitude, velocity, and position are estimated. Let us now take a closer look at this particular block. As illustrated in Figure 4.1, two different strategies for estimating C, v, and p are discussed here. Given an application with a specific goal in mind (cf. item (a) on page 10), the corresponding state equation could be written directly in terms of the estimated quantity. It is not in the scope of this thesis to focus on such specific situations. Instead, the treatment starts with the generic situation where C, v, and p are estimated together. In Figure 4.1, this is illustrated by the block at the bottom of the diagram. When this is the case, the obtained state and observation equations are generally non-linear and written in the previously encountered form

$$f_s(x) = z_s + w_s \tag{4.1}$$
$$f_o(x) = z_o + w_o, \tag{4.2}$$

where $x \in \mathbb{R}^{kN}$. The properties and effects of the error terms w_s and w_o are discussed later, in Chapter 5. The solution process for such non-linear equations usually requires knowledge of the respective Jacobians

$$J_s(x) = \frac{\partial f_s(x)}{\partial x} \tag{4.3}$$
$$J_o(x) = \frac{\partial f_o(x)}{\partial x}. \tag{4.4}$$

Therefore, the construction of the Jacobians (4.3) and (4.4) is also presented in this chapter. A detailed discussion of the techniques for estimating C, v, and p together follows in sections 4.1.1 and 4.1.2.

It is not always necessary to estimate C, v, and p together. If each individual observation can be represented either in terms of attitude C or position p, it is not merely possible, but often advantageous to split the estimation problem into two sub-problems: Attitude C is obtained as the solution of the problem depicted in the upper left corner of Figure 4.1, while the velocity v and position p are obtained as the solution to the problem depicted in the upper right corner of the diagram. The attitude estimation problem is discussed in P4, and the velocity/position estimation problem in P2 – P3. In this thesis – as was done in the referred publications – the treatment of this solution strategy is limited to situations where the respective state and observation equations can be presented in the linear form

$$J_s X = Z_s + W_s \tag{4.5}$$
$$J_o X = Z_o + W_o, \tag{4.6}$$

where the unknown, X, is generally a matrix and the coefficient matrices J_s and J_o are constants. As mentioned above, the properties of the respective error terms W_s and W_o are considered in the following chapter. Sections 4.1.3 and 4.1.4 present a detailed discussion of this estimation technique.

For notational convenience, it is assumed that the measurements are obtained with a constant sample rate F_s. This yields a constant step length $h = 1/F_s$ and a constraint $T_1 = T_0 + (N-1)h$ for the end points of the time interval $[T_0, T_1]$. However, all the methods presented here can be readily generalized to accept a variable step length. Indeed, the IMUs used in the examples discussed in Chapter 6 do actually employ a variable step length.

4.1 State equation

This section presents the general principles for constructing the state equation from the obtained measurements $\omega_i, f_i \ \forall \ i \in [1, N]$. In sections 4.1.1 – 4.1.3, the underlying dynamics model is a first-order ODE and the state equations are constructed using a relaxation method. In section 4.1.4, the underlying dynamics model is a second-order ODE and the respective state equation is constructed with FEM.

4.1.1 Estimating attitude, velocity, and position

Here, a technique suitable for estimating C, v, and p simultaneously is discussed. This is done in terms of the ODE (2.26) introduced on page 21,

whose linearity allows it to be written it in the shorthand form

$$\dot{y} = \frac{\partial f(t,y)}{\partial y} + z = Ay + z, \tag{4.7}$$

where

$$A = \begin{bmatrix} 0 & \omega_z I & -\omega_y I & 0 & 0 \\ -\omega_z I & 0 & \omega_x I & 0 & 0 \\ \omega_y I & -\omega_x I & 0 & 0 & 0 \\ f_x I & f_y I & f_z I & 0 & 0 \\ 0 & 0 & 0 & I & 0 \end{bmatrix} \quad \text{and} \quad z = \begin{bmatrix} 0 \\ 0 \\ 0 \\ g \\ 0 \end{bmatrix} \tag{4.8}$$

are independent of y.

Here, we shall employ a relatively simple FDE based on the trapezoidal rule, which yields

$$g_i(y_i, y_{i+1}) = y_{i+1} - y_i - \frac{h}{2}[f(t_i, y_i) + f(t_{i+1}, y_{i+1})] = 0 \; \forall \; i \in [1, N-1], \tag{4.9}$$

when applied to the generic dynamics model $\dot{y} = f(t, y)$ [58]. As an implicit method, the trapezoidal rule is stable and, with sufficiently small time steps, it is reasonably accurate [6]. Moreover, the resulting FDEs are symmetric, as they treat both directions alike. Hence, the FDEs presented here are well suited for relaxation methods, which do not have a preferred direction [58]. When (4.7) is applied to (4.9), it follows that

$$\begin{aligned} 0 &= y_{i+1} - y_i - \frac{h}{2}[A_i y_i + A_{i+1} y_{i+1}] - h z_i \\ h z_i &= (-\frac{h}{2} A_i - I) y_i + (-\frac{h}{2} A_{i+1} + I) y_{i+1} \;\; \forall \;\; i \in [1, N-1], \end{aligned} \tag{4.10}$$

where A_i, z_i are shorthand forms of $A(t_i)$, $z(t_i)$ and $I \in \mathbb{R}^{k \times k}$. The state equation follows by writing

$$f_s(x) = \begin{bmatrix} (-h/2A_1 - I) y_1 + (-h/2A_2 + I) y_2 \\ (-h/2A_2 - I) y_2 + (-h/2A_3 + I) y_3 \\ \vdots \\ (-h/2A_{N-1} - I) y_{N-1} + (-h/2A_N + I) y_N \end{bmatrix} \tag{4.11}$$

and

$$z_s = h \begin{bmatrix} z_1 & z_2 & \cdots & z_{N-1} \end{bmatrix}^T. \tag{4.12}$$

Based on (4.11) and (4.9), the Jacobian can be written in the form

$$J_s = \begin{bmatrix} J_{1,1} & J_{1,2} & & & \\ & J_{2,2} & J_{2,3} & & \\ & & \ddots & \ddots & \\ & & & J_{N-1,N-1} & J_{N-1,N} \end{bmatrix} \tag{4.13}$$

where

$$J_{i,i} = \frac{\partial g_i(y_i, y_{i+1})}{\partial y_i} = -\frac{h}{2}A_i - I$$
$$J_{i,i+1} = \frac{\partial g_i(y_i, y_{i+1})}{\partial y_{i+1}} = -\frac{h}{2}A_{i+1} + I$$

for all $i \in [1, N-1]$. The properties of $J_{i,i}$ and $J_{i,i+1}$ guarantee that the rank of the obtained Jacobian $J_s \in \mathbb{R}^{k(N-1)\times kN}$ is $k(N-1)$ [58]. The missing k rows are because of the boundary values, which were intentionally left out of the state equation.

4.1.2 Estimating attitude, velocity, and position with online calibration

This section presents the state equation for the estimation technique discussed above, in section 3.6. There, the sensor-error compensation terms ($\hat{s}_f$, $\hat{s}_\omega$, $\hat{b}_f$ and $\hat{b}_\omega$) are estimated along with the actual state estimates C, v, and p. As above, the FDEs used in this section will be based on the trapezoidal rule.

Let us start by writing the dynamics model (3.44) in the form

$$\begin{bmatrix}\dot{y}\\ \dot{\hat{y}}\end{bmatrix} = \begin{bmatrix}f(t, y, \hat{y})\\ 0\end{bmatrix} + \begin{bmatrix}z\\ 0\end{bmatrix}, \tag{4.14}$$

where y and z are the same terms as were presented in the previous section, and $\hat{y}$ represents the constant sensor-error compensation terms. Then, it holds that

$$A = \frac{\partial f(t, y, \hat{y})}{\partial y} = \begin{bmatrix} 0 & \omega_z I & -\omega_y I & 0 & 0\\ -\omega_z I & 0 & \omega_x I & 0 & 0\\ \omega_y I & -\omega_x I & 0 & 0 & 0\\ f_x I & f_y I & f_z I & 0 & 0\\ 0 & 0 & 0 & I & 0 \end{bmatrix}$$

and

$$B = \frac{\partial f(t, y, \hat{y})}{\partial \hat{y}} = \begin{bmatrix} \begin{bmatrix} 0 & -c_3 & c_2\\ c_3 & 0 & -c_1\\ -c_2 & c_1 & 0 \end{bmatrix} S_\omega \begin{bmatrix}\mathrm{diag}(\hat{\omega}) & I\end{bmatrix} & 0^{3\times 6}\\ 0^{3\times 6} & C S_f \begin{bmatrix}\mathrm{diag}(\hat{f}) & I\end{bmatrix}\\ 0^{3\times 6} & 0^{3\times 6} \end{bmatrix}.$$

In A and B, (ω, f) denote the compensated measurements on the left hand side and $(\hat{\omega}, \hat{f})$ the uncompensated measurements on the right hand side of

(3.42) and (3.43). In matrix B, $0^{3\times 6}$ stands for a 3×6 zero matrix, and $C = \begin{bmatrix} c_1 & c_2 & c_3 \end{bmatrix}$, as before.

Since it holds that $\dot{\hat{y}} = 0$, it is not necessary to treat $\hat{y}$ as variables for all $i \in [1, N]$. Instead, $\hat{y}$ is treated as a set of constant parameters, which yields FDEs of the form

$$hz_i = (-\frac{h}{2}A_i - I)y_i + (-\frac{h}{2}A_{i+1} + I)y_{i+1} - \frac{h}{2}(B_i + B_{i+1})\hat{y} \ \ \forall \ \ i \in [1, N-1]. \quad (4.15)$$

Then, by denoting

$$J_{i,i} = -\frac{h}{2}A_i - I \qquad J_{i,i+1} = -\frac{h}{2}A_{i+1} + I$$
$$\hat{J}_{i,i} = -\frac{h}{2}B_i \qquad \hat{J}_{i,i+1} = -\frac{h}{2}B_{i+1},$$

(4.15) can be written in the matrix form

$$J_s(x) = \begin{bmatrix} J_{1,1} & J_{1,2} & & & & \hat{J}_{1,1} + \hat{J}_{1,2} \\ & J_{2,2} & J_{2,3} & & & \hat{J}_{2,2} + \hat{J}_{2,3} \\ & & \ddots & \ddots & & \vdots \\ & & & J_{N-1,N-1} & J_{N-1,N} & \hat{J}_{N-1,N-1} + \hat{J}_{N-1,N} \end{bmatrix}, \quad (4.16)$$

where

$$x = \begin{bmatrix} y_1^T & \cdots & y_N^T & \hat{s}_\omega^T & \hat{b}_\omega^T & \hat{s}_f^T & \hat{b}_f^T \end{bmatrix}^T. \quad (4.17)$$

In this case, the dimensions of $J_s(x)$ are $k(N-1) \times (kN+12)$. The rank of the matrix, $\text{rank}[J_s(x)] = k(N-1)$ is the same as above. The overall form of the state equation is $f_s(x) = z_s$, where z_s is given in (4.12). Note that in this case, the state equation is non-linear.

4.1.3 Estimating attitude

This section explains how the state equation based on the attitude equation $\dot{C} = C[\omega]_\times$ is constructed. Here, we seek for a linear state equation in the form of (4.5), where $X = \begin{bmatrix} C_1 & C_2 & \cdots & C_N \end{bmatrix}^T$. Because of the orthogonality constraint (2.30), the FDE is chosen in a slightly different fashion. As the constraint is quadratic, it is also possible to choose method (iii) from the list presented on page 21. As an example of a numerical method which automatically preserves the orthogonality, consider a matrix-valued variant of the FDE based on the midpoint method [6] applied to (2.28):

$$C_{i+1} - C_i - hC_{i+1/2}[\omega_{i+1/2}]_\times = 0 \ \ \forall \ \ i \in [1, N-1]\,. \quad (4.18)$$

Now, let us replace the terms evaluated at the midpoint by their linear approximations (cf. Crank-Nicholson [70])

$$[\omega_{i+1/2}]_\times \approx \frac{1}{2}([\omega_i]_\times + [\omega_{i+1}]_\times) \ \text{ and } \ C_{i+1/2} \approx \frac{1}{2}(C_i + C_{i+1}).$$

By substituting the approximations in (4.18) and transposing the result, the equation

$$\left[\frac{h}{4}([\omega_i]_\times + [\omega_{i+1}]_\times) - I\right] C_i^T + \left[\frac{h}{4}([\omega_i]_\times + [\omega_{i+1}]_\times) + I\right] C_{i+1}^T = 0 \;\; i \in [1, N-1]$$

is obtained. The last equation can be written in the short hand form

$$J_{i,i} C_i^T + J_{i,i+1} C_{i+1}^T = 0 \;\; \forall \;\; i \in [1, N-1]\,, \tag{4.19}$$

where

$$J_{i,i} = \frac{h}{4}([\omega_i]_\times + [\omega_{i+1}]_\times) - I \;\text{ and }\; J_{i,i+1} = \frac{h}{4}([\omega_i]_\times + [\omega_{i+1}]_\times) + I.$$

Based on (4.19), the overall structure of the resulting Jacobian J_s is of the form presented in (4.13).

Orthogonality constraint

It still needs to be shown that the chosen FDE actually does preserve orthogonality. Based on (4.19), it holds that

$$C_{i+1}^T = J_{i,i+1}^{-1}(-J_{i,i}) C_i^T \;\; \forall \;\; i \in [1, N-1]\,.$$

Since Ω is skew-symmetric, $-J_{i,i}^T = J_{i,i+1}$. Thus,

$$\begin{aligned} C_{i+1} C_{i+1}^T &= \left[J_{i,i+1}^{-1}(-J_{i,i}) C_i^T\right]^T \left[J_{i,i+1}^{-1}(-J_{i,i}) C_i^T\right] \\ &= C_i J_{i,i+1} J_{i,i}^{-1} J_{i,i+1}^{-1} J_{i,i} C_i^T \;\; \forall \;\; i \in [1, N-1]\,. \end{aligned}$$

Now, assume that the equation

$$\frac{h}{4}([\omega_i]_\times + [\omega_{i+1}]_\times) x_\omega = \lambda_\omega x_\omega \tag{4.20}$$

holds, where $\lambda_\omega \in \mathbb{C}$ represents the eigenvalues and $x_\omega \in \mathbb{C}^3$ the respective eigenvectors of matrix $J_{i,i} + I = J_{i,i+1} - I$. Then, it also holds that

$$J_{i,i} x_\omega = (\lambda_\omega - 1) x_\omega \tag{4.21}$$

$$J_{i,i+1} x_\omega = (\lambda_\omega + 1) x_\omega. \tag{4.22}$$

Based on (4.21) and (4.22), the eigenvectors of $J_{i,i}$ and $J_{i,i+1}$ are x_ω. The eigenvalues of $J_{i,i}$ and $J_{i,i+1}$ are $\lambda_\omega - 1$ and $\lambda_\omega + 1$ respectively. Then, based on the eigen-decomposition theorem, $J_{i,i}$ can be written in the form $Q \Lambda_{i,i} Q^{-1}$, where $\Lambda_{i,i} \in \mathbb{C}^{3\times 3}$ is a diagonal matrix consisting of the eigenvalues of $J_{i,i}$

and $Q \in \mathbb{C}^{3\times 3}$ is a square matrix consisting of the eigenvectors of $J_{i,i}$ [71]. Similarly, $J_{i,i+1} = Q\Lambda_{i,i+1}Q^{-1}$. Hence, it follows that

$$\begin{aligned} C_{i+1}C_{i+1}^T &= C_iQ\Lambda_{i,i+1}\Lambda_{i,i}^{-1}\Lambda_{i,i+1}^{-1}\Lambda_{i,i}Q^{-1}C_i^T \\ &= C_iQ\Lambda_{i,i+1}\Lambda_{i,i+1}^{-1}\Lambda_{i,i}^{-1}\Lambda_{i,i}Q^{-1}C_i^T \\ &= C_iC_i^T \ \ \forall \ \ i \in [1, N-1]\,. \end{aligned} \tag{4.23}$$

Thus, if the orthogonality constraint holds for time instance i, based on (4.23), it follows that it holds for all time instances $j \neq i$ as well.

4.1.4 Estimating position and velocity

This section discusses the estimation of velocity and position, given specific force measurements f_i and DCMs C_i. As stated in (2.24), it holds that $\ddot{p} = Cf + g$.

Based on the sections above, one could simply represent the second-order ODE as two first-order ODEs and apply a suitable FDE to obtain the state equation. Here, we proceed differently and estimate the position p with a custom-made finite element method (FEM). The main incentive for using this approach is an energy-related optimality criterion, which is discussed below.

The energy-minimizing property of FEM

Let us start by investigating the functional

$$G(\hat{v}) = \frac{1}{2}m\int_{T_0}^{T_1} \delta v^T\delta v \; dt = \frac{1}{2}m\int_{T_0}^{T_1} (\hat{v}-v)^T(\hat{v}-v) \; dt, \tag{4.24}$$

where $m \in \mathbb{R}_+$ represents the mass of the object. In (4.24), $\delta v = \hat{v} - v$, where $\hat{v}$ represents the measured velocity and v the true velocity[1], as before. Then, (4.24) is proportional to the average kinetic energy of the velocity error δv over the fixed time interval $[T_0, T_1]$. Obviously, the smaller the value of $G(\hat{v})$, the better. The following discussion defines the conditions for the sought-after situation in which $G(\hat{v})$ is minimized.

The fact that a solution does exist can be proved by first showing that the cost function (4.24) is *convex*. Then, if the cost function has a *critical point*, it will be located at the *global minimum* of $G(\hat{v})$. Both of these properties

[1]Here, v represents the velocity obtained by solving equation $\ddot{p} = Cf + g$ without errors caused by any numerical discretization method. In the real world, given only discretized samples of C and f, v is unattainable.

can be stated in terms of the first and second Gâteaux derivatives of the cost function [72]. To compute the derivatives, let us first write

$$\begin{aligned} G(\hat{v}+\tau\dot{u}) &= \frac{1}{2}m\int_{T_0}^{T_1}(\hat{v}-v+\tau\dot{u})^T(\hat{v}-v+\tau\dot{u})\;dt \\ &= \frac{1}{2}m\int_{T_0}^{T_1}(\hat{v}-v)^T(\hat{v}-v)+2\tau(\hat{v}-v)^T\dot{u}+\tau^2\dot{u}^T\dot{u}\;dt \end{aligned} \tag{4.25}$$

for all $u \in H^1$. In (4.25), $\tau \in \mathbb{R}$ and H^1 is a Sobolev space, whose functions $u : \mathbb{R} \to \mathbb{R}^3$ and their *weak derivatives* u' are square-integrable [73,74].

The first and second Gâteaux derivatives of (4.25) are

$$\left[\frac{d}{d\tau}G(\hat{v}+\tau\dot{u})\right]_{\tau=0} = m\int_{T_0}^{T_1}(\hat{v}-v)^T\dot{u}\;dt \tag{4.26}$$

$$\left[\frac{d^2}{d\tau^2}G(\hat{v}+\tau\dot{u})\right]_{\tau=0} = m\int_{T_0}^{T_1}\dot{u}^T\dot{u}\;dt. \tag{4.27}$$

Now, since the second gateaux derivative is always positive, the cost function $G(\hat{v})$ is convex [72]. Therefore, the global minimizer of $G(\hat{v})$ is located at the point where the it holds for the first Gâteaux derivative that

$$m\int_{T_0}^{T_1}(\hat{v}-v)^T\dot{u}\;dt = 0 \;\;\forall\;\; u \in H^1. \tag{4.28}$$

Then, by integrating (4.28) by parts and dividing both sides with $m \neq 0$, we obtain

$$[(\hat{v}-v)^Tu]_{T_0}^{T_1} - \int_{T_0}^{T_1}(\hat{a}-a)^Tu\;dt = 0 \;\;\forall\;\; u \in H^1, \tag{4.29}$$

where the notation "$[\cdot]_{T_0}^{T_1}$" stands for substitution and a for acceleration, as before. This can be further reduced to

$$\int_{T_0}^{T_1}\hat{a}^Tu\;dt = \int_{T_0}^{T_1}a^Tu\;dt \;\;\forall\;\; u \in H^1 \tag{4.30}$$

whenever the boundary term

$$[(\hat{v}-v)^Tu]_{T_0}^{T_1} = \delta v^T(T_1)u(T_1) - \delta v^T(T_0)u(T_0) \tag{4.31}$$

vanishes. This occurs when the boundary values imply that $\delta v = 0$ at the boundaries. In other words, when the exact velocity of the object is known at the boundaries. A less obvious fact is that the boundary values that imply $\delta p = 0$ at either of the boundaries also impose the variation u to zero at the same boundary [75].

Derivation of the finite element method

The equation (4.30), as derived above, is inspiring. The construction of the custom-made finite element method can be started from

$$\int_{T_0}^{T_1} \ddot{p} \circ u \; dt = \int_{T_0}^{T_1} a \circ u \; dt \;\; \forall \;\; u \in H^1, \tag{4.32}$$

where $\circ$ represents the Hadamard product for matrices [66]. Notice that the three equations contained in (4.32) guarantee that (4.30) holds whenever the boundary term (4.31) vanishes.

We continue by eliminating the second derivative $\ddot{p}$ by integrating the left hand side of (4.32) by parts, which yields

$$\int_{T_0}^{T_1} \ddot{p} \circ u \; dt = [v \circ u]_{T_0}^{T_1} - \int_{T_0}^{T_1} \dot{p} \circ \dot{u} \; dt = \int_{T_0}^{T_1} a \circ u \; dt \;\; \forall \;\; u \in H^1. \tag{4.33}$$

By evaluating the substitution term and rearranging the terms, we obtain

$$\int_{T_0}^{T_1} \dot{p} \circ \dot{u} \; dt = - \int_{T_0}^{T_1} a \circ u \; dt + [v \circ u]_{T_0}^{T_1} \;\; \forall \;\; u \in H^1. \tag{4.34}$$

Now, as H^1 is an infinite-dimensional space, (4.34) it is not well-suited for numerical computations. Thus, let us correspondingly approximate H^1 with a finite-dimensional sub-space $V \subset H^1$, spanned by the *basis functions* $\phi_i \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}^T$ $(i \in [1, N])$. Let us also exploit the very same basis functions ϕ_j to discretize p (cf. the Galerkin method [76]). This yields an approximation

$$p \approx \sum_{j=1}^{N} p_j \phi_j, \tag{4.35}$$

which allows us to approximate (4.34) in the form of a linear system of equations

$$\sum_{j=1}^{N} p_j^T \int_{T_0}^{T_1} \dot{\phi}_i \dot{\phi}_j \; dt \;\; = \;\; - \int_{T_0}^{T_1} \phi_i a^T \; dt + [\phi_i v^T]_{T_0}^{T_1} \;\; \forall \;\; i \in [1, N]. \tag{4.36}$$

In order to evaluate (4.36) further, we need to choose suitable basis functions ϕ_i. In this context, our choice for the basis functions are the "tent" functions

$$\phi_i = \begin{cases} (t - t_{i-1}) / h & t_{i-1} \leq t \leq t_i \\ 1 - (t - t_i) / h \quad , & t_i < t \leq t_{i+1} \quad \forall \;\; i \in [1, N], \\ 0 & \text{otherwise} \end{cases} \tag{4.37}$$

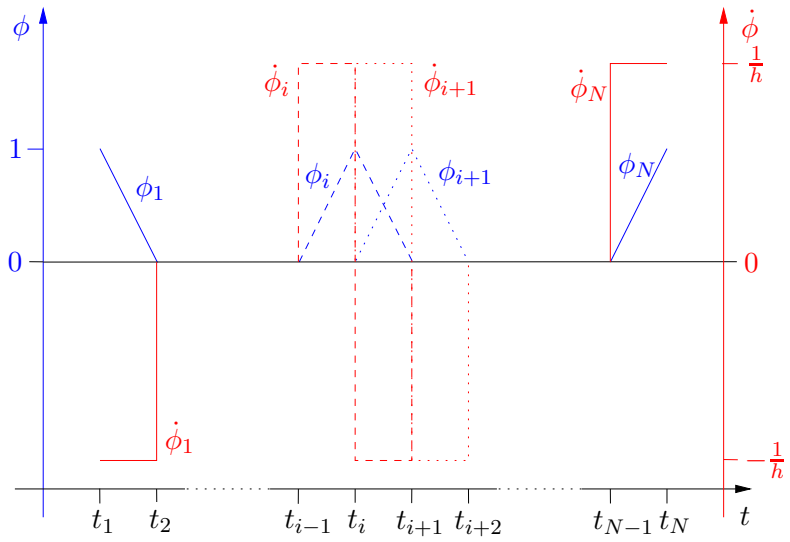

Figure 4.2: An illustration of the "tent" functions ϕ_i in blue and their derivatives $\dot{\phi}_i$ in red.

whose derivatives are

$$\dot{\phi}_i = \begin{cases} 1/h & t_{i-1} < t < t_i \\ -1/h & t_i < t < t_{i+1} \\ 0 & \text{otherwise} \end{cases} \quad \forall \ i \in [1, N]. \tag{4.38}$$

The basis functions ϕ_i and their derivatives $\dot{\phi}_i$ are illustrated in Figure 4.2. Note that the classic derivative of each $\dot{\phi}_i$ is not defined at nodes t_{i-1}, t_i, and t_{i+1}; however, this is not a problem since it is only required that the derivatives exist in the weak sense. In the case of the basis functions selected here, weak derivatives must exist, as the derivatives are square integrable [77].

We can now simplify (4.36) further by evaluating the integrals and the term evaluated at the boundary. For the integral on the left-hand side of (4.36), it holds that

$$\int_{T_0}^{T_1} \dot{\phi}_i \dot{\phi}_j \, dt = \begin{cases} 1/h & i = j = 1 \\ 2/h & i = j \neq 1 \\ -1/h & |i - j| = 1 \\ 1/h & i = j = N \\ 0 & \text{otherwise} \end{cases} . \tag{4.39}$$

The integral on the right-hand side of (4.36) can be evaluated by fitting a linear interpolant between every pair (a_{i-1}, a_i). This yields

$$\int_{T_0}^{T_1} \phi_i a \, dt = \begin{cases} h/6(2a_1 + a_2) & i = 1 \\ h/6(a_{i-1} + 4a_i + a_{i+1}) & i \in [2, N-1] \\ h/6(a_{N-1} + 2a_N) & i = N \end{cases} . \tag{4.40}$$

And finally for the boundary term, it holds that

$$[\phi_i v^T]_{T_0}^{T_1} = \begin{cases} -v_1 & i = 1 \\ 0 \quad , & 1 < i < N \\ v_N & i = N \end{cases} \tag{4.41}$$

At this point we have to proceed with care, as we do *not* want to include *any* information about the boundary values in the state equation. In fact, we do not necessarily have any knowledge of either v or p at the boundaries. Consequently, let us approximate [78] the initial and final velocity with

$$v_1 \approx \frac{1}{2h}(-3p_1 + 4p_2 - p_3) \tag{4.42}$$

$$v_N \approx \frac{1}{2h}(p_{N-2} - 4p_{N-1} + 3p_N). \tag{4.43}$$

The practical consequence of these approximations is that the proof presented above is no longer generally valid, as the boundary term (4.31) does not necessarily vanish. However, let us emphasize that this only applies to the general estimation case, where it cannot be guaranteed that the position or velocity at times T_0 and T_1 are known. If such information exists – as it does in any "traditional" two-point BVP – the presented approximation does not need to be applied, and the optimality criterion is satisfied. In this case, the FEM solution presented here yields optimal results in the sense that (4.24) is minimized. It is possible that this also happens in a more general context, but this possibility is not explored further in this thesis.

Now, by substituting all the evaluated terms into (4.36), the resulting state equation can be written in the form

$$\begin{bmatrix} -1 & 2 & -1 & & & & \\ -2 & 4 & -2 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & & -2 & 4 & -2 \\ & & & -1 & 2 & -1 \end{bmatrix} \begin{bmatrix} p_1^T \\ p_2^T \\ \vdots \\ p_{N-1}^T \\ p_N^T \end{bmatrix} = \frac{h^2}{3} B Z_s, \tag{4.44}$$

where

$$BZ_s = \begin{bmatrix} 2 & 1 & & & & & \\ 1 & 4 & 1 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & & 1 & 4 & 1 \\ & & & & 1 & 2 \end{bmatrix} \begin{bmatrix} (C_1 f_1 + g)^T \\ (C_2 f_2 + g)^T \\ \vdots \\ (C_{N-1} f_{N-1} + g)^T \\ (C_N f_N + g)^T \end{bmatrix}.$$

Notice that the first and last two rows of $J_s \in \mathbb{R}^{N \times N}$ are linearly dependent and thus, $\text{rank}(J_s) = N - 2$. This is a consequence of substituting (4.42) and

(4.43) into the boundary terms (4.41). The result of this is what we were looking for, i.e. the state equation no longer contains built-in assumptions about the boundary values and consequently, it is rank-deficient.

Note that the method presented above is not the only possible way to eliminate the boundary terms. Another, possibly more straightforward, way to achieve this is simply to bypass the integration by parts done in (4.33), which was that caused the unwanted boundary terms in the first place. This leads to a solution method which is commonly referred to as the *method of weighted residuals* [79]. However, this alternative is not studied further in this thesis.

4.2 The observation equation

This section presents the general principles for constructing the observation equation from the available observations. While the treatment is much the same as above, it does not reveal much about the properties of the resulting observation equation. This is because the particular realizations of the observation equation are highly case-dependent. At this point, we are only concerned with presenting a few generic examples of particular observations to provide the reader with some background to the observation equation. There are more examples of actual, realized observations in Chapter 6.

Let us start by presenting the basic "building blocks" of the observation equation (4.2), which are expressed in the form

$$g_i(x) = z_i \ \ \forall \ \ i \in [1, M], \tag{4.45}$$

where $g_i(x) \in \mathbb{R}$ and $x = \begin{bmatrix} y_1^T & \cdots & y_N^T \end{bmatrix}^T$. Then, the observation equation $f_o(x) : \mathbb{R}^{kN} \to \mathbb{R}^M$ is obviously of the form

$$\begin{bmatrix} g_1(x) \\ g_2(x) \\ \vdots \\ g_M(x) \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_M \end{bmatrix}. \tag{4.46}$$

The Jacobian of (4.46) is

$$J_o(x) = \begin{bmatrix} J_{1,1} & J_{1,2} & \cdots & J_{1,N} \\ J_{2,1} & J_{2,2} & \cdots & J_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ J_{M,1} & J_{M,2} & \cdots & J_{M,N} \end{bmatrix}, \tag{4.47}$$

where

$$J_{i,j} = \frac{\partial g_i}{\partial y_j} \in \mathbb{R}^{1 \times k} \ \ \forall \ \ i \in [1, M] \text{ and } j \in [1, N]. \tag{4.48}$$

Notice that in this case, no general statements can be made about the structure of J_o.

As mentioned above, at this point we shall demonstrate the treatment of individual observations with a few examples. The first, more basic, example provides us with a generic estimation problem with initial values. Assuming that a set of initial values y_0 is provided at time instance t_j, the respective observations are

$$Iy_j = y_0 \tag{4.49}$$

and the observation equation becomes

$$\begin{bmatrix} 0^{k\times k} & \cdots & 0^{k\times k} & I^{k\times k} & 0^{k\times k} & \cdots & 0^{k\times k} \end{bmatrix} \begin{bmatrix} y_1 \\ \vdots \\ y_{j-1} \\ y_j \\ y_{j+1} \\ \vdots \\ y_N \end{bmatrix} = y_0. \tag{4.50}$$

The second example concerns a single observation about the average speed

$$\overline{\|v\|}_2 = \frac{1}{T_1 - T_0} \int_{T_0}^{T_1} (v^T v)^{1/2}\, dt \approx \frac{1}{N} \sum_{j=1}^{N} (v_j^T v_j)^{1/2} \tag{4.51}$$

of the object during the time interval $[T_0, T_1]$. The non-zero components of the respective partial derivatives are

$$\frac{\partial \overline{\|v\|}_2}{\partial v_j} = \frac{1}{N} \frac{v_j^T}{(v_j^T v_j)^{1/2}} \quad \forall \;\; j \in [1, N]. \tag{4.52}$$

Finally, a set of observations related to the orthogonality constraints associated with the DAE (2.28), are discussed. This is necessary because the formulations presented in sections 4.1.1 and 4.1.2 have no built-in way to control the orthogonality of the estimated DCMs. For this purpose, we introduce so-called "pseudo-observations", which state that $C_i C_i^T = I$ for any given index i. Referring back to the solution methods listed on page 21, this method is similar to method (ii). In terms of the vectors c_i, the orthogonality constraint is realized as the following six observations:

$$\begin{bmatrix} c_1^T c_1 \\ c_2^T c_2 \\ c_3^T c_3 \\ c_1^T c_2 \\ c_1^T c_3 \\ c_2^T c_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.53}$$

In (4.53), the first three equations state that each column of the respective DCM is a unit vector. In turn, the last three equations of (4.53) state that the columns of the DCM are mutually orthogonal. The block of the Jacobian matrix related to (4.53) is

$$\begin{bmatrix} 2c_1^T & 0 & 0 \\ 0 & 2c_2^T & 0 \\ 0 & 0 & 2c_3^T \\ c_2^T & c_1^T & 0 \\ c_3^T & 0 & c_1^T \\ 0 & c_3^T & c_2^T \end{bmatrix}.$$

The pseudo-observations are actually implemented in the second example in Chapter 6.

4.3 Summary

This chapter discussed a number of ways to approximate the dynamics model in the form of a state equation. While it was decided to present the construction of the state equation with specific choices of numerical methods, this in no way means that they are the only possible choices. On the contrary, when it comes to general solution strategies, such as the relaxation method, the finite element method or the method of weighted residuals, there are already a number of potential choices to choose from. When it comes to selecting the discretization method used in each general solution strategy, the possibilities are literally boundless. It must be emphasized that while each of the methods considered here have their own merits, there is no particular reason to believe that they are necessarily the best possible choices for the task at hand. However, it can be stated that the approximations shown here have been extensively tested, and are sufficiently accurate and efficient.

While it is a good thing to have plenty of options, it is worth noting a few points about the choice of the state equation, particularly in the field of inertial navigation. Firstly, whatever numerical method is chosen, it only has an effect on the realization of the numerical discretization error. In other words, assuming that there is no additional information, there are no methods for automatically compensating for errors in the dynamics model. This is particularly important in the case of short-term inertial navigation, where the effects of the dicretization error are typically small compared to the effects of the measurement errors[2]. Therefore, there is not much sense in minimizing the discretization error beyond the point where it no

[2]Assuming, of course, that the dynamics model is based on IMU measurements.

longer plays a significant role in the total error. On the other hand, one should also ensure that the discretization error does not limit the potential accuracy of the system. Hence – as one might expect – the requirements for the numerical method increase along with the accuracy of the sensors.

The way we used the IMU measurements to construct the state equation significantly deviates from the typical approach. The standard way is to assume that the time evolution of the system is characterized with an external dynamics model (cf. item (c) on page 10) and the IMU measurements are used strictly as observations. Firstly, as explained above, the generic approach constructed here allows the user to build the state equation based on an external dynamics model. Secondly, it is important to realize that this requires application-specific information. And thirdly, not all processes are simple enough to be accurately modeled with an explicitly given ODE. However, the chosen ODE has an inherent effect on the obtained results. While based on the erroneous IMU measurements, the generic state equation often describes the dynamics of the system with fairly good accuracy as compared to an explicitly given ODE. Especially so, if the dynamics of the system is complicated, if the sensors are well-calibrated, and if the time interval is short. Thus, the availability of an externally given ODE does not automatically imply that it will improve the accuracy of the system.

If the computations involves a significant number of time instances t_i, the dimensions of the resulting equations will be substantial. Thus, if efficiency and memory requirements are of any concern, J_s should be as sparse a matrix as possible. In practice, this is not too much of a restriction, as a number of numerical methods – like the ones considered here – lead to such matrices. The requirement for sparse matrices merely rules out some specific solution strategies – such as FEM equipped with basis functions with global support – which lead to full matrix J_s. While there is really nothing we can do about the structure of the observation equation, the number of observations M is often reasonably small. On the other hand, if it so happens that J_s and J_o are both necessarily large matrices with few zero elements, the proposed method is certainly up to the task as long as the hardware is.

The next chapter goes on to discuss the details of finding a solution to the estimation problem we are about to pose. It can correctly be assumed that the next chapter presents the key elements of this thesis. The current chapter has provided a valuable input for this process as it gives the reader some idea of the contents of the state and observation equations. This is necessary in order to get a proper understanding of the concepts discussed below. It should now be understood that, in its presumed form, the state equation does not contain the necessary information for finding a unique solution to the problem. One can only say that the state equation provides a set of curves as a solution to the underlying ODE, but it does not point

out any particular member of the set.

Finally, the following treatment will deal with the state and the observation equations *together*[3]. While the constructed formalism does cover the trivial situations where an exact solution exists, most of the treatment is about the more interesting situation where there is no exact solution. Then, it really starts to make sense to talk about errors w_s and w_o (respectively W_s and W_o), as the available "redundancy" gives some room for manoeuvre against them. In another words, it is possible to *filter* the solution. The treatment begins with the easy situation, in which the properties of any supposedly unbiased errors are known. From there, the treatment moves onto more challenging "real-life" problems, in which the properties of the errors are not known.

[3]The laboratory calibration method – where the state equation does not exist – introduces the only exception to this.

Chapter 5

Solution methods

The aim of this chapter is use the given state and observation equations to pose the estimation problem; and then to solve it. When working with non-linear estimation problems, finding a *feasible solution* – i.e. a solution that satisfies a necessary, but not sufficient condition for optimality – is often the best one can do. With linear estimation problems, however, it is also possible to show that the obtained solution is unique and optimal in some well-defined sense. The treatment starts with linear estimation problems in section 5.1 and goes on to deal with non-linear estimation problems in section 5.2.

As neither the state nor the observation equation necessarily provide enough information on their own for the sought-after solution, the driving force behind this chapter is to find a way to *combine* the information contained in the two equations. This gives the concept of error a whole new meaning. The state and observation equations, of course, both contain many kinds of errors. Until now, however, there has not been much one could do about this, since error detection and/or reduction usually requires supplementary data. And this is exactly what can be achieved by combining the two equations, whenever the two equations contain redundant (or "overlapping") information.

Hereafter, it is assumed that $X \in \mathbb{R}^{kN \times q}$ ($q \in \mathbb{N}$) and $x \in \mathbb{R}^{kN}$. In both situations, it is assumed that the observation equation contains M observations. As we saw in the previous chapter, the number of rows in the state equation depends on the situation. As a working hypothesis, however, it is assumed that the state equation contains $k(N-1)$ rows. In this chapter, notations like J_i, Z_i with subscript i are repeatedly used. In the context of this chapter, this implies that the index $i \in \{s, o\}$.

5.1 Linear estimation problems

Let us start the treatment of linear estimation problems with a conceptually simple prototype situation, where q is assumed to be equal to 1 and the key statistical properties of the respective random vectors $w_s \in \mathbb{R}^{k(N-1)}$ and $w_o \in \mathbb{R}^M$ are known. That is, we start by deriving a solution to the case where

$$J_s x = z_s + w_s \ , \ \mathbb{E}(w_s) = 0 \ , \ \mathbb{V}(w_s) = \Sigma_s = \sigma_s^2 L_s L_s^T \tag{5.1}$$
$$J_o x = z_o + w_o \ , \ \mathbb{E}(w_o) = 0 \ , \ \mathbb{V}(w_o) = \Sigma_o = \sigma_o^2 L_o L_o^T . \tag{5.2}$$

In (5.1) and (5.2), $\mathbb{E}(w_i)$ represents the *expectation value* and $\mathbb{V}(w_i)$ the *variance* of w_i. The symmetric, positive definite, real-valued matrices Σ_i describe the variance-covariance matrices of the *uncorrelated random variables* w_s and w_o. Given that Σ_i is symmetric and positive definite, it can be represented in terms of *Cholesky decomposition* $\sigma_i^2 L_i L_i^T$, where L_i is a lower triangular matrix [66]. In this context, we assume that $\sigma_i^2 \in \mathbb{R}_+$ represents a scale factor[1] used to set $\left\|\Sigma_i^{-1}\right\|_{\max} = 1$, where $\|\cdot\|_{\max}$ returns the maximum absolute value among those appearing in the matrix $\cdot$. It must be emphasized that it is *not* assumed that the random vectors w_s and w_o are drawn from any particular distribution.

Now, as already mentioned, we wish to combine the state and observation equations. One way to do this is to form an *overdetermined* system of equations

$$\begin{bmatrix} J_s \\ J_o \end{bmatrix} x = \begin{bmatrix} z_s \\ z_o \end{bmatrix} \tag{5.3}$$

denoted as $Jx = z$ for short. When seeking a solution to such an equation, it should come as no surprise that there is no exact solution unless J has full rank and

$$z \in \mathcal{R}(J) \ \Leftrightarrow \ \mathcal{N}(J^T) = \{0\} ,$$

where $\mathcal{R}$ and $\mathcal{N}$ represent the *range* and the *null space* [80]. In a more general situation, an intuitive and commonly used approach is to find x as a solution of the minimization problem

$$\text{argmin}_x \ \|w\|_p = \text{argmin}_x \ \|Jx - z\|_p , \tag{5.4}$$

where

$$\|x\|_p = \left(\sum_{i=1}^{kN} |[x]_i|^p \right)^{1/p} \tag{5.5}$$

represents the p-norm ($p \in \mathbb{R}_+$) [71]. Deciding on p is a somewhat trickier business, as there are infinite possibilities to choose from, and often there

[1] When $\Sigma_i = L_i = I$, this scale factor is commonly referred to as the variance.

is no particular reason to favor one norm over another. In such cases, a frequently used choice is to go for the square norm, where $p = 2$ [80]. While it is hard to show that this particular choice is necessarily superior to any of the other possible choices, it does come with two significant advantages, i.e. it is good for statistical reasons (see below) and it is a simple computational problem [80]. On the downside, the Euclidean norm often performs poorly if there are outliers, in which case, for example, $p = 1$ is often a more robust choice [80].

Following from the choice $p = 2$, the *ordinary least squares* (OLS) solution of (5.3) is the solution of

$$(J_s^T J_s + J_o^T J_o)x_{OLS} = (J_s^T z_s + J_o^T z_o). \tag{5.6}$$

A unique solution

$$x_{OLS} = (J_s^T J_s + J_o^T J_o)^{-1}(J_s^T z_s + J_o^T z_o)$$

for (5.6) exists whenever matrix $(J_s^T J_s + J_o^T J_o)$ is invertible. This is characterized by the equivalent conditions

$$\text{rank}(\begin{bmatrix} J_s \\ J_o \end{bmatrix}) = kN \quad \Leftrightarrow \quad \ker(J_s) \cap \ker(J_o) = 0, \tag{5.7}$$

where "ker" refers to the kernel [54, 80]. Subsequently, unless otherwise noted, we shall assume that (5.7) holds.

Obviously, (5.6) represents a rather naive way to obtain a solution to the case depicted in (5.1) and (5.2). Indeed, the only situation where x_{OLS} presents an estimate of any statistical relevance is when $\sigma_s^2 = \sigma_o^2$ and $L_i = I$. That is, in the special case where w_i are of equal variance and where $\mathbb{E}(w_i^T w_i) = 0$. In this case, based on the Gauss-Markov theorem, x_{OLS} is known to be the *best linear unbiased estimate* (BLUE) [80,81]. In BLUE, the term best refers to the fact that $\|w\|_2$ from (5.4) is minimized (c.f. minimum mean squared error: MMSE [44]).

In a less specific, more generally applicable set-up – still assuming that $\mathbb{E}(w_s w_o^T)$ is zero – the components of w_i are correlated and of unequal variance. Then, in terms of the general, symmetric, positive definite matrices Σ_s and Σ_o,

$$\begin{bmatrix} J_s^T & J_o^T \end{bmatrix} \begin{bmatrix} \Sigma_s^{-1} & 0 \\ 0 & \Sigma_o^{-1} \end{bmatrix} \begin{bmatrix} J_s \\ J_o \end{bmatrix} x_{GLS} = \begin{bmatrix} J_s^T & J_o^T \end{bmatrix} \begin{bmatrix} \Sigma_s^{-1} & 0 \\ 0 & \Sigma_o^{-1} \end{bmatrix} \begin{bmatrix} z_s \\ z_o \end{bmatrix}, \tag{5.8}$$

where x_{GLS} is known as the *generalized least squares* (GLS) solution. In (5.8), since the covariance matrices are known to be invertible, a unique solution

$$x_{GLS} = (J_s^T L_s^{-T} L_s^{-1} J_s + \frac{\sigma_s^2}{\sigma_o^2} J_o^T L_o^{-T} L_o^{-1} J_o)^{-1}(J_s^T L_s^{-T} L_s^{-1} z_s + \frac{\sigma_s^2}{\sigma_o^2} J_o^T L_o^{-T} L_o^{-1} z_o)$$

exists whenever (5.7) holds. It is known that x_{GLS} is BLUE [81]. Notice then, that (5.8) can [80] be written in the equivalent form

$$\begin{aligned} x_{GLS} &= \text{argmin}_x \, (w_s^T \Sigma_s^{-1} w_s)^2 + (w_o^T \Sigma_o^{-1} w_o)^2 \\ &= \text{argmin}_x \, \left\| L_s^{-1}(J_s x - z_s) \right\|_2^2 + \frac{\sigma_s^2}{\sigma_o^2} \left\| L_o^{-1}(J_o x - z_o) \right\|_2^2 . \end{aligned} \quad (5.9)$$

It is worth noting that *when all the information required by x_{GLS} is available*, GSL is the method of choice for solving linear fixed-interval smoothing problems.

However, from our point of view, the GLS method contains one very practical problem. Namely, while it is easy to assume that Σ_s and Σ_o are given, it is a complicated problem to resolve their realizations in "real life" [82]. Especially so, since the covariance matrices tend to depend on a particular measurement. This thesis focuses on constructing a generic solution method suitable for all situations where the detailed statistical properties of w_s and w_o are not available.

Since the OLS solution depends on unrealistically strict assumptions and the GLS solution simply requires too much information, we will consider a situation located somewhere between these two extremes. Therefore, let us assume that Σ_s and Σ_o are diagonal matrices. This choice constitutes a *weighted least squares* (WLS) problem, whose solution x_{WLS} (BLUE) can be written in the form

$$x_{WLS} = (J_s^T L_s^{-2} J_s + \frac{\sigma_s^2}{\sigma_o^2} J_o^T L_o^{-2} J_o)^{-1} (J_s^T L_s^{-2} z_s + \frac{\sigma_s^2}{\sigma_o^2} J_o^T L_o^{-2} z_o), \quad (5.10)$$

where L_i is a diagonal matrix. When handling a WLS problem, it makes more sense to treat σ_s^2 and σ_o^2 separately from L_i. Then, L_s and L_o only contain information about the weighting coefficients with respect to an *internal reference*, whose weight is set to the value 1. It is considerably easier to do this than to render the variance of each individual equation into some globally meaningful units. Hence, to obtain comparable information about the applied weighting coefficients, L_i^2 has to be multiplied by σ_i^2.

The analysis presented in this section is based on the assumption that the errors are additive and applied to the right-hand-side terms z_i. With the observation equation, this assumption is valid as long as the models used to couple the erroneous observations to the state estimates are accurate. The situations presented in section (4.2) are examples of such models. In the case of the state equation, the assumption is valid as long as the dynamics equation is approximated with sufficient accuracy, and the measurement data only applies to z_s. However, sections 4.1.1 – 4.1.3 presented estimation schemes where the measurement data only applies to the coefficient matrix. In general, these situations can not be accurately modeled with the

formalism presented here. In such cases, tools like the *total least squares* (TLS) method are required [66, 83]. However, it must be remembered that the errors apply strictly to the non-zero elements of the matrix, whereas the TLS method assumes that the errors are evenly distributed amongst all the elements of the matrix [66,84]. Given these parameters, and considering the situations presented in sections 4.1.1 – 4.1.3, the model in which the errors are assumed to apply to the right hand side is used merely because it is a less incorrect option. In other words, how to best model the errors in the depicted situations is still open to question.

5.1.1 Problem formulation

Now, a comparison of equations (5.9) – (5.10) reveals that x_{WLS} is clearly the solution for the minimization problem

$$x_\lambda = \text{argmin}_x \ \left\|L_s^{-1}(J_s x - z_s)\right\|_2^2 + \lambda \left\|L_o^{-1}(J_o x - z_o)\right\|_2^2, \tag{5.11}$$

where $\lambda = \sigma_s^2/\sigma_o^2$. When written in this form, (5.11) can be recognized as a *Tikhonov regularization problem*, which is a much used tool in the field of *inverse problems* [48–50, 54]. The importance of this lies in the theory associated with inverse problems and Tikhonov regularization. The past few decades have witnessed the development of methods capable of detecting the *regularization parameter* $\lambda > 0$ without prior knowledge of the errors [54]. These methods allow us to infer the optimal solution for the problem (5.13) *without explicit knowledge of the ratio* σ_s^2/σ_o^2. This is the reason for using such an apparently complex method for defining σ_s^2 and σ_o^2. Since it is possible to infer the ratio σ_s^2/σ_o^2 from the context, the knowledge of the relative weighting coefficients provided in L_i is all that is needed.

So, in the end we are seeking a solution method for those linear fixed-interval smoothing problems in which the unknown X is a matrix. This being so, the vector norm employed in (5.11) is clearly insufficient. Therefore, let us consider a minimization problem in the form

$$X_\lambda = \text{argmin}_X \ \left\|L_s^{-1}(J_s X - Z_s)\right\|_F^2 + \lambda \left\|L_o^{-1}(J_o X - Z_o)\right\|_F^2, \tag{5.12}$$

where it is assumed that the errors associated with each column of Z_s and Z_o are drawn from the same distribution. Also, let us multiply J_i and Z_i by L_i^{-1} from the left. This yields an equation of the form

$$X_\lambda = \text{argmin}_X \ \|J_s X - Z_s\|_F^2 + \lambda \|J_o X - Z_o\|_F^2, \tag{5.13}$$

where we have – for notational convenience – used the same symbols J_i and Z_i: the weighting does not change the information content of the matrices. It should also be pointed out that, technically, this solution method does

not require the respective variance-covariance matrices Σ_i to be diagonal. Instead, it works for any symmetric, positive definite matrices. However, the diagonal form of the variance-covariance matrices ensures that the decorrelation (multiplication by L_i^{-1}) does not spoil the sparsity of the state equation, which is a crucial step in finding a computationally-feasible solution method.

So, here (5.13) is used to model linear fixed-interval smoothing problems. Notice that (5.13) can be equivalently written in the form

$$X_\lambda = \text{argmin}_X \ \text{Tr}(W_s^T W_s) + \lambda \text{Tr}(W_o^T W_o), \tag{5.14}$$

which is particularly useful when justifying the choice of the Frobenius norm to measure the errors. As it turns out, there are several good reasons for this choice. Firstly, as we will soon see, the Frobenius norm allows X_λ to be computed efficiently. Secondly, it is realized as the previously encountered 2-norm for vector-valued problems. And lastly – if the basic assumptions are valid – the chosen norm yields the BLUE solution for each q column of X_λ whenever $\lambda = \sigma_s^2/\sigma_o^2$ and the errors associated with each column are drawn from the same distribution. As with the vector-valued problems, which achieved the same result, this holds true because of the mutual independence of the columns

5.1.2 The existence and uniqueness of the solution

Now, the interesting questions regarding problem (5.13) are:

1. does it have a solution?
2. is the solution unique?
3. how can one find the solution?

In order to answer these questions, let us denote the respective cost function by

$$G_\lambda(X) = \|J_s X - Z_s\|_F^2 + \lambda \|J_o X - Z_o\|_F^2 . \tag{5.15}$$

We will prove the existence of a solution by first showing that $G_\lambda(X)$ is *convex*. Then, if the cost function has a *critical point*, it will be located at the *global minimum* of the problem (5.13) [5].

To show that $G_\lambda(X)$ is convex, we first point out that the domain of $G_\lambda : \mathbb{R}^{kN\times q} \to \mathbb{R}$ is convex. That is, if $X_1 \in \mathbb{R}^{kN\times q}$ and $X_2 \in \mathbb{R}^{kN\times q}$ hold $(X_1 \neq X_2)$, it follows that $\alpha X_1 + (1-\alpha)X_2 \in \mathbb{R}^{kN\times q}$ for any $\alpha \in (0,1)$.

Thus, we can write

$$\begin{aligned} G_\lambda\left[\alpha X_1 + (1-\alpha)X_2\right] &= \left\|\alpha J_s X_1 + (1-\alpha) J_s X_2 - Z_s\right\|_F^2 \\ &+ \lambda \left\|\alpha J_o X_1 + (1-\alpha) J_o X_2 - Z_o\right\|_F^2 \\ &= \left\|\alpha(J_s X_1 - Z_s) + (1-\alpha)(J_s X_2 - Z_s)\right\|_F^2 \\ &+ \lambda \left\|\alpha(J_o X_1 - Z_o) + (1-\alpha)(J_o X_2 - Z_o)\right\|_F^2 \\ &\leq \alpha \left\|J_o X_1 - Z_o\right\|_F^2 + (1-\alpha) \left\|J_s X_2 - Z_s\right\|_F^2 \\ &+ \lambda\alpha \left\|J_o X_1 - Z_o\right\|_F^2 + \lambda(1-\alpha) \left\|J_o X_2 - Z_o\right\|_F^2 \\ &= \alpha G_\lambda(X_1) + (1-\alpha) G_\lambda(X_2), \end{aligned}$$

which proves that G_λ is convex [85].

The critical point of (5.15) is the point where the Gâteaux derivative satisfies

$$\mathrm{d}G_\lambda(X, W) = \frac{d}{ds} G_\lambda(X + sW)\bigg|_{s=0} = 0 \ \ \forall \ \ W \in \mathbb{R}^{kN \times q}. \tag{5.16}$$

The Gâteaux derivative of the first term of $G_\lambda(X)$ is

$$\begin{aligned} & \frac{d}{ds} \left\|J_s(X + sW) - Z_s\right\|_F^2 \bigg|_{s=0} \\ = \ & \frac{d}{ds} \mathrm{Tr}\left[(J_s(X + sW) - Z_s)^T (J_s(X + sW) - Z_s)\right] \bigg|_{s=0} \\ = \ & 2\mathrm{Tr}\left[(J_s W)^T (J_s X - Z_s)\right]. \end{aligned}$$

Similarly, the Gâteaux derivative of the latter term is

$$\frac{d}{ds} \lambda \left\|J_o(X + sW) - Z_o\right\|_F^2 \bigg|_{s=0} = 2\lambda \mathrm{Tr}\left[(J_o W)^T (J_o X - Z_o)\right].$$

When put together, it follows that

$$\mathrm{Tr}\left[(J_s W)^T (J_s X - Z_s) + \lambda (J_o W)^T (J_o X - Z_o)\right] = 0 \ \ \forall \ \ W \in \mathbb{R}^{kN \times q},$$

which we can present in the form

$$\mathrm{Tr}\left[W^T \left((J_s^T J_s + \lambda J_o^T J_o) X - (J_s^T Z_s + \lambda J_o^T Z_o)\right)\right] = 0 \ \ \forall \ \ W \in \mathbb{R}^{kN \times q}.$$

Since this has to hold for an arbitrary W, it has to hold that

$$(J_s^T J_s + \lambda J_o^T J_o) X_\lambda = (J_s^T Z_s + \lambda J_o^T Z_o). \tag{5.17}$$

The critical point is unique as long as (5.17) has a unique solution. The condition for this is given in (5.7). Then, the critical point

$$X_\lambda = (J_s^T J_s + \lambda J_o^T J_o)^{-1} (J_s^T Z_s + \lambda J_o^T Z_o) \tag{5.18}$$

is the global minimizer of $G_\lambda(X)$.

Alternatively, X_λ can be obtained as the least squares solution of

$$\begin{bmatrix} J_s \\ \lambda^{1/2} J_o \end{bmatrix} X_\lambda = \begin{bmatrix} Z_s \\ \lambda^{1/2} Z_o \end{bmatrix}. \tag{5.19}$$

In terms of numerical stability, one would generally prefer (5.19) over (5.17) for solving X_λ [80]. However, in some applications (5.17) is realized in a tridiagonal form which allows one to find the solution very efficiently. Then, it may become a feasible option to use the *corrected semi-normal equations* method to obtain X_λ [80]. Such a situation often occurs in position estimation, for example, and this technique is investigated further in P3.

5.1.3 Choosing the regularization parameter

In this section, the objective is to find a systematical method for specifying the parameter λ. In the literature, the parameter-choice methods are generally based on the 2-norm rather than the Frobenius norm. The similar nature of the two norms indicates that the treatment can be readily extended to the Frobenius norm. According to [54], parameter-choice methods can be divided into two classes depending on their assumptions of $\|W_s\|_F$:

1. methods based on knowledge of $\|W_s\|_F$
2. methods that do not require $\|W_s\|_F$.

Of these two classes, our main interest is in class two as $\|W_s\|_F$ is, in general, not available.

In class two, there are three popular methods. These are the *quasi-optimality criterion* [54,86], the *generalized cross-validation* [54,87] and the *L-curve criterion* [54, 88, 89]. Let us now take a closer look at the L-curve method, which will be used here to choose the value of λ. The main reason for this choice is the need for a systematic way to estimate the unknown parameter λ. Secondly, the L-curve criterion seems to work well in the given circumstances. However, it is not really in the scope of this thesis to compare different parameter-choice methods.

The L-curve criterion is arguably quite an intuitive tool for the selection of λ. At the heart of this method is the shape of a certain curve (see Figure 5.1), which is parametrized as

$$[\rho(\lambda), \eta(\lambda)] = [\log \|W_s\|_F, \log \|W_o\|_F]. \tag{5.20}$$

The main idea behind using the L-curve criterion is to choose λ in such a way that there is a certain balance between $\|W_s\|_F$ and $\|W_o\|_F$. Specifically,

the L-curve criterion chooses the value of λ maximizing the curvature

$$\kappa(\lambda) = \frac{\dot{\rho}\ddot{\eta} - \ddot{\rho}\dot{\eta}}{\left[(\dot{\rho})^2 + (\dot{\eta})^2\right]^{3/2}} \tag{5.21}$$

of the curve (5.20) [54]. In (5.21), the dots represent first ($\dot{}$) and second ($\ddot{}$) derivatives of ρ and η with respect to λ. The location of the point with maximum curvature is shown in Figure 5.1, below. For a more complete discussion of the theoretical and practical aspects of the L-curve method, see, for example, [54].

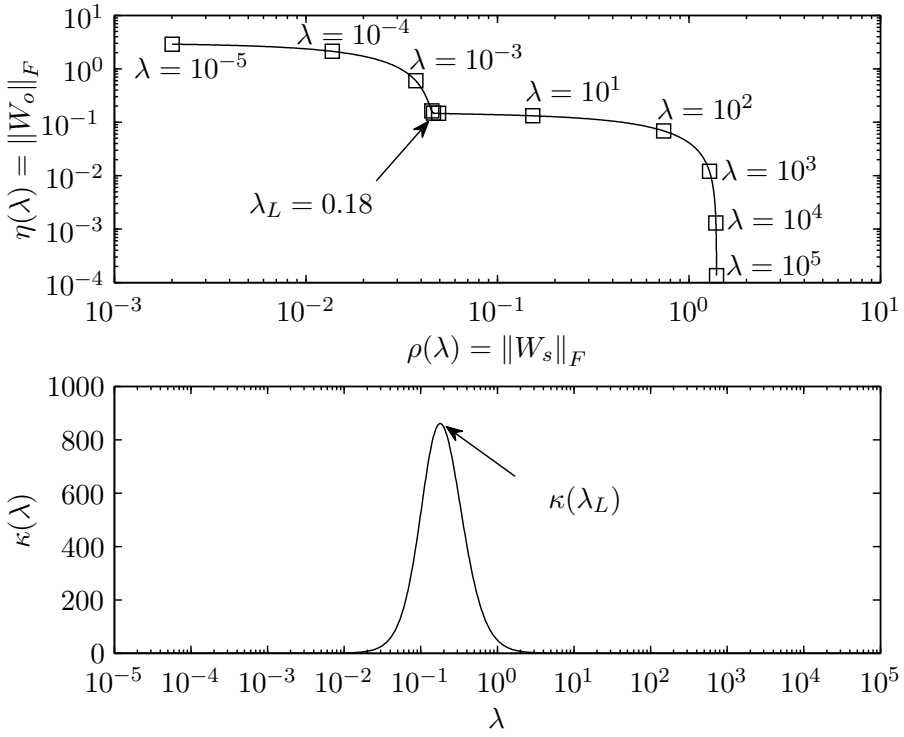

Figure 5.1: An example of the L-curve and its curvature.

The upper graph in Figure 5.1 presents an example of a typical L-curve. The squares drawn on the curve demonstrate the behavior of the curve as a function of λ. The lower graph shows the curvature of the L-curve, $\kappa(\lambda)$, plotted as a function of λ. The details of the problem yielding the L-curve shown here are described in Example 1 of publication P4. The horizontal part of the curve with the small values for the regularization parameter depicts a phase where the state equation is strongly preferred over the observation equation. In this phase, $\|W_s\|_F$ approaches zero rapidly as λ decreases. In turn, the vertical part of the curve with the large values for λ depicts a phase where the observation equation is strongly preferred

over the state equation. In this phase, $\|W_o\|_F$ approaches zero rapidly as the regularization parameter is increased. As seen from the figure, the actual L-shaped part of the curve is located between these two extremes.

The most interesting part of the curve is located near the corner. In the example, it takes place approximately between $0.01 \leq \lambda \leq 3$, where the curvature κ shown in Figure 5.1 clearly deviates from zero. Firstly, we notice the regular shape of the curvature function $\kappa(\lambda)$. In this case, practically any decent *line search algorithm* should be able to find λ_L, at least when its approximative location is known.

Secondly, it is important to notice that a wide range of regularization parameters end up close to the corner, as demonstrated by the squares shown in Figure 5.1. Based on this, reasonably accurate estimates of X_λ could be obtained even if the value used for λ does not coincide with λ_L. Therefore, if the aim is to maximize efficiency rather than accuracy, it seems a valid approach to use a predetermined value for λ for many particular problems. Otherwise, an iterative solution scheme is required, where $\kappa(\lambda)$ is evaluated at a number of points determined by the line search algorithm. In the following section, we propose a robust and computationally efficient way to evaluate $\kappa(\lambda)$, which forms the key part of the line search procedure.

5.1.4 Evaluating the curvature

Given a linear Tikhonov regularization problem and continuously differentiable functions

$$\begin{aligned} \rho(\lambda) &= \|J_s X_\lambda - Z_s\|_F = \|W_s\|_F \\ \eta(\lambda) &= \|J_o X_\lambda - Z_o\|_F = \|W_o\|_F \,, \end{aligned}$$

the derivatives shown in (5.21) can be computed analytically.

To begin with, let us compute the first derivative of $\|W_i\|_F$ with respect to λ. This yields

$$\begin{aligned} \frac{d}{d\lambda}\|W_i\|_F &= \frac{1}{2}\left[\mathrm{Tr}(W_i^T W_i)\right]^{-1/2} \frac{d}{d\lambda}\mathrm{Tr}(W_i^T W_i) \\ &= \left[\mathrm{Tr}(W_i^T W_i)\right]^{-1/2} \mathrm{Tr}\left(W_i^T \frac{dW_i}{d\lambda}\right). \end{aligned} \tag{5.22}$$

The second derivative is obtained by differentiating (5.22) with respect to

λ:

$$\begin{aligned}\frac{d^2}{d\lambda^2}\|W_i\|_F &= \frac{d}{d\lambda}\left[\left[\mathrm{Tr}(W_i^T W_i)\right]^{-1/2}\mathrm{Tr}\left(W_i^T\frac{dW_i}{d\lambda}\right)\right] \\ &= -\left[\mathrm{Tr}(W_i^T W_i)\right]^{-3/2}\left[\mathrm{Tr}\left(W_i^T\frac{dW_i}{d\lambda}\right)\right]^2 \\ &+ \left[\mathrm{Tr}(W_i^T W_i)\right]^{-1/2}\mathrm{Tr}\left(\frac{dW_i^T}{d\lambda}\frac{dW_i}{d\lambda}+W_i^T\frac{d^2W_i}{d\lambda^2}\right). \quad (5.23)\end{aligned}$$

In (5.22) and (5.23), it holds for the derivatives of W_i that

$$\frac{dW_i}{d\lambda} = \frac{d}{d\lambda}(J_i X_\lambda - Z_i) = J_i\dot{X}_\lambda \quad (5.24)$$

$$\frac{d^2W_i}{d\lambda^2} = J_i\ddot{X}_\lambda, \quad (5.25)$$

where $\dot{X}_\lambda$ and $\ddot{X}_\lambda$ represent the first and second derivative of X_λ with respect to λ.

Then, after some algebraic manipulation, the derivatives of $\rho(\lambda)$ and $\eta(\lambda)$ can be presented in the form

$$\begin{aligned}\dot{\rho}(\lambda) &= \|W_s\|_F^{-1}\,\mathrm{Tr}\left(W_s^T J_s\dot{X}_\lambda\right) \\ \dot{\eta}(\lambda) &= \|W_o\|_F^{-1}\,\mathrm{Tr}\left(W_o^T J_o\dot{X}_\lambda\right) \\ \ddot{\rho}(\lambda) &= \|W_s\|_F^{-1}\left[\mathrm{Tr}\left((J_s\dot{X}_\lambda)^T J_s\dot{X}_\lambda + W_s^T J_s\ddot{X}_\lambda\right) - \dot{\rho}(\lambda)^2\right] \\ \ddot{\eta}(\lambda) &= \|W_o\|_F^{-1}\left[\mathrm{Tr}\left((J_o\dot{X}_\lambda)^T J_o\dot{X}_\lambda + W_o^T J_o\ddot{X}_\lambda\right) - \dot{\eta}(\lambda)^2\right].\end{aligned}$$

In order to evaluate these, we need the values of $\dot{X}_\lambda$ and $\ddot{X}_\lambda$. For this, let us differentiate (5.17) with respect to λ. Then, by applying the result

$$\frac{dJ_i^{-1}}{d\lambda} = -J_i^{-1}\frac{dJ_i}{d\lambda}J_i^{-1}$$

from [90], the derivatives of X_λ can be written in the form

$$\dot{X}_\lambda = (J_s^T J_s + \lambda J_o^T J_o)^{-1}J_o^T(-W_o) \quad (5.26)$$

$$\ddot{X}_\lambda = (J_s^T J_s + \lambda J_o^T J_o)^{-1}J_o^T\left(-2J_o\dot{X}_\lambda\right). \quad (5.27)$$

Given (5.17), (5.26) and (5.27), it is easy to show that X_λ and its derivatives satisfy

$$\begin{bmatrix}J_s \\ \lambda^{1/2}J_o\end{bmatrix}\begin{bmatrix}X_\lambda & \dot{X}_\lambda & \ddot{X}_\lambda\end{bmatrix} = \begin{bmatrix}Z_s & 0 & 0 \\ \lambda^{1/2}Z_o & -\lambda^{-1/2}W_o & -2\lambda^{-1/2}J_o\dot{X}_\lambda\end{bmatrix} \quad (5.28)$$

in the least squares sense. Since the derivatives depend on X_λ, system (5.28) can not be directly solved. Instead, the "economy size" QR decomposition of the system matrix can be used [66]. Once it is available, each unknown of (5.28) can be efficiently resolved by exploiting the computed factorization.

Appendix A (p. 107) presents a Matlab® implementation of an algorithm which can be used to find X_λ for an unknown λ. Note that this algorithm takes the logarithms of the upper and lower limits of λ: the Matlab® function FMINBND works more efficiently when used this way.

Stopping criterion

Let us take a quick look at the possible ways to terminate whichever line search algorithm is used for finding λ_L. The implementation presented in Appendix A implies the use of the default tolerance

$$\left|\log(\lambda^{n+1}) - \log(\lambda^n)\right| < 10^{-4},$$

where λ^n denotes the nth iterate of λ. This is a rather vague way to terminate the iteration, as it is hard to come up with a decent a priori value for the tolerance. A considerably better method is to base the termination condition on the change in the solution X_λ, which enables us to terminate the iteration with more intuitive arguments. The easiest way to implement this is to invoke an a posteriori condition

$$\|X_{\lambda^{n+1}} - X_{\lambda^n}\|_p < \delta_X.$$

However, this requires us to evaluate $X_{\lambda^{n+1}}$ in order to know that X_{λ^n} would already have been enough. To avoid this small flaw, the knowledge of $\dot{X}_\lambda$ and $\ddot{X}_\lambda$ can be used to derive a corresponding a priori condition. Based on the Taylor series, it holds that

$$\begin{aligned} X_{\lambda^{n+1}} - X_{\lambda^n} &= \delta_\lambda \dot{X}_{\lambda^n} + \frac{1}{2}\delta_\lambda^2 \ddot{X}_{\lambda^n} + \cdots \\ \delta_X &\lesssim \delta_\lambda \left\| \dot{X}_{\lambda^n} \right\|_p + \frac{1}{2}\delta_\lambda^2 \left\| \ddot{X}_{\lambda^n} \right\|_p \end{aligned} \tag{5.29}$$

for all $\delta_\lambda = \left|\lambda^{n+1} - \lambda^n\right|$ [78]. Then, we can solve (5.29) for

$$\delta_\lambda \lesssim \left[-\left\| \dot{X}_{\lambda^n} \right\|_p + \left(\left\| \dot{X}_{\lambda^n} \right\|_p^2 + 2\delta_X \left\| \ddot{X}_{\lambda^n} \right\|_p \right)^{1/2} \right] / \left\| \ddot{X}_{\lambda^n} \right\|_p, \tag{5.30}$$

in which the negative root has been neglected. Then, the line search algorithm can be terminated once the realized $\delta_\lambda = \left|\lambda^{n+1} - \lambda^n\right|$ is smaller than the right hand side of (5.30). When implemented properly, this obviates the need to evaluate $X_{\lambda^{n+1}}$.

5.1.5 A few remarks

Ultimately, the analysis presented here on the choice of λ relies on the assumption that the curvature (5.21) has a unique and well-defined maximum. Whether this is the case can be tested with the *discrete Picard condition*, whose fulfillment has to be numerically confirmed [91]. Moreover, as J_s is often dependent on the measurements f and ω, this test needs to be performed separately for each application. Therefore, for the purposes of this thesis, this analysis holds no particular value. From experience, we can say that the L-curve criterion is a feasible technique for a surprisingly large class of problems. In practice, however, problems have been encountered in the following two situations:

- $$\mathrm{Tr}(W_s^T W_s) + \lambda \mathrm{Tr}(W_o^T W_o) \to 0$$

  When this happens, the presented algorithm has no means to adjust λ, because the error terms are close to zero. This can occur if the number of observations (M) is small or when all the available observations are temporally clustered around a single time instance. Then, the "effective number of observations" may be small even if the number of observations M is high. Of course, if the cost function reaches zero, then W_s and W_o are both necessarily zeros and λ becomes irrelevant.

- $$\mathrm{Tr}(W_s^T W_s) + \lambda \mathrm{Tr}(W_o^T W_o) \to \infty$$

  In this case, the state and the observation equations are highly contradictory, which has a tendency to reduce the value of $\mathrm{argmax}_\lambda\, \kappa(\lambda)$. This can lead to a multi-modal curvature $\kappa(\lambda)$ and/or unrealistic values for λ_L. More often than not, this indicates an incorrectly posed problem or a bug in the implementation.

While it holds that

$$\mathrm{Tr}(X_\lambda X_\lambda^T) = \mathrm{Tr}(X_\lambda^T X_\lambda),$$

it is the way that this is evaluated numerically that makes all the difference. Namely, $X_\lambda X_\lambda^T \in \mathbb{R}^{kN \times kN}$ while $X_\lambda^T X_\lambda \in \mathbb{R}^{q \times q}$. Moreover, as the trace only exploits the main diagonal of the input, it is seldom necessary and/or sensible to form the input matrix – which is generally dense – explicitly. Note that in the provided reference implementation, the latter consideration is not accounted for.

In section 4.1.3, the corresponding FDE was chosen in such a way that the state equation automatically preserves the orthogonality of the DCMs. Regrettably, this does not imply that the orthogonality constraints are fulfilled

by X_λ. To guarantee this, one should instead solve the problem

$$X_\lambda = \text{argmin}_X \ \|J_s X - Z_s\|_F^2 + \lambda \|J_o X - Z_o\|_F^2 \ \text{ where } \ C_i^T C_i = I \ \ \forall \ \ i \in [1, N].$$

This problem closely resembles the orthogonal Procrustes/Wahba's problem mentioned on page 21, although it has neither an analytical solution nor a known numerical solution method [66, 92, 93]. However, in many situations the form of the chosen J_s leads to solutions X_λ of problem (5.13) where the individual DCMs are almost orthogonal. Experience has shown that this is the case when the observation equation contains strictly orthogonal attitude estimates in reasonable agreement with each other. Here, the "reasonable agreement" is determined in terms of the state equation. To minimize any remaining orthogonality errors, a post-stabilization method like (2.33) can be employed.

It is also important to note that the method for choosing λ is merely one choice amongst a number of other possibilities. While we do not have any reason to doubt its capabilities – in fact, given the small amount of prior information, the employed method works amazingly well – we can not claim that it is *the best possible* way of determining λ in terms of some well-defined criterion. In this sense, the choice of λ has still not been completely resolved.

5.2 Non-linear estimation problems

This section proposes a method capable of solving non-linear fixed-interval smoothing problems. As there is no particular reason not to, the overall treatment of the non-linear problems largely follows the treatment introduced above. Then, the Frobenius norm of the error – now realized as the standard 2-norm – is minimized and we seek the WLS solution for the problem. Moreover, the regularization parameter λ is determined based on the curvature $\kappa(\lambda)$. In statistical terms, however, one should note that, unlike before, the obtained solution is certainly not BLUE, since the problem is no longer linear. And, more importantly, the WLS estimate obtained in the non-linear situation will no longer necessarily be unbiased [94].

As usual, non-linearity tends to make things harder as most of the convenient properties encountered in the linear case are no longer valid. Here, the key issue is that we can no longer guarantee the convexity of the cost function

$$G_\lambda(x) = \|f_s(x) - z_s\|_2^2 + \lambda \|f_o(x) - z_o\|_2^2 , \tag{5.31}$$

the form of which is based solely on its compatibility with its linear counterpart (5.15). Because of the (possible) non-convexity, the results for the existence and uniqueness of the solution derived for the linear estimation

problems are no longer valid. In this case, $G_\lambda(x)$ may have one or more global minimizers, or none at all. On the other hand, if one or more global minimizers do exist, there is no definitive way of recognizing whether the *iterative* solution method used found the global minimizer, or merely a local one. This occurs because we only have access to the necessary and sufficient [95] conditions for a *local* minima x^*_λ:

$$\nabla G_\lambda(x^*_\lambda) = 0 \text{ and } w^T \nabla^2 G_\lambda(x^*_\lambda) w > 0 \; \forall \; w \in \mathbb{R}^{kN}, \tag{5.32}$$

where $\nabla G_\lambda(x)$ represents the gradient and $\nabla^2 G_\lambda(x)$ the Hessian of the cost function. Here, we refer to x^*_λ as a feasible solution; in the case of non-linear estimation problems, this is often the best one can do. In practice, one can perform additional tests to see if the iterative solution method converges to the same feasible solution from different *initial guesses* x^0_λ. While a positive test result certainly provides us with new information regarding whether x^*_λ might actually be the global minimizer of G_λ, this obviously does not prove anything.

If a local minimum exists, it is up to the chosen solution method to find it. Most of the alternative solution methods do so, as long as the realized problem agrees "well enough" with the assumptions behind the chosen solution method, and we have access to an initial guess x^0_λ that is "reasonably close" to x^*_λ. It is indeed a matter of regret that this thesis cannot be more specific about this. However, we do have an influence on the "well enough" part by the choice of the solution method and on the "reasonably close" part by choosing the initial guess carefully.

It is often possible to compute a decent initial guess x^0_λ by solving a related linear estimation problem. The easiest way is to switch to a linear state equation and/or to zero out the weights associated with non-linear observations. The solution thus obtained is usually a good initial guess for the non-linear estimation problem. On the other hand, from experience we know that a zero initial guess is often sufficient to solve many practical non-linear estimation problems: and indeed, this is the case in the examples provided in chapter 6.

5.2.1 Problem formulation

The next step is to pose the optimization problem. Based on the cost function (5.31), the problem can be presented as follows:

$$\hat{x}_\lambda = \text{argmin}_x \; \|f_s(x) - z_s\|_2^2 + \lambda \|f_o(x) - z_o\|_2^2 . \tag{5.33}$$

Equation (5.33) can be written equivalently as

$$\hat{x}_\lambda = \text{argmin}_x \; w_s^T w_s + \lambda w_o^T w_o, \tag{5.34}$$

which makes it clear that the objective is again to find a solution which minimises the square norm of the errors.

Inspired by (5.19), obtained for the linear estimation problem, the solution for (5.33) is sought by solving

$$\begin{bmatrix} f_s(x_\lambda) \\ \lambda^{1/2} f_o(x_\lambda) \end{bmatrix} = \begin{bmatrix} z_s \\ \lambda^{1/2} z_o \end{bmatrix} \tag{5.35}$$

in the least squares sense. However, the solution for x_λ in the *substitute problem* (5.35) does not necessarily agree with the solution $\hat{x}_\lambda$ for the desired minimization problem (5.34). Generally speaking, $\mathbb{E}(x_\lambda) \neq \mathbb{E}(\hat{x}_\lambda)$, which agrees with the above-mentioned fact that the solution for the non-linear problem may be biased.

On the other hand, we do not have any obvious reason to doubt the rationale of the substitute problem: it seems to be a reasonable choice on its own. And more importantly, we do know how to solve (5.35) efficiently. Therefore, the substitute problem is used to tackle the non-linear estimation problems.

5.2.2 Solution method

As a non-linear least squares problem, (5.35) has several distinctive features:

- On the whole, the dimensions of the formed problem are noticeably large.
- Due to the expected form of $f_s(x_\lambda)$ and $f_o(x_\lambda)$, the respective Jacobian and Hessian matrices often contain only a few non-zero elements. Thus, despite the dimensions, optimization methods which depend on the Hessian are not automatically ruled out.
- Due to the sparsity of the coefficient matrices, the choice of the solution method is not necessarily limited only to those which are free of matrix operations.
- When solving (5.35) with particularly small or large values for λ, any optimization algorithm will eventually run into convergence problems. This is due to the inadequate scaling of (5.35). Similar problems will also occur when the range of the associated weighting coefficients in matrices L_i is too high.

In the end, it has to be accepted that the performance of different optimization methods will depend on the application. It is simply beyond the scope of this thesis to perform a complete survey of all the possible non-linear

optimization methods. Instead, we merely pick one method, which is then employed to solve the problems encountered in Chapter 6.

To find a solution to (5.35), we use the Gauss-Newton method with an additional line search procedure . While the method is easy to implement, it will converge more slowly than more advanced methods. The line search is used to find a step length which minimises the value of the cost function in the direction provided by Gauss-Newton. This is especially necessary when the current iterate is far from x_λ^*, in which case the steps taken using a plain Gauss-Newton do not necessarily decrease the value of the cost function. The slower convergence speed of Gauss-Newton is not usually a significant problem, as the optimization can often be initiated with a good initial guess. Especially so, when searching for an appropriate value for λ. In that case, a particularly good initial guess is provided by $x_{\lambda^{n+1}}^0 \approx x_{\lambda^n}^*$. [80, 96]

For the given regularization parameter λ and the initial guess x_λ^0, the proposed Gauss-Newton algorithm proceeds as follows:

1. Set $n = 1$.
2. Calculate increment Δx_λ as the least squares solution of
$$\begin{bmatrix} J_s(x_\lambda^{n-1}) \\ \lambda^{1/2} J_o(x_\lambda^{n-1}) \end{bmatrix} \Delta x_\lambda = \begin{bmatrix} z_s - f_s(x_\lambda^{n-1}) \\ \lambda^{1/2} \left(z_o - f_o(x_\lambda^{n-1}) \right) \end{bmatrix}$$
3. Minimize the cost function in the direction of Δx_λ by solving the line search problem
$$\alpha = \text{argmin}_\alpha \begin{bmatrix} f_s(x_\lambda^{n-1} + \alpha \Delta x_\lambda) - z_s \\ \lambda^{1/2} \left(f_o(x_\lambda^{n-1} + \alpha \Delta x_\lambda) - z_o \right) \end{bmatrix} \quad , \quad 0 < \alpha \leq 1$$
4. Set $x_\lambda^n = x_\lambda^{n-1} + \alpha \Delta x_\lambda$
5. If $n < n_{max}$ and
$$2 \left\| J_s^T(x_\lambda^n) \left(f_s(x_\lambda^n) - z_s \right) + \lambda J_o^T(x_\lambda^n) \left(f_o(x_\lambda^n) - z_o \right) \right\|_2 > \delta_\nabla$$
set $n = n + 1$ and go back to step 2. Otherwise, stop the iteration.

The iteration is stopped once the maximum amount of iterations n_{max} has been reached or when the gradient is less than the positive tolerance $\delta_\nabla << 1$, indicating that $x_\lambda^n \approx x_\lambda^*$.

The existence and uniqueness of Δx_λ computed in the second step of the solution algorithm is determined by condition (5.7). If Δx_λ fails to exist for a well-posed non-linear problem, a different initial guess can be used or – failing that – a more robust solution method. Whenever Δx_λ exists, the included line search procedure will then find a point where the value of the cost function does not increase.

5.2.3 Evaluating the curvature

The task of finding a suitable regularization parameter for non-linear estimation problems proceeds much as it did above in section 5.1.4. In fact, the algorithm presented here is exactly the same. This is possible if the non-linear estimation problem is first solved with the chosen optimization method. Then, the non-linear equation is linearized at the solution and converted into the form expected in section 5.1.4. Assuming that the solution method found x_λ^n, it holds for iterate x_λ^{n+1} that

$$\begin{bmatrix} J_s(x_\lambda^n) \\ \lambda^{1/2} J_o(x_\lambda^n) \end{bmatrix} x_\lambda^{n+1} = \begin{bmatrix} z_s - f_s(x_\lambda^n) \\ \lambda^{1/2}\left(z_o - f_o(x_\lambda^n)\right) \end{bmatrix} + \begin{bmatrix} J_s(x_\lambda^n) \\ \lambda^{1/2} J_o(x_\lambda^n) \end{bmatrix} x_\lambda^n. \tag{5.36}$$

In practice, the associated step Δx_λ should already be short, indicating that the linearization accurately predicts the behavior of the system at the (local) minimum x_λ^*. Then, the corresponding linear model can be obtained by setting

$$J_i = J_i(x_\lambda^n) \text{ and } Z_i = z_i - f_i(x_\lambda^n) + J_i(x_\lambda^n) x_\lambda^n.$$

While the linearization strategy presented here has been observed to work well in a number of applications, this does not imply that it necessarily works in every situation. It is, however, beyond the scope of this thesis to determine all the conditions under which this strategy works. Certainly, the existence of an L-shaped curve is required, but the non-linearity of the underlying estimation problem is likely to cause further constraints. On the whole, though, the remarks made for the linear estimation problems in section 5.1.5 are also applicable here.

5.3 Summary

This chapter presented the best linear unbiased estimators (BLUE) suitable for three linear estimation problems. In the most basic situation, the included errors are assumed to be uncorrelated and of equal variance. This constitutes an ordinary least squares problem (OLS). In a slightly more generalised case, the included errors are uncorrelated and of unequal variance. This is commonly referred to as the weighted least squares (WLS) problem. In the most general situation, the errors are correlated and of unequal variance, and this is known as the generalized least squares problem (GLS). Whenever $\mathbb{E}(w_s w_o^T) = 0$, each of the three estimators can be obtained by solving

$$(J_s^T L_s^{-T} L_s^{-1} J_s + \lambda J_o^T L_o^{-T} L_o^{-1} J_o) x_{LS} = (J_s^T L_s^{-T} L_s^{-1} z_s + \lambda J_o^T L_o^{-T} L_o^{-1} z_o)$$

for x_{LS}, where $\lambda = \sigma_s^2/\sigma_o^2$. It is the realization of L_i that defines which one of the three estimators is used. It is important to note that the distribution of the errors does not change the fact that the obtained x_{LS} is BLUE. In the case of a normal (Gaussian) distribution, the BLUE coincides with the maximum likelihood estimate [80]. In the case of non-linear estimation problems, the statistical properties of the solution were not specified.

While the GLS estimator seems to yield the most accurate results, it does require a lot of information about the errors. Moreover, it is by no means a computationally cheap task to find the GLS solution: when the dimensions are large, the solution process will inherently require a lot of memory and floating-point operations. With the task at hand, the number of unknowns is very easily measured in the order of 10^5. In this sense, the WLS method is what ultimately makes the proposed method a computationally feasible option. The key concept of the proposed solution method is that the parameter λ can be inferred from the context. Thus, the user does not need to know the correct balance between the state and observation equations, which is often a demanding task in practical situations. Ideally, if the modelling decisions are valid and the L-curve method is able to infer the correct value for λ, the proposed method produces the BLUE solution.

The Matlab® implementation of an algorithm suitable for finding the pair λ, X_λ for linear estimation problems demonstrates that the method can be easily implemented in a high-level language. While the solution process for non-linear problems is very similar, no reference implementation was provided as the iterative solution method which was used is merely a simple example, rather than a method suitable for reference purposes.

The models used here allow one to determine the weight for each row of the state and observation equations individually. With the observation equation, this possibility is regularly exploited in the examples presented in Chapter 6. For the state equation, having the option of altering the weights is particularly handy if some parts of the state equation are known to be problematic. In practice, this could be caused by the sensors having insufficient range at (short) time transients, or perhaps because the sensor has insufficient bandwidth to accurately measure some high-frequency events. On the other hand, the whole state equation can be reweighed in line with the accuracy of the used FDEs. Of course, this assumes that it is possible to evaluate the local error caused by the particular FDE used.

Whenever there is the freedom to reweigh individual equations, there is always the chance that it will cause numerical problems. This is of particular importance with non-linear estimation problems, where numerical problems may result in the used iterative solution method suffering from divergence or stagnation. In its present form, the solution method proposed here is not particularly well-suited for handling exact observations. For example,

this means that we can only associate the pseudo-observations with a finite weight, even though they are used to model constraints. Moreover, this solution method is not capable of handling (in)equality constraints, which often occur with linear least squares problems. So, the treatment of (non)linear (in)equality constraints with the proposed solution method remains open to question.

Chapter 6

Examples

In this chapter, the solution method is applied to two real-life examples using the actual measured data. The first example concerns a conceptually simple 3 DOF problem for determining the trajectory of a curling stone [10] on a plane defined by the ice track. The second example presents a full 6 DOF problem for determining the trajectory of a ski jumper [8]. The goal of this chapter is to illustrate the use of the proposed solution method in realistic environments. The purpose of the results presented here is to demonstrate the success of the calculations rather than to draw specific conclusions about the cited examples. The reader is presented with all the information necessary to understand how the situations under study can be presented in the form of the state and observation equations. For the purposes of this thesis, it is not necessary to discuss the detailed reasoning behind choosing the examples or the observations used.

Because true measurement data is used, the actual trajectory of the object is not available in either example. Thus, the accuracy of the obtained results can only be assessed by inspecting the provided graphs. Further information about the absolute accuracy of the proposed method in simulations can be found in the attached publications.

It is also important to note that the examples presented here can not be classified as "traditional" IVPs, as neither of them includes an explicit statement of the attitude at any instance of time. In fact, the first example refers only to the attitude difference between two time instances and the second example does not provide any observations related to the attitude. Moreover, the second example does not provide any observations about the velocity of the object.

In each example, the average sample rate of the 20-bit IMU which was used was approximately 800 Hz. The sample rate achieved was not constant, and

Parameter	ADXRS300	SCA610	SCA620
Dynamic range	± 300 °/s	± 3 g	± 12 g
Sensitivity	5 mv/°/s	0.75 V/g	0.15 V/g
Sensitivity error	± 8 %	± 5 %	± 5 %
Bias error	± 40 °/s	± 195 mg	± 700 mg
Non-linearity	0.1 %	2.0 %	2.5 %
Bandwidth	40 Hz	115 ± 55 Hz	400 ± 150 Hz
Linear acc. effect	0.2 °/s/g	N/A	N/A
Bias instability	32.31 °/h	3.88 m/s/h	17.33 m/s/h
Averaging time	140 s	20 s	10 s
Random walk	3.06 °/$\sqrt{h}$	0.12 m/s/$\sqrt{h}$	0.55 m/s/$\sqrt{h}$

Table 6.1: Specifications of the sensors used in the experiments. Sensitivity and bias errors are specified over the full operational temperature range of the sensors. The values under the horizontal line in the middle are obtained from Allan deviation plots of the respective sensor models devised by the author, and these are presented in Figure 6.1.

the time between adjacent samples varies within $[1, 1.5]$ milliseconds. In both examples, Analog Devices' ADXRS300 [97] gyroscopes were used to obtain samples of the angular velocity ω. The samples of specific force f were measured with the model VTI SCA610-CC5H1A accelerometer [98] in the curling example, and the model VTI SCA620-CHCV1A [99] in the ski jump example. The IMUs were calibrated for their full dynamic range as described in Chapter 3. No online calibration was carried out. The key specifications of the sensors are shown in Table 6.1.

The values under the middlemost horizontal line in Table 6.1 are determined from the Allan deviation plots displayed in Figure 6.1. The symbols $\sigma_\omega(\tau)$ and $\sigma_f(\tau)$ represent the Allan deviations of the gyroscope and of the two accelerometer models, respectively. The data for the Allan deviation plots was obtained by observing the output of the sensor models for 2.5 hours with a constant sampling rate of 100 Hz. The test was performed at room temperature without specific temperature control. As the table shows, the bias instability value of the gyroscope is observed at an averaging time of $\tau = 140$ seconds. For the accelerometers, the respective averaging times are $\tau = 20$ (SCA610) and $\tau = 10$ (SCA620) seconds.

Both of the examples under consideration are non-linear and the value for the regularization parameter is unknown. The line search problem related to the maximization of the curvature is solved using the stopping criterion

$$\delta_\lambda = \left|\log(\lambda^n) - \log(\lambda^{n-1})\right| < 10^{-3}.$$

The initial bounds set for the regularization parameter λ are $\lambda^0_{min} = 10^{-4}$

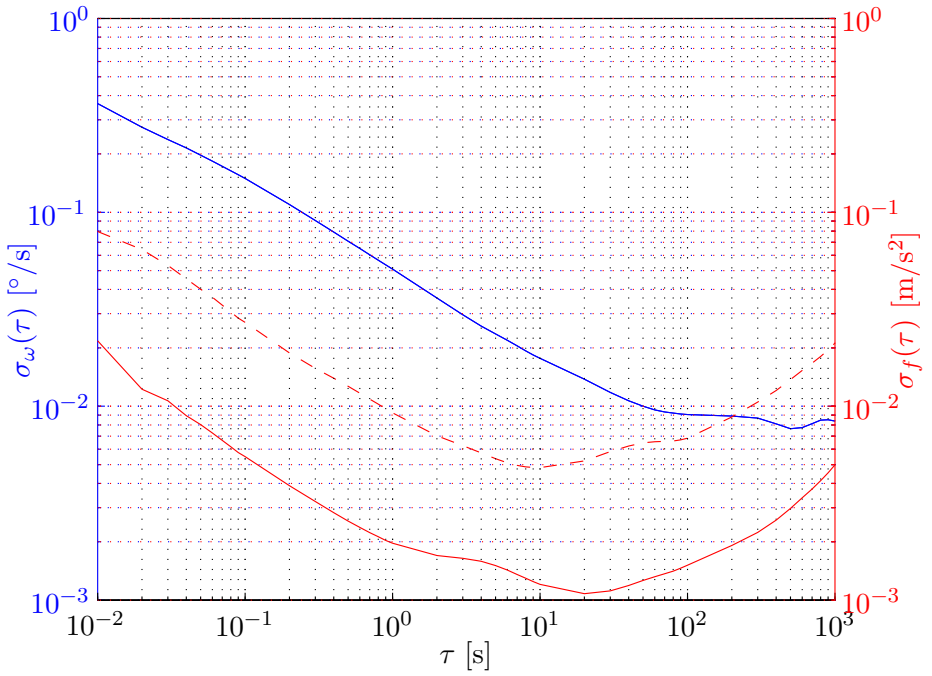

Figure 6.1: Allan deviations of the used sensors. The displayed graphs represent a typical performance level of the sensor models used in the examples. The solid red line represents accelerometer model SCA610 and the dashed line is for model SCA620. This graph is only used to demonstrate the quality of the sensors: the information in these graphs is not used in the examples.

and $\lambda^0_{max} = 10^4$. In the curling example, the initial guess is $x^0_{\lambda^0} = 0$, while in the ski jump example, the "closest analog" where $x^0_{\lambda^0}$ is constructed with

$$C_i = I \;\text{ and }\; v_i = p_i = 0 \;\;\forall\;\; i \in [1, N]. \tag{6.1}$$

The reason these initial values were chosen was to demonstrate that the proposed solution method *can* – if the situation so requires – be successfully employed even with such vague initial data. Naturally, if more accurate information concerning λ^0_{min}, λ^0_{max}, and $x^0_{\lambda^0}$ is available, one should always use it.

Within the nth step of the curvature maximization, a non-linear fixed interval smoothing problem is solved with a given λ^{n-1}. This is done with the algorithm shown in section 5.2.2 with a constant stopping criterion of $\delta_\nabla = 10^{-8}$. It is not always necessary to use such a precise stopping criterion. However, the benefit gained from using too loose a stopping criterion δ_∇ can be canceled out by the additional steps involved in maximizing the curvature: the evaluation of the curvature being based on *local* linearization of the non-linear problem. On the other hand, once the problem has been solved for

$x^*_{\lambda^0}$, the subsequent steps usually converge within a few iterations even with the seemingly strict stopping criterion. This can be verified from Tables 6.2 and 6.3, which show the convergence results of the two examples. The line search problem used in the Gauss-Newton algorithm is solved with the stopping criterion

$$\delta_\alpha = \left|\alpha^n - \alpha^{n-1}\right| < 10^{-6}.$$

The function FMINBND implemented in Matlab® is used for this with initial bounds $\alpha^0_{min} = 0$ and $\alpha^0_{max} = 2$. These bounds were used in order to set the "expected" step length $\alpha = 1$ (the one given by Gauss-Newton) in the middle of the interval.

6.1 Example 1: Curling

In this example, the IMU is mounted on a curling stone with the objective of estimating the planar trajectory of the stone during the event, which consists of a single throw. The basic assumption is that the ice track forms a plane aligned with the local horizontal plane, which yields a 3 DOF inertial navigation problem. The x-coordinate of the local geographic frame is chosen to be aligned with the long side of the rectangular ice track (see Figure 6.2). Moreover, the event is assumed to begin and end at a known location with zero velocity. The local gravitational acceleration points away from the plane and is thus not relevant in this particular example.

The state equation to be used can be derived from the more general case (2.26) by neglecting the vertical coordinate. The resulting 2×2 rotation matrix is also written in terms of the heading angle θ, as this automatically satisfies the orthogonality constraint up to a negligible round-off error. The resulting dynamics model is

$$\begin{bmatrix} \dot{\theta} \\ \dot{v}_x \\ \dot{v}_y \\ \dot{p}_x \\ \dot{p}_y \end{bmatrix} = \begin{bmatrix} 0 \\ \cos(\theta) f_x - \sin(\theta) f_y \\ \sin(\theta) f_x + \cos(\theta) f_y \\ v_x \\ v_y \end{bmatrix} + \begin{bmatrix} \omega_z \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{6.2}$$

and the respective Jacobian

$$\frac{\partial f(t,y)}{\partial y} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ -\sin(\theta) f_x - \cos(\theta) f_y & 0 & 0 & 0 & 0 \\ \cos(\theta) f_x - \sin(\theta) f_y & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}. \tag{6.3}$$

The zero velocity of the stone at both ends of the event allows one to compensate for the bias offsets of the three sensors which were used in this example. In practice – assuming that there is no information available about the stochastic properties of the errors – the bias offsets are set to the mean of the respective offsets at the two ends of the measurement. Each offset is determined by averaging the respective samples over a time interval of ≈ 5 seconds.

6.1.1 Observation equation

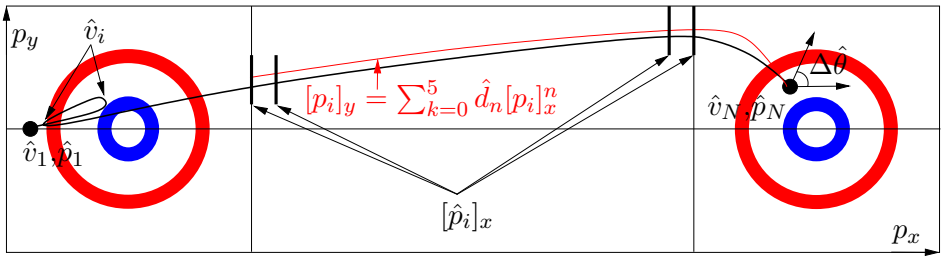

Figure 6.2: An illustration of the observations used in the curling example. The figure is not in scale and does not correspond to any particular measurement.

The following observations are utilised in this example:

(a) the initial ($\hat{p}_1$) and final ($\hat{p}_N$) position of the stone

(b) the velocity ($\hat{v}_i$) of the stone at four different time instances t_i

(c) the difference ($\Delta\hat{\theta}$) between the final and initial heading angles of the stone

(d) the x-coordinate ($[\hat{p}_i]_x$) of the stone at four different time instances t_i during the throw, and

(e) the estimated trajectory

$$[p_i]_y = \sum_{n=0}^{5} \hat{d}_n [p_i]_x^n, \quad \hat{d}_n \in \mathbb{R} \ \ \forall \ \ n \in [0, 5]$$

of the stone.

Observations (a)–(d) are treated as equally reliable, which is realized by setting the corresponding elements of L_o^{-1} to the value 1. It is worth emphasizing here that these observations are known to be accurate. The approximate uncertainty in the position observations was ± 1 cm and, neglecting the rotation of the Earth, the velocity observations are exact. Observation (e) is considered to be a significantly less reliable source of information, so this

was realized by setting the respective elements of L_o^{-1} to the value 10^{-4}. Notice that the weights only provide information about the differences in the reliability of the rows in the observation equation. The overall reliability of the state equation with respect to an observation equation is determined by the unknown regularization parameter λ.

The velocity of the stone is known to be zero at four distinct phases of the throw. In addition to the initial and final velocity of the stone, the event contains two other phases where the velocity of the stone is known to be momentarily zero. See the sketch of the stone's trajectory in Figure 6.2 for the approximate locations of these events. The loop next to the stone on the left of the sketch represents the "initialization" phase of the event, i.e. the first few seconds during which the athlete prepares for the upcoming throw. The velocity is momentarily zero at the two extreme ends of this loop. The time indices corresponding to these moments were extracted from the measurement data by hand. In the example data, shown below in Figure 6.3, these two moments occur approximately at the times $t = 2.5$ and $t = 5.0$ seconds.

Observation (c) is utilised because it is significantly easier to implement in practice than the initial and final attitudes, which are dependent upon the inner alignment of the IMU with respect to the stone. This is denoted by the angle $\Delta\hat{\theta}$ shown in Figure 6.2. Type (d) observations were taken using four permanently magnetised strips aligned with the y-coordinate at locations

$$p_x = \begin{bmatrix} 0.000 & 0.295 & 21.950 & 22.245 \end{bmatrix} \text{ m}.$$

The four thick vertical lines in Figure 6.2 show the locations of the magnetic strips on the ice track. The respective time instances are isolated by hand using the output from a vertically aligned magnetometer installed in the IMU.

All observations concerning the heading ($\hat{\theta}_i$), velocity ($\hat{v}_i$) or position ($\hat{p}_i$) of the curling stone at a time instant t_i can be taken into account with an equation of the form

$$\begin{bmatrix} w_\theta & 0 & 0 \\ 0 & w_v I & 0 \\ 0 & 0 & w_p I \end{bmatrix} \begin{bmatrix} \theta_i \\ v_i \\ p_i \end{bmatrix} = \begin{bmatrix} w_\theta \hat{\theta}_i \\ w_v \hat{v}_i \\ w_p \hat{p}_i \end{bmatrix}, \tag{6.4}$$

where $I \in \mathbb{R}^{2\times 2}$ and the respective elements of L_o^{-1} are denoted by $w_\theta, w_v, w_p \in \mathbb{R}_+$. Observations concerning the change in the same variables between time instances t_i and t_j are accounted for with equations of the form

$$\begin{bmatrix} w_\theta & 0 & 0 \\ 0 & w_v I & 0 \\ 0 & 0 & w_p I \end{bmatrix} \left(\begin{bmatrix} \theta_i \\ v_i \\ p_i \end{bmatrix} - \begin{bmatrix} \theta_j \\ v_j \\ p_j \end{bmatrix} \right) = \begin{bmatrix} \Delta\hat{\theta} \\ \Delta\hat{v} \\ \Delta\hat{p} \end{bmatrix}. \tag{6.5}$$

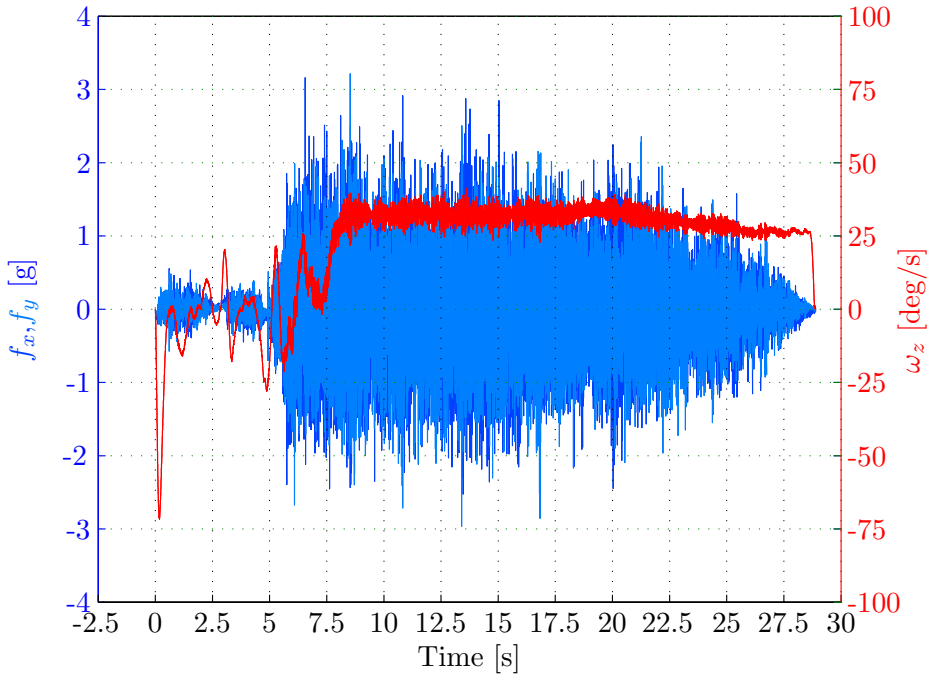

Figure 6.3: The samples of specific forces and angular velocities gathered in measurement 1.

The trajectory estimate concerns that part of the trajectory where the motion of the stone is only governed by frictional forces. In Figure 6.3, the start of this phase of the throw is most easily recognized from the angular velocity plot, which is nearly constant after $t \approx 8.0$ seconds. The red graph in Figure 6.2 illustrates the shape of the trajectory estimates. The coefficients $\hat{d}_n$ which define the form of the trajectory estimate, are determined from knowledge of the initial and final points of the stone and the expected shape of the trajectory. Each measurement uses different coefficients. The degree of the polynomial used to represent the trajectory (5) is the lowest possible which would still produce a reasonable estimate of the trajectory. The respective observation of weight $w_{tr} \in \mathbb{R}_+$ at time instance t_i can be written in the form

$$w_{tr}\left(\sum_{n=1}^{5} \hat{d}_n [p_i]_x^n - [p_i]_y\right) = -w_{tr}\hat{d}_0, \tag{6.6}$$

and the respective block of the Jacobian is

$$J_{\cdot,i} = w_{tr}\begin{bmatrix} 0 & 0 & 0 & \sum_{n=1}^{5} n\hat{d}_n [p_i]_x^{n-1} & -1 \end{bmatrix}, \tag{6.7}$$

where $J_{\cdot,i}$ refers to an arbitrary row. The knowledge of the trajectory was used with every 100th recorded sample, which yielded 168 trajectory observations in the measurement shown in Figure 6.3.

	Measurement							
	1	2	3	4	5	6	7	8
$\log \lambda^1$	−0.9(13)	−0.9(17)	−0.9(14)	−0.9(18)	−0.9(13)	−0.9(16)	−0.9(22)	−0.9(18)
$\log \lambda^2$	0.944(4)	0.944(4)	0.944(4)	0.944(7)	0.944(4)	0.944(5)	0.944(3)	0.944(5)
$\log \lambda^3$	2.111(5)	2.111(5)	2.111(5)	2.111(7)	2.111(5)	2.111(6)	2.111(5)	2.111(7)
$\log \lambda^4$	2.833(5)	2.833(6)	2.833(5)	2.833(7)	2.833(4)	2.833(5)	2.833(6)	2.833(7)
$\log \lambda^5$	1.961(5)	2.044(6)	2.279(5)	1.958(7)	2.021(5)	2.082(5)	3.707(6)	2.159(6)
$\log \lambda^6$	1.914(3)	2.308(5)	2.411(3)	1.600(5)	2.165(3)	2.227(4)	3.171(5)	2.379(5)
$\log \lambda^7$	2.007(3)	2.509(5)	2.356(3)	1.973(5)	2.109(2)	2.156(3)	2.557(5)	2.553(4)
$\log \lambda^8$	2.004(3)	2.237(5)	2.357(1)	1.977(3)	2.107(1)	2.158(2)	2.390(5)	2.353(5)
$\log \lambda^9$	2.004(2)	2.237(3)	2.355(1)	1.975(3)	2.108(1)	2.156(1)	2.605(5)	2.349(3)
$\log \lambda^{10}$	2.005(2)	2.236(3)	2.355(1)	1.976(3)	2.107(1)	2.157(1)	2.581(4)	2.350(3)
$\log \lambda^{11}$		2.188(3)		1.975(3)		2.156(1)	2.582(3)	2.349(3)
$\log \lambda^{12}$		2.236(3)					2.581(3)	
$\log \lambda^{13}$		2.235(3)						
λ_L	101.051	171.941	226.506	94.419	127.979	143.199	380.829	223.404
$\kappa(\lambda_L)$	8405.88	7798.57	5614.45	6954.33	9773.32	7532.62	7668.68	10562.8
δ_λ	2.2610	3.8000	4.5419	2.4086	3.6996	3.6782	10.013	5.0506

Table 6.2: Convergence of the line search procedure used to find λ_L in the curling example. The numbers in parentheses represent the number of Gauss-Newton steps required within each iteration. Iterations below the additional horizontal line in each column represent the unnecessary iterations in terms of the stopping criterion δ_λ.

6.1.2 Results

As shown in Table 6.2, between 10 and 13 iterations along the L-curve are required for convergence. The values obtained for λ_L range from 94.419 to 380.829. Remembering that $\lambda_L = 1$ represents an equal weight among the two sources of information, the obtained values indicate a clear preference for the observation equation. Given knowledge of the overall accuracy of the observations, this is of course a reasonable preference. Given that absolutely no information about the reliability of the state and observation equations was used in the solution method, this is a very respectable achievement.

As can be seen from the table, the solution of this non-linear problem for λ^1 requires from 13 to 22 Gauss-Newton iterations. Such a high number of iterations is due to the facts that the initial guess $x^0_{\lambda^0} = 0$ was practically worthless and, as expected, Gauss-Newton converged slowly when used far away from the optimum x^*_λ [96]. For λ^2, Gauss-Newton required between 3–7 iterations to converge, and this gradually fell to 1–3 iterations for the final iterate λ_L. These figures clearly illustrate the fact that the lower the value of δ_λ, the better the quality of the available initial value. Then, the number of Gauss-Newton steps required for convergence decreases as the iteration proceeds.

These results indicate that no accurate prior knowledge of λ^0_{min}, λ^0_{max} and x^0_λ is necessary. On the other hand, if such knowledge is available, this allows us to decrease the computational cost of finding the solution. Firstly, as $\lambda^0_{max} - \lambda^0_{min} \to 0$, fewer iterates λ^n are required to get sufficiently close to λ_L. This is significant, as about half of all Gauss-Newton steps are taken

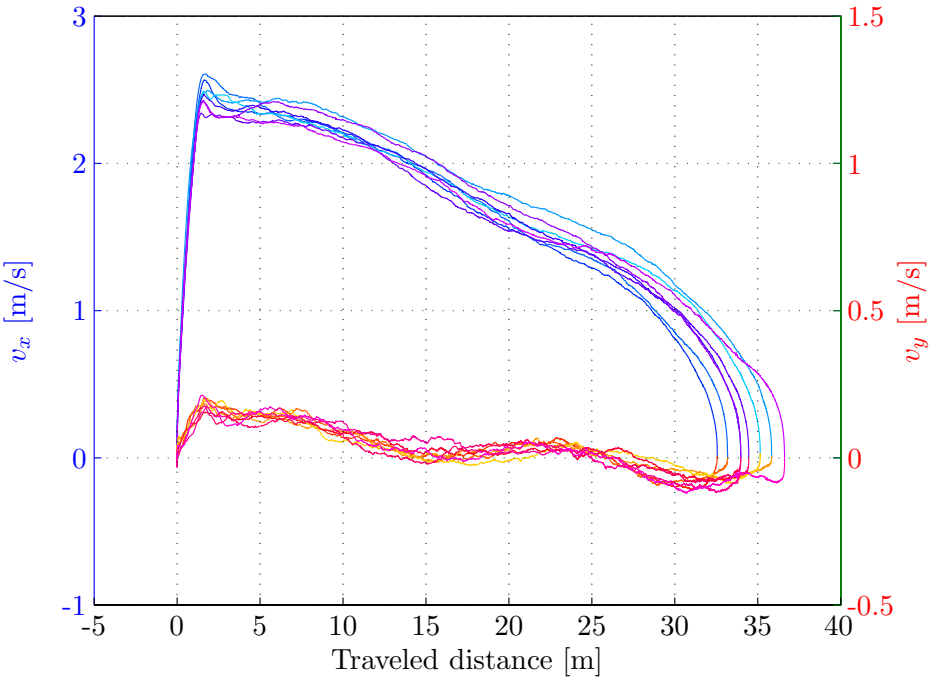

Figure 6.4: The velocity of the stone as a function of traveled distance.

within the first four iterates of λ. Secondly, the quality of the given initial value $x^0_{\lambda^0}$ determines the effort required to obtain λ^1.

The a priori stopping criterion presented in section 5.1.4 allows one to terminate the iteration once the change in the obtained result is small enough. Depending on the chosen tolerance level, this can also lead to a reduced number of iterations. The parameter δ_λ in Table 6.2 indicates how much λ can be shifted away from λ_L before $\|\Delta x\|_\infty$ reaches the level of 10^{-3}. In terms of position estimates, this corresponds to an uncertainty of 1 mm. The iterates below the horizontal line in each column denote the unnecessary iterations in terms of the stopping criterion δ_λ. In other words, the iterations which change the solution by less than the specified tolerance. The reported stopping criterions δ_λ are evaluated at λ_L, but the horizontal lines are based on the current values of δ_λ. This does not have an effect on the locations of the horizontal lines.

Yet another way to interpret the situation is that the values obtained for λ_L are all reasonably close to each other. Thus, sufficiently accurate estimates could be obtained for x_λ with a fixed value for λ for all measurements. This hypothesis was tested using the average of the logarithmic values shown in Table 6.2 ($\lambda_L = 166.10$). In that case, in terms of the position estimates, the maximum observed difference was less than 5 centimeters. Note that the respective non-linear problem can be easily and quickly resolved, since the

curvature does not need to be evaluated and a significantly looser stopping criterion δ_∇ can be employed.

The estimated velocity of the stone is illustrated in Figure 6.4 and its estimated trajectory in Figure 6.5. In order to illustrate the actual event, the trajectory plot is presented with the actual aspect ratio. For the sake of clarity, the figures only show the actual "throw phase" of the events. In the example data shown in Figure 6.3, this phase starts after the momentary pause at $t = 5.0$ seconds. Note that the two velocity components are plotted against the traveled distance.

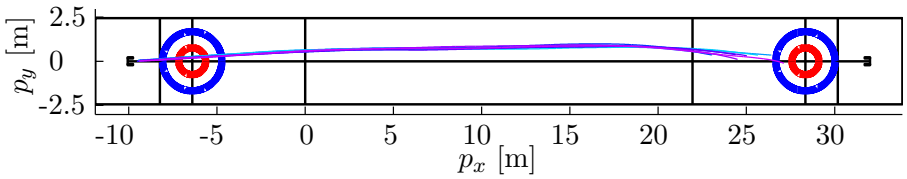

Figure 6.5: Resolved trajectories in the curling example.

The main conclusion to be drawn from Figures 6.4 and 6.5 is that the proposed solution method seems to produce consistent results, at least in this particular application.

6.2 Example 2: Ski jump

In this example, the IMU is installed in the boot of a ski-jumper. The objective is to resolve the trajectory of the boots during a single jump. This is a full 6 DOF problem, where the trajectory of the IMU is partially dictated by the profile of the hill. The profile of the hill has already been externally measured with an accuracy better than ±1 cm. The measurements were collected using two IMUs and two athletes. The last of the three measurements for each IMU was taken on a different day than the first two measurements. The state equation used in this example is the one presented in (2.26). The x-axis was chosen to be aligned with the horizontal direction of the inrun. For reference, the magnitude of the gravitational acceleration on site ($L = 66.17292°$, $H = 355$ m) is $\|g\|_2 \approx 9.822588$ m/s^2 when evaluated with (2.27).

6.2.1 Observation equation

This application exploits the following observations:

(a) the position ($\hat{p}_i$) of the jumper at a total of nine time instants t_i

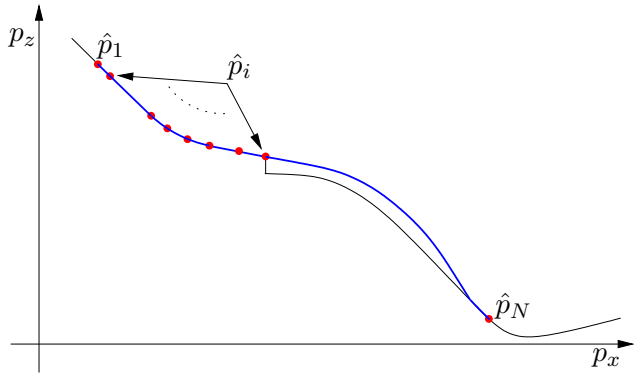

Figure 6.6: An illustration of the observations used in the ski jump example. The figure is not in scale and does not correspond to any particular measurement.

(b) the known trajectory of the boot during the inrun

(c) a number of "pseudo-observations" stating that DCM $C_i^T C_i = I$.

Observations (a) and (b) are treated as equally reliable, and the respective entries for L_o^{-1} are set to the value 10^{-2}. While we should treat the pseudo-observations as exact information, the proposed solution method only allows us to associate a finite weight with them. In this case, the weight associated with the pseudo-observations is 1, which was applied to every 50th sample.

The type (a) observations were established using the same set-up as in the previous example: a permanently magnetised strip, whose proximity is sensed with a magnetometer installed in the IMU. A total of eight magnetic strips were located in the ice track on the inrun, and one on the landing slope. The red dots in Figure 6.6 show the locations of these strips. The y-component of the position observation on the landing slope is not fixed, as the magnetic strip do not provide information about this component. As seen in Figure 6.6, the outermost magnetic strips are used to determine the initial and final points of the fixed time interval. All observations concerning attitude ($\hat{C}_i$), velocity ($\hat{v}_i$) or position ($\hat{p}_i$) of the object at time instant t_i can be taken into account with an equation of the form

$$\begin{bmatrix} w_a I & 0 & 0 & 0 & 0 \\ 0 & w_a I & 0 & 0 & 0 \\ 0 & 0 & w_a I & 0 & 0 \\ 0 & 0 & 0 & w_v I & 0 \\ 0 & 0 & 0 & 0 & w_p I \end{bmatrix} \begin{bmatrix} c_1(t_i) \\ c_2(t_i) \\ c_3(t_i) \\ v(t_i) \\ p(t_i) \end{bmatrix} = \begin{bmatrix} w_a \hat{c}_1(t_i) \\ w_a \hat{c}_2(t_i) \\ w_a \hat{c}_3(t_i) \\ w_v \hat{v}(t_i) \\ w_p \hat{p}(t_i) \end{bmatrix}, \tag{6.8}$$

where $I \in \mathbb{R}^{3\times 3}$ and the respective weights are denoted by $w_a, w_v, w_p \in \mathbb{R}_+$.

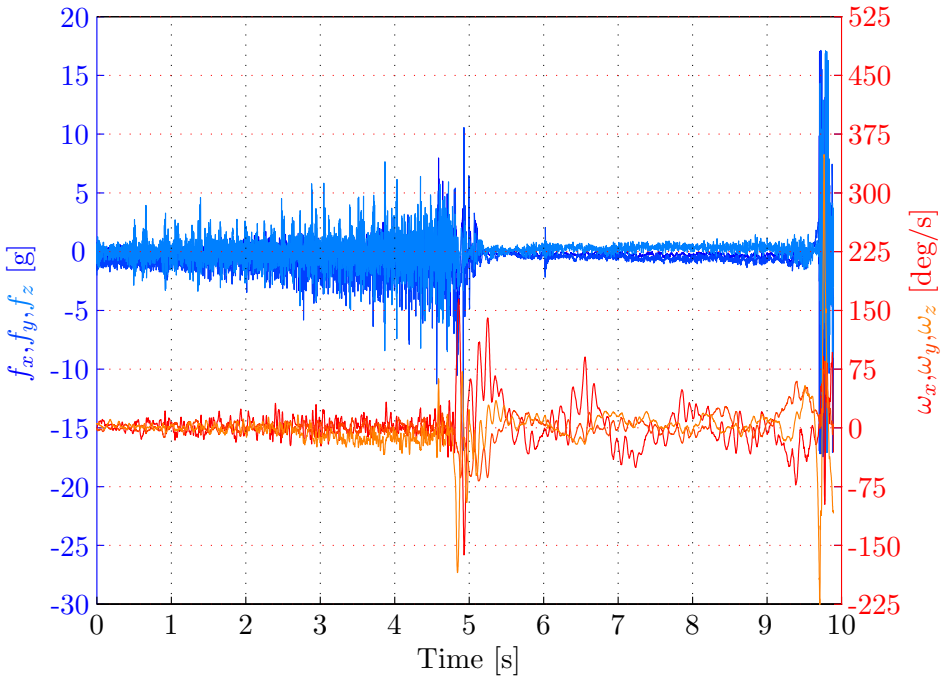

Figure 6.7: The samples of specific forces and angular velocities gathered in measurement 1.

Based on the measured profile, the inrun can be divided into three parts: two straight parts with a third, curved, part between them. For each part, the trajectory of the boot is presented as a first (straight) or third (curve) degree polynomial. Two of the eight magnetic strips were placed between the three parts of the inrun in order to be able to determine the relevant part of the inrun for each time instant (in Figure 6.6, these are the third and sixth dots from the left). During the whole inrun, the y-coordinate is taken to be a constant (d). Denoting the weight by $w_{tr} \in \mathbb{R}_+$, the trajectory information is presented in the form

$$w_{tr} \begin{bmatrix} \sum_{n=1}^{3} \hat{d}_n [p_i]_x^n - [p_i]_z \\ [p_i]_y \end{bmatrix} = w_{tr} \begin{bmatrix} -\hat{d}_0 \\ d \end{bmatrix} \tag{6.9}$$

and the non-zero part of the respective Jacobian is

$$J = w_{tr} \begin{bmatrix} \sum_{n=1}^{3} n\hat{d}_n [p_i]_x^{n-1} & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}. \tag{6.10}$$

For the straight parts of the inrun, it holds for the coefficients that $\hat{d}_2 = \hat{d}_3 = 0$. The known trajectory of the boot is used in every 25th sample.

As discussed above, this example does not contain attitude or velocity observations. While the orthogonality of the DCMs is a matter related

	Measurement					
	1	2	3	4	5	6
$\log \lambda^1$	−0.9(22)	−0.9(22)	−0.9(19)	−0.9(20)	−0.9(22)	−0.9(20)
$\log \lambda^2$	0.944(12)	0.944(10)	0.944(27)	0.944(7)	0.944(7)	0.944(8)
$\log \lambda^3$	2.111(14)	2.111(11)	2.111(34)	2.111(9)	2.111(10)	2.111(7)
$\log \lambda^4$	0.603(11)	0.468(8)	0.748(27)	0.583(6)	0.573(6)	0.574(6)
$\log \lambda^5$	1.268(12)	1.390(9)	1.401(27)	1.390(8)	1.390(5)	1.390(6)
$\log \lambda^6$	1.590(8)	1.666(7)	1.672(15)	0.904(5)	0.952(6)	1.048(5)
$\log \lambda^7$	1.227(8)	1.223(6)	1.420(15)	0.938(3)	0.982(3)	1.093(4)
$\log \lambda^8$	1.233(5)	1.207(6)	1.460(9)	0.933(3)	1.138(4)	1.090(3)
$\log \lambda^9$	1.236(4)	1.218(5)	1.458(6)	0.933(2)	0.968(4)	1.088(4)
$\log \lambda^{10}$	1.236(2)	1.217(2)	1.458(4)	0.933(2)	0.968(1)	1.088(2)
$\log \lambda^{11}$	1.236(3)	1.218(3)	1.458(3)		0.968(2)	1.088(2)
λ_L	17.197	16.514	28.689	8.561	9.298	12.241
$\kappa(\lambda_L)$	266.361	256.416	151.992	359.846	421.382	395.790
δ_λ	0.08512	0.12922	0.08979	0.07068	0.11154	0.13897
δ_C $[\times 10^{-3}]$	2.263	1.869	8.561	0.981	0.910	1.627

Table 6.3: Convergence of the line search procedure used to find λ_L in the ski jump example. The numbers in parentheses represent the number of Gauss-Newton steps required within each iterate. Iterations below the additional single-column-wide horizontal line represent the unnecessary iterations in terms of the stopping criterion δ_λ. In some measurements, the stopping criterion is fulfilled at the last iteration.

to attitude representation, it does not restrict the actual attitude in any way. An example of the specific force and angular velocity measurements which were obtained is illustrated in Figure 6.7. As can be verified from the figure, the jumper passes the end of the inrun at $t \approx 5.0$ seconds and lands at $t \approx 9.7$ seconds after passing the first magnetic strip. The event stops at $t \approx 9.9$ seconds by which time the boot has traveled almost 250 meters. No measurement-specific bias compensation was performed in this example.

6.2.2 Results

Table 6.3 shows that between 10 and 11 iterations were required for convergence. For the first IMU, the values obtained for λ_L ranged from 16.514 to 28.689 while for the second IMU they ranged from 8.561 to 12.241. It is notable that the regularization parameters related to the first two measurements of each IMU differ from each other by less than one unit. This is clearly not the case with the regularization parameter related to the third measurement. While more data is required to make any firm conclusions, the difference might well be a consequence of the different weather conditions on the two different days. As compared to the previous example, the most obvious reason for the generally smaller values of λ_L is the three-times-shorter time interval, which is known to significantly increase the accuracy

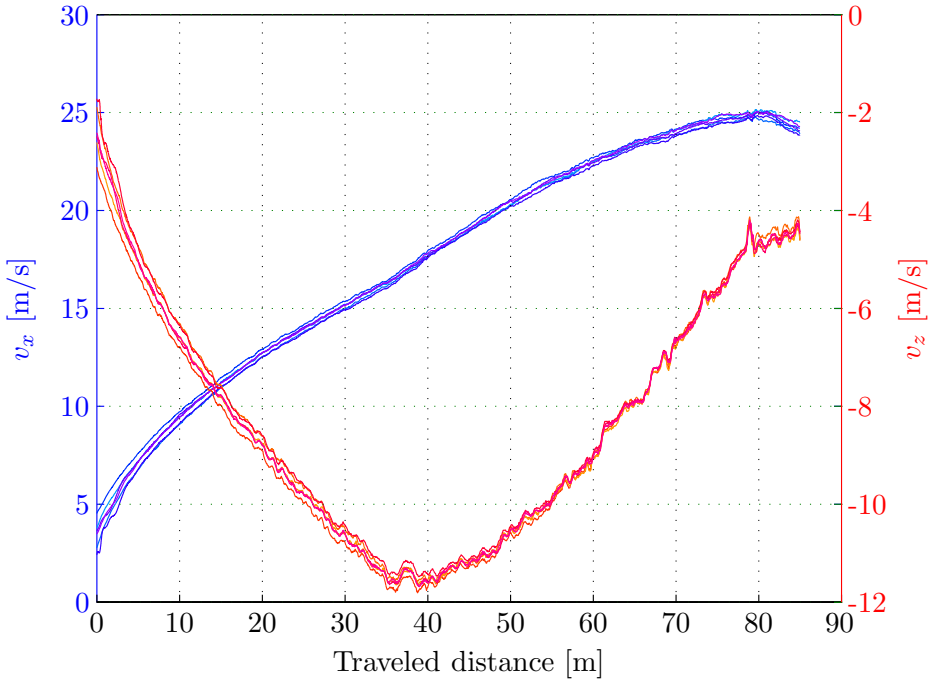

Figure 6.8: The velocity of the jumper as a function of traveled distance.

of any inertial navigation system. The overall accuracy of the observations used is on a par with the accuracy of the observations used in the previous example.

The last row of Table 6.3, δ_C, defined in (2.31), depicts the maximum orthogonality error within the navigation period. With the applied weights, the maximum observed value for δ_C is 8.561×10^{-3}. In practice, the maximum error tends to occur in the middle of two adjacent pseudo-observations. Hence, if necessary, the value obtained for δ_C could be further reduced by decreasing the number of samples between adjacent pseudo-observations. Another possible solution is to increase the weight difference of different types of observations. In both cases, any reduction of δ_C seems to lead to increased computational complexity. While a change in the weights does not alter the dimensions of the system, it will make the problem numerically less stable. As is well known, numerical stability issues often lead to decreased convergence speed, and ultimately to convergence problems [5]. Thus, in a nutshell, the reduction of the orthogonality error δ_C is a good thing in terms of the accuracy, but it comes at the cost of increased computational complexity.

In this case, the solution of the relevant non-linear problem for λ^1 took between 19 and 22 Gauss-Newton iterations. For λ^2, Gauss-Newton required 7–27 iterations and this gradually fell to 2–3 iterations for λ_L. Again, these

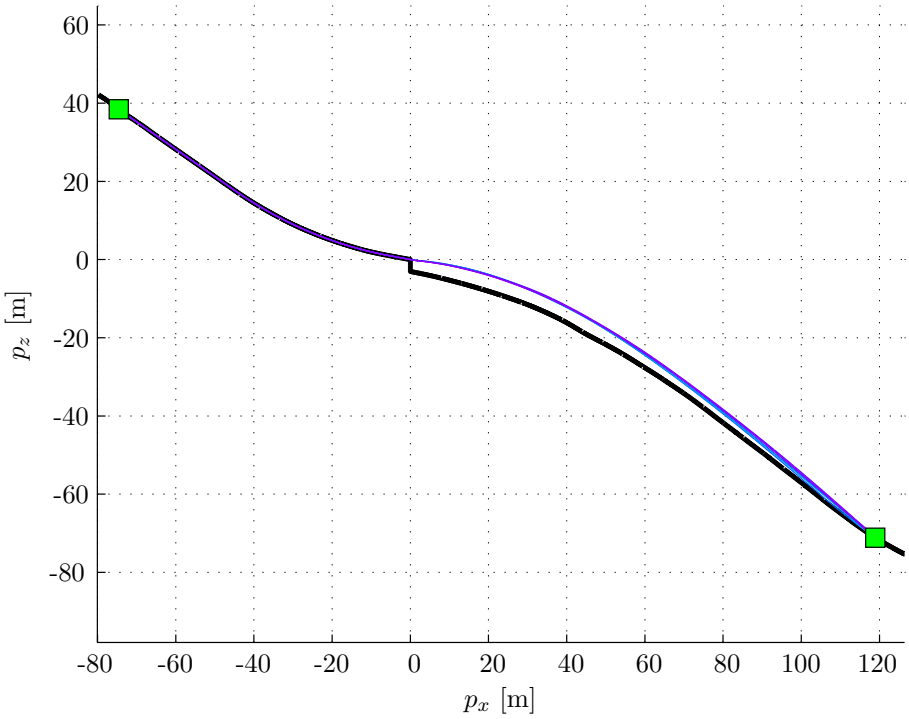

Figure 6.9: Resolved trajectories in the ski jumping example. The thick black line represents the profile of the hill while the squares represent the initial and final points of the navigation period.

results indicate that no accurate prior knowledge of λ or x_λ is necessary. To reduce the computational complexity, the same rules apply as in the previous example. The use of a predetermined value for λ is an option here, especially if such a value is determined for each IMU. In this case, the maximum position difference caused by the use of a single regularization parameter $\lambda_L = 14.131$ for both IMUs is less than 5 centimeters, as in the previous example.

The computed velocity of the boot is illustrated in Figure 6.8 and its trajectory in Figure 6.9. The velocity plot only shows the velocity of the jumper during the inrun. The differences during the first 40 meters of the inrun, particularly evident in the vertical velocity plots, are due to the different starting points of the jumpers. Notice that the clearly visible "bumps" seen on the vertical velocity plots between the traveled distances of 35 m and 80 m are not the result of sundry imperfections in the IMUs, but are due to the physical "bumps" on the ice track. This is why they overlap so clearly when plotted against the traveled distance.

Based on Figure 6.9, all six measurements seem to produce consistent results. The only visible problem happens at the landing point, where the

obviously high curvature of the trajectory is not properly visible. The main reason for this is the insufficient dynamic range of the accelerometers during those few milliseconds around the landing point.

Chapter 7

Conclusions and open questions

This thesis has constructed a new solution method for short term inertial navigation problems based on *non-Markovian fixed-interval smoothing*. In the current literature, the treatment of fixed-interval smoothing problems has focused on IVP-driven problems which satisfy the *Markov property*. This is particularly so in the field of inertial navigation, where the discussion has been exclusively about IVP-driven problems. While solution methods for BVP-driven, non-Markovian fixed-interval smoothing problems exist, they are based on recursive algorithms. This limits the way observations are treated and compels the user to pick specific boundary values.

Here, a non-recursive solution method for non-Markovian fixed-interval smoothing problems is proposed, where each observation can involve an arbitrary number of time instances. This solution method also applies to problems which satisfy the Markov property. One of the key properties of this method is its ability to treat both IVPs and BVPs without making a distinction between the two alternatives. Moreover, the user does not need to specify the *boundary values*. Boundary values are, of course, required, but they are treated no differently than the rest of the observations. Thus, the solution is independent of any specific boundary values, which are often as erroneous as the rest of the observations. This is particularly important in inertial navigation, where such things as exact boundary values do not exist. The solution method proposed here is not only limited to (short-term) inertial navigation, but can be applied to many types of general fixed-interval smoothing problems.

The thesis also presented new methods of calibration. Firstly, a laboratory calibration method was presented, which allows the user to allocate the calibration to a specific dynamic range. This has the potential to improve the

accuracy of the sensors used for a specific application. Secondly, the thesis presented a new method for online calibration which can be used to perform the basic calibration of an IMU in dynamic situations, where it is typically not possible to use specific equipment to aid the calibration. Both the calibration methods proposed here yield the same form of equations, allowing the user to employ the very same algorithm to obtain the solution.

The problem presented here is recognized to be a *Tikhonov regularization* problem, and these frequently arise in the field of *inverse problems*. This not only allows the problem to be solved efficiently, but provides powerful tools for automatically recognizing the differences in the reliability of the state and observation equations. The reliability difference is governed by the *regularization parameter* λ. A regularization parameter is particularly useful in those practical situations where errors are always present, but are often not reliably characterized.

Given a linear fixed-interval smoothing problem where the value of the regularization parameter λ is known, the exact conditions for the uniqueness of the respective solution X_λ are known. When the problem fulfills certain assumptions, the solution is known to be the best linear unbiased estimator (BLUE). As a rule of thumb, X_λ can be very efficiently and robustly determined as long as the matrices involved are sparse. Depending on the chosen numerical solution method, some particularly "nasty" non-linear fixed-interval smoothing problems may prove to be an exception to this rule, especially if the initial guess is a long way from the desired solution. This could cause the chosen numerical solution method to stagnate or descend into some local minimum representing an undesired solution. However, it often turns out that one can make a reasonably good initial guess by first solving a linear substitute problem. The statistical properties of the solution obtained for the non-linear problem are unknown.

If the regularization parameter is to be determined along with the respective solution, it must be further assumed that the relevant *L-curve* actually possesses the expected shape. If it does, linear problems represented by sparse matrices can be both efficiently and robustly solved. A Matlab® code for this purpose is provided in Appendix A. In principle, the above statements also hold for non-linear problems. However, when there is non-linearity involved, there is always the increased possibility of something going wrong. On the other hand, if one has a specific application in mind, it is often possible to say something more definite about that particular application. At the very least, simulations can then be used to learn more about the properties of the situation.

Finally, the solution method was applied to two novel sports applications. Real-life measurements were gathered to demonstrate the use of the proposed method in practical situations.

In developing this method for solving certain fixed-interval smoothing problems, we have encountered a number of open questions. While most of these are closely related to the particular solution method, a few open questions of more general significance have emerged. This thesis concludes by presenting the key questions left open in this context.

Open question 1

The question about the well-posedness of the problem of finding the value of λ as the point with maximum curvature $\kappa(\lambda)$ depends on the shape of the L-curve. This is characterized by the (discrete) Picard condition, and how to find out the exact conditions when the maximization of $\kappa(\lambda)$ is a well-posed question in the case of generally non-linear fixed-interval smoothing problem remains an open question.

Open question 2

This thesis presented a number of ways of dealing with the orthogonality issue related to the attitude representation. In practice, the methods presented here can be considered valid, as they are usually able to produce results in which the relevant orthogonality error is "reasonably small". However, these methods will fail at some point when the tolerances are tightened. Ultimately, the only right way is to seek the solution from a specific subset of $\mathbb{R}^{kN\times q}$. It remains an open question as to how to solve the generally non-linear problem

$$x_\lambda = \text{argmin}_x \ \|f_s(x) - z_s\|_2^2 + \lambda \|f_o(x) - z_o\|_2^2 \ \text{ where } \ C_i C_i^T = I \ \ \forall \ \ i \in [1, N]\,.$$

Open question 3

With regard to the applicability of this solution method to practical applications, the next step is to analyze the sensitivity of the obtained solution to perturbations in the input data. In addition to any analysis of perturbation caused by the available measurements and observations, the solution's sensitivity to the type and quality of the available observations needs to be studied. Such an analysis should also investigate the performance of the proposed estimation method against the known alternatives in compatible applications.

Open question 4

Statistical inversion has often replaced Tikhonov regularization as the method of choice in inverse problems [101, 102]. Without a doubt, given the properties of the errors, statistical inversion techniques are capable of producing at least equally accurate estimates as the method presented here. This is especially so if the problem is non-linear. It remains as an open question as to whether the premises for statistical inversion are acceptable in fixed-interval smoothing; and in what sense the results are better.

Open question 5

One possible extension of this solution method is to replace the deterministic dynamics model with a *stochastic differential equation* (SDE) [103]. Then, the more accurate dynamics model obtained can reasonably be expected to yield better results, at least when there are enough observations to accurately reproduce the modeled stochastic processes. Thus, the actual open question is whether one can obtain more accurate results by modeling the imperfections of the sensors with SDEs (cf. item (b) on page 10).

Bibliography

[1] A. Lawrence, *Modern Inertial Technology: Navigation, Guidance and Control*. New York, NY: Springer, second ed., 1998.

[2] D. H. Titterton and J. L. Weston, *Strapdown Inertial Navigation Technology*. Stevenage, United Kingdom and Reston, VA: IEE and AIAA, second ed., 2004.

[3] M. S. Grewal, L. R. Weill, and A. P. Andrews, *Global Positioning Systems, Inertial Navigation, and Integration*. New York, NY: John Wiley & Sons, 2001.

[4] A. B. Chatfield, *Fundamentals of High Accuracy Inertial Navigation*. Reston, VA: AIAA, third ed., 1997.

[5] W. H. Press, B. P. Flannery, *et al.*, *Numerical Recipes*. New York, NY: Cambridge University Press, third ed., 2007.

[6] U. M. Ascher and L. R. Petzold, *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. Philadelphia, PA: SIAM, 1998.

[7] B. C. Levy, R. Frezza, and A. J. Krener, "Modeling and estimation of discrete-time gaussian reciprocal processes," *IEEE Trans. Autom. Control*, vol. 35, pp. 1013–1023, September 1990.

[8] Wikipedia, "Ski jumping." `http://en.wikipedia.org/wiki/Ski_jumping`, December 2012.

[9] Wikipedia, "Javelin throw." `http://en.wikipedia.org/wiki/Javelin_throw`, December 2012.

[10] Wikipedia, "Curling." `http://en.wikipedia.org/wiki/Curling`, December 2012.

[11] A. Wägli, *Trajectory Determination and Analysis in Sports by Satellite and Inertial Navigation*. PhD thesis, École polytechnique fédérale de Lausanne, Switzerland, 2009.

[12] J. Alves, J. Lobo, and J. Dias, "Camera-inertial sensor modeling and alignment for visual navigation," *J. Robotic Syst.*, vol. 21, no. 1, pp. 6–12, 2004.

[13] Z. F. Syed, P. Aggarwal, *et al.*, "A new multi-position calibration method for mems inertial navigation systems," *Meas. Sci. Technol.*, vol. 18, no. 7, pp. 1897–1907, 2007.

[14] J. C. Lötters, J. Schipper, *et al.*, "Procedure for in-use calibration of triaxial accelerometers in medical applications," *Sensor Actuat. A-Phys.*, vol. 68, no. 1-3, pp. 221–228, 1998.

[15] S. Y. Cho and C. G. Park, "A calibration technique for a redundant imu containing low-grade inertial sensors," *ETRI J.*, vol. 27, no. 4, pp. 418–426, 2005.

[16] M. Hwangbo and T. Kanade, "Factorization-based calibration method for mems inertial measurement unit," in *IEEE Int. Conf. Robot. Autom. (Pasadena, CA)*, pp. 1306–1311, May 2008.

[17] I. Skog and P. Händel, "Calibration of a mems inertial measurement unit," in 17$^{\text{th}}$ *IMEKO World Congress (Rio de Janeiro, Brazil)*, September 2006.

[18] G. Aslan and A. Saranli, "Characterization and calibration of mems inertial measurement units," in 16$^{\text{th}}$ *Eur. Signal Process. Conf (Lausanne, Switzerland)*, August 2008.

[19] Z. C. Wu, Z. F. Wang, and Y. Ge, "Gravity based online calibration for monolithic triaxial accelerometers' gain and offset drift," in *Proc. of* 4$^{\text{th}}$ *World Congr. on Intell. Cont. and Autom. (Shanghai, China)*, vol. 2, pp. 2171–2175, June 2002.

[20] Y. Li, X. Niu, *et al.*, "An in-situ hand calibration method using pseudo observation scheme for low-end inertial measurement units," *Meas. Sci. Technol.*, vol. 23, 2012.

[21] W. T. Fong, S. K. Ong, and A. Y. C. Nee, "Methods for in-field user calibration of an inertial measurement unit without external equipment," *Meas. Sci. Technol.*, vol. 19, no. 8, 2008.

[22] A. Gelb, *Applied Optimal Estimation.* Cambridge, MA: The M.I.T. Press, 1974.

[23] A. Bryson and M. Frazier, "Smoothing for linear and non-linear dynamic systems," *Proc. Opt. Sys. Synthesis Conf.*, vol. 19, pp. 353–364, February 1963.

[24] G. J. Bierman, "A new computationally efficient fixed-interval, discrete-time smoother," *Automatica*, vol. 19, pp. 503–511, September 1983.

[25] R. G. Gibbs, "Square root modified bryson-frazier smoother," *IEEE Trans. Autom. Control*, vol. 56, pp. 452–456, February 2011.

[26] H. E. Rauch, F. Tung, and C. T. Striebel, "Maximum likelihood estimates of linear dynamic systems," *AIAA J.*, vol. 3, no. 8, pp. 1445–1450, 1965.

[27] D. C. Fraser and J. E. Potter, "The optimum linear smoother as a combination of two optimum linear filters," *IEEE Trans. Autom. Control*, vol. 14, pp. 387–390, August 1969.

[28] B. Boujemaa Ait-El-Fquih and F. Desbouvries, "On bayesian fixed-interval smoothing algorithms," *IEEE Trans. Autom. Control*, vol. 53, pp. 2437–2442, November 2008.

[29] K. Watanabe, "Forward-pass bryson-frazier smoother in discret-time systems," *Int. J. Syst. Sci.*, vol. 23, no. 1, pp. 71–84, 1992.

[30] C. K. Chui and G. Chen, *Kalman Filtering with Real-Time Applications*. New York, NY: Springer, fourth ed., 2009.

[31] S. Sarkka, "Unscented rauch–tung–striebel smoother," *IEEE Trans. Autom. Control*, vol. 53, pp. 845–849, April 2008.

[32] E.-H. Shin, *Estimation techniques for low-cost inertial navigation*. PhD thesis, The University of Calgary, 2005.

[33] J. L. Crassidis and J. L. Junkins, *Optimal Estimation of Dynamic Systems*. Boca Raton, FL: Chapman & Hall, 2004.

[34] H. Liu, S. Nassar, and N. El-Sheimy, "Two-filter smoothing for accurate ins/gps land-vehicle navigation in urban centers," *IEEE Trans. Veh. Technol.*, vol. 59, pp. 4256–4267, November 2010.

[35] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. New York, NY: John Wiley & Sons, first ed., 2001.

[36] K. Gade, "Navlab, a generic simulation and post-processing tool for navigation," *European J. Nav.*, vol. 2, November 2004.

[37] Y. Yang, Z. Jin, *et al.*, "Application of fixed interval smoothing to gps/dr integrated navigation system," in *Proc. of Intell. Transport. Syst. (Shanghai, China)*, vol. 2, pp. 1027–1031, October 2003.

[38] A. B. Willumsen and Ø. Hegrenæs, "The joys of smoothing," in *OCEANS 2009-EUROPE (Bremen, Germany)*, pp. 1–7, May 2009.

[39] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Norwood, MA: Artech House, 2008.

[40] J. Crassidis, F. Markley, and Y. Cheng, "Survey of nonlinear attitude estimation methods," *J. Guid. Control Dynam.*, vol. 30, no. 1, pp. 12–28, 2007.

[41] K. Watanabe, "A new forward-pass fixed-interval smoother using the u-d information matrix factorization," *Automatica*, vol. 22, no. 4, pp. 465–475, 2003.

[42] S. G. Tzafestas, "On optimum distributed-parameter filtering and fixed-interval smoothing for colored noise," *IEEE Trans. Autom. Control*, vol. 17, pp. 448–458, August 1972.

[43] M. L. Psiaki and M. Wada, "Derivation and simulation testing of a sigma-points smoother," *J. Guid. Control Dynam.*, vol. 30, January-February 2007.

[44] B. P. Gibbs, *Advanced Kalman Filtering, Least-Squares and Modeling*. Hoboken, NJ: John Wiley & Sons, 2011.

[45] H. L. Weinert, "Smoothing for multipoint boundary value models," *Syst. Control Lett.*, vol. 17, no. 6, pp. 445–452, 1991.

[46] H. L. Weinert, *Fixed Interval Smoothing for State-Space Models*. Norwell, MA: Kluwer, first ed., 2001.

[47] MathWorks, "Matlab (r2011a), 32-bit (win32)." `http://www.mathworks.com/`, March 2011.

[48] A. N. Tikhonov, "On the stability of inverse problems," *Dokl. Akad. Nauk SSSR+*, vol. 39, no. 5, pp. 195–198, 1943.

[49] A. N. Tikhonov, "Solution of incorrectly formulated problems and the regularization method," *Dokl. Akad. Nauk SSSR+*, vol. 151, pp. 501–504, 1963.

[50] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-posed Problems*. Washington, WA: VH Winston, 1977.

[51] D. L. Phillips, "A technique for the numerical solution of certain integral equations of the first kind," *J. ACM*, vol. 9, no. 1, pp. 84–97, 1962.

[52] A. E. Hoerl, "Application of ridge analysis to regression problems," *Chem. Eng. Prog.*, vol. 58, pp. 54–59, 1962.

[53] D. W. Marquardt, "Generalized inverses, ridge regression, biased linear estimation, and nonlinear estimation," *Technometrics*, vol. 12, no. 3, pp. 591–612, 1970.

[54] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*. Philadelphia, PA: SIAM, 1998.

[55] R. S. Anderssen and P. M. Prenter, "A formal comparison of methods proposed for the numerical solution of first kind integral equations," *J. Aust. Math. Soc. B*, vol. 22, pp. 488–500, 1981.

[56] M. A., D. L., and D. S. V., "The wiener filter and regularization methods for image restoration problems," in *Proc. of Int. Conf. on Image Anal. and Processing (Venice, Italy)*, pp. 394–399, September 1999.

[57] R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering*. Hoboken, NJ: John Wiley & Sons, third ed., 1997.

[58] L. F. Shampine, I. Gladwell, and S. Thompson, *Solving ODEs with Matlab*. Cambridge, UK: Cambridge University Press, 2003.

[59] C. W. Misner, K. S. Thorne, and J. A. Wheeler, *Gravitation*. New York, NY: W. H. Freeman & Company, 1973.

[60] T. W. B. Kibble and F. H. Berkshire, *Classical Mechanics*. Harlow, UK: Addison Wesley Longman, fourth ed., 1996.

[61] R. G. Takwale and P. S. Puranik, *Introduction to Classical Mechanics*. New Delhi, India: Tata McGraw-Hill, 1979.

[62] J. M. Finn, *Classical Mechanics*. Hingham, MA: Infinity Science Press, 2008.

[63] P. H. Zipfel, *Modeling and Simulation of Aerospace Vehicle Dynamics*. Reston, VA: AIAA, second ed., 2007.

[64] M. Mansfield and C. O'Sullivan, *Understanding Physics*. New York, NY: John Wiley & Sons, 1998.

[65] G. Wahba, "A least squares estimate of satellite attitude," *SIAM Rev.*, vol. 7, no. 1, p. 409, 1965.

[66] G. H. Golub and C. F. V. Loan, *Matrix Computations*. Baltimore, MD: The Johns Hopkins University Press, third ed., 1996.

[67] I. Y. Bar-Itzhack and K. A. Fegley, "Orthogonalization techniques of a direction cosine matrix," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-5, pp. 798–804, September 1969.

[68] IEEE(TAES) and AIAA, "IEEE recommended practice for precision centrifuge testing of linear accelerometers," tech. rep., IEEE, 2004.

[69] M. El-Diasty and S. Pagiatakis, "A rigorous temperature-dependent stochastic modelling and testing for mems-based inertial sensor errors," *Sensors*, vol. 9, pp. 8473–8489, October 2009.

[70] J. Crank and P. Nicolson, "A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type," *Adv. Comput. Math.*, vol. 6, pp. 207–226, 1996.

[71] J. W. Demmel, *Applied Numerical Linear Algebra*. Philadelphia, PA: SIAM, 1997.

[72] C. P. Niculescu and L.-E. Persson, *A Contemporary Approach: Convex Functions and Their Applications*. New York, NY: Springer, 2006.

[73] A. Bossavit, *Computational Electromagnetism*. New York, NY: Academic Press, 1998.

[74] S. C. Brenner and R. Scott, *The Mathematical Theory of Finite Element Methods*. New York, NY: Springer, third ed., 2008.

[75] I. M. Gelfand and S. V. Fomin, *Calculus of Variations*. Mineola, NY: Dover Publications, 1991.

[76] S. Larsson and V. Thomée, *Partial Differential Equations with Numerical Methods*. Berlin, Germany: Springer, 2003.

[77] M. S. Gockenbach, *Partial Differential Equations: Analytical and Numerical Methods*. Philadelphia, PA: SIAM, 2002.

[78] D. Kincaid and W. Cheney, *Numerical Analysis*. Florence, KY: Brooks/Cole, third ed., 2002.

[79] R. Garg, *Analytical and Computational Methods in Electromagnetics*. Norwood, MA: Artech House, 2008.

[80] A. ke Björck, *Numerical Methods for Least Squares Problems*. Philadelphia, PA: SIAM, 1996.

[81] A. Aitken, "On least squares and linear combination of observations," *Proc. R. Soc. Edinb.*, vol. 55, pp. 42–48, 1934.

[82] R. A. Johnson and D. W. Wichern, *Applied Multivariate Statistical Analysis*. Upper Saddle River, NJ: Pearson Prentice Hall, 2007.

[83] G. H. Golub, P. C. Hansen, and D. P. O'Leary, "Tikhonov regularization and total least squares," *SIAM J. Matrix Anal. A.*, vol. 21, no. 1, pp. 185–194, 1999.

[84] P. de Groen, "An introduction to total least squares," *Nieuw Arch. Wisk.*, vol. 4, pp. 237–253, 1996.

[85] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization-Analysis, Algorithms, and Engineering Applications*. Philadelphia, PA: SIAM, first ed., 2001.

[86] M. Hanke and T. Raus, "A general heuristic for choosing the regularization parameter in ill-posed problem," *SIAM J. Sci. Comput.*, vol. 17, pp. 956–972, July 1996.

[87] G. H. Golub, M. Heath, and G. Wahba, "Generalized cross-validation as a method for choosing a good ridge parameter," *Technometrics*, vol. 21, no. 2, pp. 215–223, 1979.

[88] H. W. Engl and W. Grewer, "Using the l-curve for determining optimal regularization parameters," *Numer. Math.*, vol. 69, pp. 25–31, November 1994.

[89] P. C. Hansen, "Analysis of discrete ill-posed problems by means of the l-curve," *Siam Rev.*, vol. 34, pp. 561–580, December 1992.

[90] K. B. Petersen and M. S. Pedersen, "The matrix cookbook," tech. rep., http://matrixcookbook.com, November 2008.

[91] P. C. Hansen, "The discrete picard condition for discrete ill-posed problems," *BIT Num. Math.*, vol. 30, pp. 658–672, 1990.

[92] T. Viklands, *Algorithms for the weighted orthogonal procrustes problem and other least squares problems*. PhD thesis, Umeå University, 2006.

[93] A. W. Bojanczyk and A. Lutoborski, "The procrustes problem for orthogonal stiefel matrices," *SIAM J. Sci. Comput.*, vol. 21, December - March 2000.

[94] C. T. Kelley, *Iterative Methods for Optimization*. Philadelphia, PA: SIAM, 1999.

[95] D. P. Bertsekas, *Nonlinear Programming*. Belmont, MA: Athena Scientific, first ed., 1995.

[96] R. Fletcher, *Practical methods of optimization*. New York, NY: John Wiley & Sons, 1987.

[97] *Analog Devices ADXRS300ABG Datasheet*, 2004. Revision B, Checked on 22.12.2009.

[98] *VTI SCA610-CC5H1A Datasheet*, 2005. Revision 3, Checked on 23.09.2010.

[99] *VTI SCA620-CHCV1A Datasheet*, 2006. Revision 2/2, Checked on 22.12.2009.

[100] M. Virmavirta, T. Nieminen, *et al.*, "High precision inertial measurement for tracking the trajectories of ski jumpers - "smart boot project"," in *13th ECSS Congress (Estoril, Portugal)*, July 2008.

[101] J. Kaipio and E. Somersalo, *Statistical and Computational Inverse Problems*, vol. 160 of *Applied Mathematical Sciences*. New York, NY: Springer, 2005.

[102] T. Bui-Thanh, "A gentle tutorial on statistical inversion using the bayesian paradigm," tech. rep., ICES Reports, May 2012.

[103] F. C. Klebaner, *Introduction to Stochastic Calculus with Applications*. London, UK: Imperial College Press, second ed., 2005.

Appendix A

Reference implementation

```
function [lambda,X] = optimizeLCurve(Js,Zs,Jo,Zo,x_min,x_max)
    lambda = 10^(fminbnd(@curvature,x_min,x_max));
    function [kappa] = curvature(x)
            R = qr([ Js ; 10^(x/2) * Jo ]);
            X = R\(R'\(Js'*Zs+10^x*Jo'*Zo));
           Ws = Js*X-Zs;
           Wo = Jo*X-Zo;

           dX = -R\(R'\(Jo'*Wo));
         JsdX = Js*dX;
         JodX = Jo*dX;

          ddX = -2*R\(R'\(Jo'*JodX));
        JsddX = Js*ddX;
        JoddX = Jo*ddX;

        Rho   =  trace(Ws'*Ws)^(1/2);
        Eta   =  trace(Wo'*Wo)^(1/2);
        dRho  =  trace(Ws'*JsdX)/Rho;
        dEta  =  trace(Wo'*JodX)/Eta;
        ddRho = (trace(JsdX'*JsdX+Ws'*JsddX)-dRho^2)/Rho;
        ddEta = (trace(JodX'*JodX+Wo'*JoddX)-dEta^2)/Eta;

        kappa = -(dRho*ddEta-ddRho*dEta)/(dRho^2+dEta^2)^(3/2);
    end
end
```

A Matlab® algorithm for solving linear fixed-interval smoothing problems with unknown λ.
Input: sparse matrices J_s, Z_s, J_o, Z_o and initial bounds $\lambda_L \in [\log(\lambda^0_{min}), \log(\lambda^0_{max})]$.
Output: λ_L and X_λ.

Publication 1

Tuukka Nieminen, Jari Kangas, Saku Suuriniemi and Lauri Kettunen
"An enhanced multi-position calibration method for consumer-grade inertial measurement units applied and tested"
Institute of Physics: Measurement Science and Technology **21** (11pp), 2010
doi:10.1088/0957-0233/21/10/105204

Meas. Sci. Technol. **21** (2010) 105204 (11pp) doi:10.1088/0957-0233/21/10/105204

An enhanced multi-position calibration method for consumer-grade inertial measurement units applied and tested

Tuukka Nieminen, Jari Kangas, Saku Suuriniemi and Lauri Kettunen

Tampere University of Technology, Electromagnetics, PO Box 692, FI-33101 Tampere, Finland

E-mail: tuukka.nieminen@tut.fi

Received 26 March 2010, in final form 22 June 2010
Published 13 August 2010
Online at stacks.iop.org/MST/21/105204

Abstract
An accurate inertial measurement unit (IMU) is a necessity when considering an inertial navigation system capable of giving reliable position and velocity estimates even for a short period of time. However, even a set of ideal gyroscopes and accelerometers does not imply an ideal IMU if its exact mechanical characteristics (i.e. alignment and position information of each sensor) are not known. In this paper, the standard multi-position calibration method for consumer-grade IMUs using a rate table is enhanced to exploit also the centripetal accelerations caused by the rotation of the table. Thus, the total number of measurements rises, making the method less sensitive to errors and allowing use of more accurate error models. As a result, the accuracy is significantly enhanced, while the required numerical methods are simple and efficient. The proposed method is tested with several IMUs and compared to existing calibration methods.

Keywords: multi-position calibration, inertial measurement unit, rate table, centripetal acceleration

1. Introduction

An inertial measurement unit (IMU) is the part of an inertial navigation system that provides the measurement data. A typical IMU consists of accelerometers and gyroscopes measuring accelerations and angular velocities which have to be numerically integrated to get the position estimates. Consequently, inertial measurement systems are very sensitive to measurement errors, which we can significantly decrease by means of calibration.

We classify the measurement errors of the IMU into two different categories.

1. The errors caused by the unknown mechanical characteristics of the IMU.
   - An uncalibrated IMU is assumed to provide us with specific force and angular rate measurements in an orthogonal basis. In practice, an IMU never yields measurements in an orthogonal basis as such. This is due to inevitable imperfections in the manufacturing process of the IMU.
   - The accelerometers are assumed to measure the specific force of a single point of the IMU. Because of the physical size of the accelerometers, they will measure the specific forces of different points. This is sometimes called the 'size effect' [1, 2].

   To compensate for these effects, one needs to find the measurement axes of the sensors and the locations of the accelerometers with respect to a chosen point. These errors will be present regardless of the quality of the employed sensors.
2. The errors caused by individual sensors within the IMU.
   - There are both stochastic and deterministic errors present in the data given by a single sensor. The purpose of calibration is to compensate for the most significant deterministic error sources. The most commonly encountered error sources include bias, scale factor and cross-correlation errors. In addition, consumer-grade gyroscopes are typically sensitive to linear accelerations [1].

1.1. Background

Multi-position calibration is a well-known and a widely employed calibration method. It is based on the idea of keeping the IMU in different positions with respect to the local gravitational acceleration and the angular rate of the Earth [1, 3, 4]. With several independent measurements, it is possible to find the most important error terms of the IMU. Due to the relatively large gravitational acceleration ($\approx$9.80 m s^{-2}), this method can be applied to practically any set of three accelerometers. However, the extremely small angular rate of the Earth ($\approx 4.17 \times 10^{-3}$ deg s^{-1}) limits the use of this method to only the very accurate and expensive gyroscopes. With consumer-grade gyroscopes, the required reference signals must be obtained by another means.

Considering the calibration of accelerometers, the majority of the proposed methods are based on the basic idea of the multi-position calibration method. In every position, the magnitude of the specific force should be equal to the local gravitational acceleration [3, 5–12]. While the exploited calibration methods and sensor error models have been essentially the same, much progress is achieved in making the required calibration equipment less expensive and thus more suitable for calibration of consumer-grade accelerometers [5, 7, 9, 10]. The calibration of redundant sensor configurations is considered in [8, 9].

The basic problem of the calibration methods relying on the gravitational acceleration is the limited magnitude of the reference signals. In practice, also the number of linearly independent measurements is limited in order to keep the method simple. To overcome these limitations, some dynamic calibration methods have also been suggested. These include the use of a three-dimensional vibration generator [19] and the centripetal accelerations caused by a pendulum [20]. The method considered in [19] also gives a possibility of considering frequency-dependent errors, but requires costly equipment. In the method considered in [20], the rotation radii must be provided and the reference angular rates are computed by numerically differentiating the angle data of the pendulum given by an encoder.

In addition to gyroscope-based inertial measurement units, it is possible to design a gyroscope-free IMU (or GF-IMU). It is based solely on a number of accelerometers mounted on a rigid body. Recent discussions of GF-IMU design can be found from [13–16]. A typical calibration procedure for a GF-IMU is based on the multi-position calibration [14, 16–18] and possibly to a dynamical test making it possible to also determine the locations of the accelerometers [18]. The main motivation for developing such an instruments has been the lack of affordable and reasonably accurate gyroscopes [14, 15]. The disadvantages include the inherently larger size of any reasonably accurate GF-IMU as compared to gyroscope-based IMUs and the degraded measurement accuracy due to indirect measurement of the angular velocity. Furthermore, at the present moment, there are several gyroscopes of a fair price–quality ratio available from many manufacturers. In addition to typical 'yaw-rate gyros', there is a growing number of devices also measuring the other two directions when mounted on the same circuit board.

Some methods to calibrate consumer-grade gyroscopes are considered in [5, 9, 20]. In [9], the required rotations are performed on a rate table operated by hand. Because of this, the true angular rate is unknown and additional rotations with a known rotation angle are required to determine the scale factors. In [5], the calibration is based on multi-position calibration performed on a rate table with known angular rates. Similarly in [20], the numerically computed angular rates of the pendulum can be directly exploited in the calibration. In these studies, the exploited sensor error model has been essentially the same, where g-sensitivity of the gyroscopes is not taken into account.

A totally different kind of an approach is to calibrate the IMU while on the move, using additional information acquired, for example, from a GPS system. These methods are well exploited, for example, in [1, 2, 4, 21, 22] and are not considered here.

1.2. Goal of this study

We consider the calibration of IMUs performed on a low-cost rate table. Methodologically, the most relevant sources of this work are [5, 20]. Both consider the calibration of consumer-grade IMUs containing three accelerometers and gyroscopes. We will combine the basic ideas of these studies and make some important generalizations and additions to the methods considered therein.

We propose a method that is based on multi-position calibration generalized to exploit the large range of centripetal accelerations caused by different rotation rates of the rate table. As a result,

- the number of measurement data increases and the calibration procedure becomes less sensitive to errors;
- we can exploit more accurate error models than the ones used in the referred studies and
- it is possible to calibrate the IMU for a certain dynamical range.

The basic idea of the proposed calibration method is to compensate simultaneously for a number of different error sources instead of seeking every modeled error term separately. That is, we will not give a specific meaning to each parameter within the error model, but rather concentrate on compensating for them. The gain of this approach will realize below as somewhat simpler mathematical treatment and more importantly, efficient calibration method based on well-known and widely exploited mathematical tools. If considered necessary, it is just a question of interpretation to distinguish specific error terms from the computed results (as will be discussed later).

Formally said, the proposed calibration method is based on affine [23] inputs. The reference accelerations contain a constant gravitational acceleration term and a centripetal acceleration term, which is a linear function of the rotation radii. Correspondingly, we model the errors of the accelerometer and gyroscope triads as affine functions. That is, the calibrated output of the sensor triad is modeled by a

constant term added to a linear function of the raw sensor output.

Although one is in many cases well aware of the exact type of the sensors within the IMU, the proposed method is also well suited for situations where this is not true. As the calibration methods does not require any specific prior knowledge of the sensors or their locations and directions within the IMU, it can be considered as a black box.

In order for any kind of a calibration method to be useful in practice, it should contain some kind of checks with which one can reliably make sure that the calibration was successful. Along with some standard tools for controlling the quality of the calibration, we introduce a new kind of a control quantity. This is the rotation radius needed to exploit centripetal accelerations. As it could be a difficult task to measure this externally with adequate accuracy, we leave it for the calibration routine to determine. This value is then used as a control quantity, whose approximate value we can easily measure *independently from the calibration routine*.

We will focus especially on the calibration of consumer-grade inertial sensors. The practical issues of the actual calibration procedure, such as the accuracy of the reference data, are emphasized. Throughout the paper, we assume that the IMU is stable over time. This is to say, the effect of temporal instability is assumed to be negligible as compared to the other error sources. While this might sound like a major disadvantage for calibration method consumer-grade sensors with known problems with temporal stability, it actually is not. The possible temporal instability can be compensated by occasionally comparing the output of the sensors when they are known to be in the same position.

1.3. Some remarks

Before going on, let us introduce a few central terms. In this context, *an error model* is a bijective relation between the raw and the calibrated output of the sensors. Construction of this relation includes both *calibration* and *compensation*. Calibration means that a certain change in the raw output of a certain sensor is related to a certain change in the input [1]. Compensation stands for correcting a number of deterministic errors present in the measurements [1].

This text is organized as follows. In section 2, the error models for an accelerometer and a gyroscope triad are presented. The calibration setup will be discussed in section 3. The calibration procedure and the underlying formalism are presented in section 4. Test results are discussed in section 5 before the conclusions in section 6. Some technical details considering the accuracy of the rate table are discussed in appendices A and B.

2. Error models

In this section, we will derive the error models mapping the raw output of the sensors to the calibrated output of the IMU. The aim is to find such a mapping for a certain IMU that minimizes the difference between the actual input and the calibrated output of the IMU (the exact meaning of this is explained later). The raw output of the IMU is assumed to be a set of AD-converted voltages, naturally expressed as bits. Thus, there is no need to convert the measurements into some more meaningful units prior to applying the constructed error model. Observe that the error model also takes the deterministic errors caused by the interface electronics of the IMU into account. Hence, the calibration should be performed with the same interface electronics as used in the actual measurements.

The calibrated accelerometer and gyroscope outputs given by the IMU are expressed as real-valued vectors $\boldsymbol{a} \in \mathbb{R}^3$ and $\boldsymbol{\omega} \in \mathbb{R}^3$, respectively. Both vectors $\boldsymbol{a}$ and $\boldsymbol{\omega}$ are given in the standard (orthogonal and right-handed) basis, defined by the calibration system (as will be discussed in section 3). The raw accelerometer and gyroscope outputs of the sensors can be readily interpreted as real-valued vectors $\hat{\boldsymbol{a}} \in \mathbb{R}^3$ and $\hat{\boldsymbol{\omega}} \in \mathbb{R}^3$, respectively. Vectors $\hat{\boldsymbol{a}}$ and $\hat{\boldsymbol{\omega}}$ are given in a basis characteristic to a certain sensor triad within a certain IMU.

Now, let $\boldsymbol{S}_\mathrm{a} \in \mathbb{R}^{3\times3}$ be a diagonal matrix containing the scale factor errors, $\boldsymbol{M}_\mathrm{a} \in \mathbb{R}^{3\times3}$ a skew-symmetric matrix containing the misalignment and cross-coupling errors, $\boldsymbol{b}_\mathrm{a} \in \mathbb{R}^3$ the bias vector and $\boldsymbol{w}_\mathrm{a} \in \mathbb{R}^3$ normally distributed, zero-mean measurement noise. Then, a commonly exploited (e.g. [1, 22]) error model for the accelerometers is[1]

$$\hat{\boldsymbol{a}} = \boldsymbol{S}_\mathrm{a}\boldsymbol{a} + \boldsymbol{M}_\mathrm{a}\boldsymbol{a} + \boldsymbol{b}_\mathrm{a} + \boldsymbol{w}_\mathrm{a}. \tag{1}$$

In a similar fashion, a commonly exploited [1, 22] error model for the gyroscopes is

$$\hat{\boldsymbol{\omega}} = \boldsymbol{S}_\omega\boldsymbol{\omega} + \boldsymbol{M}_\omega\boldsymbol{\omega} + \boldsymbol{B}_\omega\boldsymbol{a} + \boldsymbol{b}_\omega + \boldsymbol{w}_\omega, \tag{2}$$

where $\boldsymbol{B}_\omega$ describes the g-dependence of the gyroscopes [1].

Keeping (1) in mind, consider the equation

$$\hat{\boldsymbol{a}} = \boldsymbol{A}\boldsymbol{a} + \boldsymbol{b}, \tag{3}$$

where $\hat{\boldsymbol{a}} \in \mathbb{R}^3$, $\boldsymbol{A} \in \mathbb{R}^{3\times3}$, $\boldsymbol{a} \in \mathbb{R}^3$ and $\boldsymbol{b} \in \mathbb{R}^3$. It is not difficult to see that (3) is actually the deterministic part of (1), where $\boldsymbol{A}$ contains the scale factors, misalignment and cross-correlation terms for each individual sensor along with other possible corrections which can be represented as a constant matrix. Vector $\boldsymbol{b} \in \mathbb{R}^3$ contains the constant bias terms of each sensor. Correspondingly for the gyroscopes, consider the equation

$$\hat{\boldsymbol{\omega}} = \boldsymbol{C}\boldsymbol{\omega} + \boldsymbol{d} + \boldsymbol{E}\boldsymbol{a}, \tag{4}$$

where $\hat{\boldsymbol{\omega}} \in \mathbb{R}^3$, $\boldsymbol{C} \in \mathbb{R}^{3\times3}$, $\boldsymbol{\omega} \in \mathbb{R}^3$, $\boldsymbol{d} \in \mathbb{R}^3$ and $\boldsymbol{E} \in \mathbb{R}^{3\times3}$. In (4) $\boldsymbol{C}$ has a similar meaning than $\boldsymbol{A}$ in (3), $\boldsymbol{d}$ contains the constant bias terms and $\boldsymbol{E}$ takes the g-dependence of the gyroscopes into account.

While (3) and (4) are in a convenient form considering the calibration, they are not in a convenient form to be readily exploited by the inertial navigation system. For that, we need to explicitly solve $\boldsymbol{a}$ and $\boldsymbol{\omega}$ from the equations, which yields

$$\boldsymbol{a} = \boldsymbol{A}^{-1}(\hat{\boldsymbol{a}} - \boldsymbol{b}) \tag{5}$$

[1] As we are dealing with real measurement data, there will always be some errors present that are unaccounted for. For the sake of readability, we decided not to include an error term in each approximate equation, but to overload the usage of equals sign a bit. We use equals sign also in situations where, strictly speaking, it should not be used. We are, however, sure that the exact meaning of each equation becomes evident from the context.

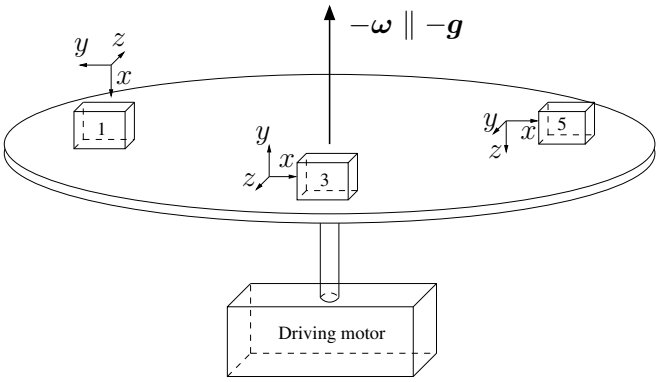

Figure 1. Schematic drawing of a rate table, where three different IMU positions (1, 3 and 5; see figure 2) are displayed.

$$\omega = C^{-1}(\hat{\omega} - d - Ea)\,. \tag{6}$$

In order for (5) and (6) to make sense, matrices A and C must be invertible, which are necessary conditions for the error models to be bijective relations.

As the formed error models (3) and (4) are clearly affine, they obviously cannot compensate for nonlinear errors present in a real situation. However, the constructed error models will be able to compensate for the major error sources of a typical IMU like the misalignment error, scale factor error and constant bias error. Note that the actual heading of each sensor and the order in which they are given in $\hat{a}$ and $\hat{\omega}$ will not matter as long as the measurement axes are not in the same plane. Thus, it is also possible to consider a certain IMU as a black box without further knowledge about the contents of the IMU.

3. Calibration system

In this section, the practical issues concerning the calibration system are discussed. In addition to general design rules, some remarks of the constructed calibration system are discussed to get an overall idea of the attainable level of accuracy.

As consumer-grade gyroscopes are typically not sensitive enough to measure Earth's angular rate, reference signals must be provided by another means. For this, a rate table with a user-controlled angular rate is used (see figure 1) [5, 8, 9]. While calibrating the gyroscopes, it is also convenient to provide accelerometers a number of reference measurements. This is done by using the accelerations created by the rotation along with the gravitational acceleration present in the measurements [20].

While concentrating on consumer-grade sensors, the reference signals are not required to be exactly known. Hence, it is safe to drop out a number of factors like the Earth's angular rate, Coriolis force (caused by the rotation of the Earth) and the changing rotation radius as a function of the angular rate (only open-loop sensors are subject to this). The effects of these will be negligible as compared to the overall accuracy of the sensors [24].

For the calibration system, the critical requirement is that the driving motor of the rate table is able to maintain a *constant angular rate* for a predetermined time (typically, a few seconds). There are several important reasons for this:

- the mean angular velocity over a longer period of time can be easily measured (see below) with an accuracy superior to the accuracy of a momentary value [24];
- the effects of the vibration present in a low-cost rate table can be significantly reduced by taking the mean of the output over a longer period of time and
- inherent delays in any inertial sensor do not affect the measurements.

In the constructed system, the realized angular rates are measured using an analog optical fork sensor, the output of which is recorded with a sampling frequency $f = 20$ kHz. In addition, a plate is attached to the rotating table cutting off the optical signal once per revolution. The measurement error of this kind of a device is analyzed in appendix A. With $k = 10$, for example, any mean angular velocity up to 1000 deg s^{-1} can be measured with an accuracy better than 5×10^{-4} deg s^{-1}. That is, with an accuracy that makes the neglected rotation of the Earth the limiting factor. The error in the reference accelerations are analyzed in appendix B. With the given level of accuracy in the angular rates, the accuracy of the reference accelerations is in practice characterized by the size of the IMU, which is discussed in the next section.

The rate table should be mounted horizontally in such a way that the gravitational acceleration will be perpendicular to the plane of the table. Fortunately, the system is not very sensitive to small mounting errors, since even an easily observed mounting error of 1° will only cause ≈1.5 mg error in the value of the gravitational acceleration. Moreover, other error components cancel out when the measurement data are averaged over several full revolutions of the table. In the constructed calibration system, the mounting error was less than 0.1°. Thus, in practice, the error in the gravitational acceleration is caused only by the limited accuracy of the externally measured reference value.

To cover all six degrees of freedom, it is required that the IMU is rotated in more than one position. The minimum number of different positions is three, with every axis of the IMU pointing once out of the plane. In this context, we will consider that the IMU can be positioned in a total of six different positions. This number of positions was chosen to cover positive and negative axes of each sensor. These positions are demonstrated in figure 2 along with the directions of the reference measurements g (gravitational acceleration), a_c (centripetal acceleration) and ω (angular velocity).

As suggested by figure 2, all of the six IMU positions are interpreted in such a way that certain axes of the calibrated IMU are collinear with the directions of the reference measurements. Because of this, the coordinates of the calibrated measurements are actually defined by the calibration system rather than the IMU. Note that it is only assumed that the different IMU positions are orthogonal with respect to each other, with no requirements for the absolute positions. In the constructed calibration system, the jig used to attach the IMU to the rate table is constructed using precision tools. Thus, the orthogonality of different positions will not be an issue.

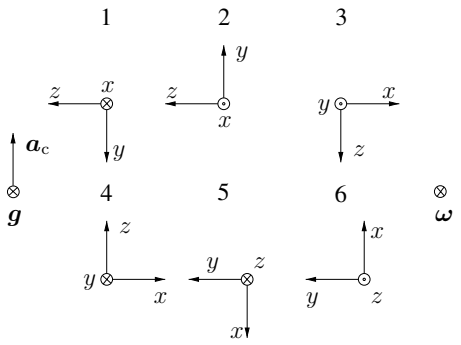

Figure 2. Six IMU positions shown by the directions of each orthogonal axis of the calibrated IMU along with the reference measurements. A dot in a circle represents a vector toward the reader and a cross in a circle represents a vector away from the reader.

For future reference, let us denote the total number of angular rates used to calibrate gyroscopes by N. Respectively, let the total number of angular rates used to calibrate accelerometers be M. With a total of six different positions, we have $6 \times N$ and $6 \times M$ measurements per single axis of the IMU. Note that M and N represent the set of rotation rates one plans to use in the computations, which is generally different than the set of actual rotation rates. In practice this could, for example, mean that one wishes to find several calibration functions, each of which is optimized to a certain range of inputs.

4. Calibration procedure

In this section, details of the calibration procedure are discussed. To give an overview of the calibration procedure, let us first state the steps for it as follows.

(i) Gather the needed data by rotating the IMU at a number of different angular rates for all six positions seen in figure 2.
(ii) Construct a set of numerical equations for the calibration parameters according to (3) and (4).
(iii) Solve the constructed, generally overdetermined and hence approximate set of equations.
(iv) Conduct a reality check for the computed results by comparing a number of control quantities against known reference values.

As step 1 is already clear, we will now discuss steps 2, 3 and 4 in detail.

4.1. Step 2: constructing the needed equations

Recall the discussion about the physical size of the IMU in section 1 and appendix B. There are two ways the physical size of the IMU will affect the output of an accelerometer located at a point different than the chosen origin of the IMU:

- tangential acceleration caused by the angular acceleration and
- centripetal acceleration caused by the angular velocity.

Table 1. Reference measurements for the accelerometers and gyroscopes when rotating the IMU in the six positions.

	Reference measurements					
	Acceleration			Angular rate		
Position	x	y	z	x	y	z
1	$-g$	$-\omega_{1,j}^2 r_1$	0	$\omega_{1,j}$	0	0
2	g	$\omega_{2,j}^2 r_2$	0	$-\omega_{2,j}$	0	0
3	0	g	$-\omega_{3,j}^2 r_3$	0	$-\omega_{3,j}$	0
4	0	$-g$	$\omega_{4,j}^2 r_4$	0	$\omega_{4,j}$	0
5	$-\omega_{5,j}^2 r_5$	0	$-g$	0	0	$\omega_{5,j}$
6	$\omega_{6,j}^2 r_6$	0	g	0	0	$-\omega_{6,j}$

As the calibration procedure exploits only constant angular rates, the observed tangential acceleration will be zero. Thus, we do not need to take this into account while constructing the needed equations. Furthermore, in a typical situation, the effects of tangential acceleration are difficult to compensate for, since the angular acceleration is unknown.

Centripetal acceleration caused by nonzero $\delta \boldsymbol{r}$ seen in appendix B is, however, observed. Thus, in general, each accelerometer has a unique rotating radius for each of the six positions of the IMU. For a 'small-sized' consumer-grade IMU, it is sometimes reasonable to assume that the effects caused by $\delta \boldsymbol{r}$ are negligible as compared to other sources of error. That is, the error estimates given in appendix B give values smaller than the expected errors in the calibrated sensors. When this is the case, we will only need one radius for each position. However, given a certain IMU, it is a far from trivial task to specify these externally with adequate accuracy. Thus, we leave these for the calibration routine to determine. This way, we do not need to commit to the location of the actual origin of the IMU in any way. Instead, we can leave it up to the calibration routine to specify the location of an origin minimizing the error caused by ignoring the size of the IMU. In the following treatment, it is assumed that the IMU is small. The presented methods can be readily generalized to the situation where this is not the case.

From figure 2, one will end up with the reference measurements seen in table 1. In the table, g is the local gravitational acceleration, $\omega_{i,j}$ the angular velocity of step j while the IMU is in position i and r_i is the rotation radius for position i.

Now, let us denote the reference acceleration of the step $j \in [1, \ldots, M]$ of the position $i \in [1, \ldots, 6]$ by $\boldsymbol{a}_{i,j}(r_i)$ and the respective measurements $\hat{\boldsymbol{a}}_{i,j}$. By recalling (3) and defining a 3×9 matrix

$$\boldsymbol{R}_{\mathrm{a}}(i, j, r_i) = \begin{bmatrix} \boldsymbol{a}_{i,j}^T(r_i) & \boldsymbol{0}^T & \boldsymbol{0}^T \\ \boldsymbol{0}^T & \boldsymbol{a}_{i,j}^T(r_i) & \boldsymbol{0}^T \\ \boldsymbol{0}^T & \boldsymbol{0}^T & \boldsymbol{a}_{i,j}^T(r_i) \end{bmatrix}, \tag{7}$$

we get an equation

$$[\boldsymbol{R}_{\mathrm{a}}(i, j, r_i) \quad -\boldsymbol{I}]\boldsymbol{x} = \hat{\boldsymbol{a}}_{i,j} \tag{8}$$

for vector $\boldsymbol{x} \in \mathbb{R}^{12}$ containing the elements of $\boldsymbol{A}$ and $\boldsymbol{b}$ in the rowwise order. The matrix $\boldsymbol{I}$ is a 3×3 identity matrix. Clearly, (8) is now a nonlinear equation because of the terms r_i in the coefficient matrix. When considering a single measurement

$\hat{\boldsymbol{a}}_{i,j}$, (8) is underdetermined with only three independent measurements for 18 unknowns. By stacking equations (8) for all i and j, one will get a system with a total of $3 \times 6 \times M$ equations for 18 unknowns. From this, it is clear that M must be at least 1 to have a well-posed problem. In practice, with several error sources unaccounted for, M should be larger thus making the resulting nonlinear equation overdetermined.

For the gyroscopes, let us first define a 3×9 matrix

$$\boldsymbol{R}_{\omega}(i,j) = \begin{bmatrix} \boldsymbol{\omega}_{i,j}^T & \boldsymbol{0}^T & \boldsymbol{0}^T \\ \boldsymbol{0}^T & \boldsymbol{\omega}_{i,j}^T & \boldsymbol{0}^T \\ \boldsymbol{0}^T & \boldsymbol{0}^T & \boldsymbol{\omega}_{i,j}^T \end{bmatrix} \tag{9}$$

containing the reference measurements. With a treatment similar to the previous paragraph and recalling (4), one will end up with an equation

$$[\boldsymbol{R}_{\omega}(i,j) \quad -\boldsymbol{R}_{\mathrm{a}}(i,j,r_i) \quad -\boldsymbol{I}]\boldsymbol{y} = \hat{\omega}_{i,j} \tag{10}$$

for vector $\boldsymbol{y} \in \mathbb{R}^{21}$ containing the elements of $\boldsymbol{C}$, $\boldsymbol{E}$ and $\boldsymbol{d}$ in the rowwise order. Again, when considering a single measurement $\hat{\omega}_{i,j}$, (10) is underdetermined with only three independent measurements for 21 unknowns. In this case, by stacking equations (10) for all i and j, we end up with a system with a total of $3 \times 6 \times N$ linear equations for 21 unknowns. The fact that (10) is linear is based on the assumption that (8) is already solved, giving access to parameters r_i.

4.2. Step 3: solving the constructed equations

In the previous section we constructed equations for the calibration functions we were looking for. As seen above, the equation for the accelerometers is nonlinear. Hence, we will have to use a nonlinear optimization routine to find out an optimal solution in some sense.

Let us denote the nonlinear system for the calibration parameters of the accelerometer triad in a form $\boldsymbol{z} = \boldsymbol{h}(\boldsymbol{x})$. Now we can write its residual $\boldsymbol{p}(\boldsymbol{x})$ as

$$\boldsymbol{p}(\boldsymbol{x}) = \boldsymbol{h}(\boldsymbol{x}) - \boldsymbol{z}. \tag{11}$$

A typically used criteria for an optimal solution is the minimum of the quadratic form $\|\boldsymbol{p}(\boldsymbol{x})\|^2$:

$$\tilde{\boldsymbol{x}} = \underset{\boldsymbol{x}}{\operatorname{argmin}} \ \boldsymbol{p}(\boldsymbol{x})^T \boldsymbol{p}(\boldsymbol{x}). \tag{12}$$

We will exploit Gauss–Newton to solve this nonlinear optimization problem [25]. For this, we will need the Jacobian of the function $\boldsymbol{h}(\boldsymbol{x})$, which we can compute using (8) and table 1. The algorithm goes as follows [25].

(i) Choose $\boldsymbol{x}_0$ and a suitable end criteria δ. Set $n = 0$.
(ii) Compute $\boldsymbol{J}_n = \frac{\mathrm{d}}{\mathrm{d}\boldsymbol{x}}\boldsymbol{h}(\boldsymbol{x}_n)$.
(iii) Compute $\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - (\boldsymbol{J}_n^T\boldsymbol{J}_n)^{-1}\boldsymbol{J}_n[\boldsymbol{h}(\boldsymbol{x}_n) - \boldsymbol{z}]$.
(iv) If $\|\boldsymbol{x}_{n+1} - \boldsymbol{x}_n\| \geqslant \delta$, set $n = n + 1$ and continue from step 2.

A suitable stopping criterion is the step length $\|\boldsymbol{x}_{n+1} - \boldsymbol{x}_n\| < \delta$.

As seen from the algorithm above, one must provide an initial guess $\boldsymbol{x}_0$. For a general nonlinear optimization problem, this is a nontrivial task having a great effect on the solution speed and possibly on the 'optimal' result obtained [25]. When this was tested using the test data considered later in this text, the choice of initial guess (provided that $\boldsymbol{J}_n^T\boldsymbol{J}_n$ is nonsingular) did not have any influence on the given solution or the solution speed. In each case, the optimization routine found the solution after a few iterations.

In the case of the gyroscopes, an approximate solution can be achieved simply by computing a least-squares solution for $\boldsymbol{y}$. When constructing the equation, the required $\boldsymbol{a}_{i,j}$ (r_i) can be provided by the known reference signals. Provided, of course, that the rotation radii are already known.

In some cases, one might have better knowledge about the reliability of the measurements or even knowledge about the correlation of different measurements. There is no problem in using readily available generalizations of the presented solution schemes to these situations [25]. In fact, as the proposed method is based on averaging the collected data, the variance of each measurement could be readily estimated as well.

4.3. Step 4: control quantities

In any practical situation, one should have some confidence about how successful the calibration was, before a particular IMU can be considered as ready to be used. For this, one can readily compute a number of control quantities right after the actual computation is done.

For an overall view about the sensitivity of the method to measurement errors, one can compute the condition numbers for the constructed matrices. For general non-square matrices, it is defined to be the condition number of the product of the transpose of the matrix and the matrix itself [26]. If reasonable values for the rotation radii are available before the computation, this check can be done before the actual computation takes place. Similarly, it is also possible to estimate the sensitivity of a particular IMU to measurement errors by computing the condition numbers of $\boldsymbol{A}$ and $\boldsymbol{C}$.

For a quantity describing how well the calibration function of the accelerometers fits the measurement data, one can estimate the standard deviation s_a of the residual (11) as follows [27]:

$$s_a = \sqrt{\frac{\boldsymbol{p}(\boldsymbol{x})^T \boldsymbol{p}(\boldsymbol{x})}{(3 \times 6 \times M) - 1}}. \tag{13}$$

This can be computed for the gyroscopes as well (s_g), by replacing $\boldsymbol{p}(\boldsymbol{x})$ with the residual of (10) and M by N.

In the case of accelerometers, matrix $\boldsymbol{J}_n^T\boldsymbol{J}_n$ evaluated at the optimum can be used to estimate the covariance matrix of the standard errors of the computed parameters [10, 28]. This gives a possibility of computing confidence intervals for the estimated parameters, if considered necessary. For the gyroscopes, this can be achieved by analyzing the coefficient matrix of the normal equations.

All of the control quantities provided above are standard methods in measurement science, which can be readily used to gain information about the calibration and goodness of fit. They can be readily used to compare different measurements, but to gain useful information about the absolute accuracy, knowledge of the actual sensors within the IMU is required.

Table 2. Calibration characteristics of a total of eight handmade IMUs.

	Accelerometers							Gyros
IMU	s_a	r_1	r_2	r_3	r_4	r_5	r_6	s_g
1	30.7	23.8 ± 0.3	26.6 ± 0.3	24.9 ± 0.3	25.7 ± 0.3	24.5 ± 0.3	25.9 ± 0.3	0.32
2	33.7	23.8 ± 0.3	26.7 ± 0.3	24.8 ± 0.3	25.6 ± 0.3	24.6 ± 0.3	25.8 ± 0.3	0.33
3	33.2	23.7 ± 0.3	26.6 ± 0.3	24.9 ± 0.3	25.7 ± 0.3	24.6 ± 0.3	25.9 ± 0.3	0.34
4	32.9	23.8 ± 0.3	26.7 ± 0.3	24.9 ± 0.3	25.7 ± 0.3	24.6 ± 0.3	25.8 ± 0.3	0.40
5	31.0	23.8 ± 0.3	26.5 ± 0.3	24.9 ± 0.3	25.7 ± 0.3	24.4 ± 0.3	25.8 ± 0.3	0.33
6	31.0	23.7 ± 0.3	26.7 ± 0.3	24.9 ± 0.3	25.7 ± 0.3	24.5 ± 0.3	25.9 ± 0.3	0.37
7	32.9	23.7 ± 0.3	26.8 ± 0.3	24.9 ± 0.3	25.5 ± 0.3	24.5 ± 0.3	25.8 ± 0.3	0.31
8	33.2	23.7 ± 0.3	26.7 ± 0.3	24.8 ± 0.3	25.7 ± 0.3	24.5 ± 0.3	25.7 ± 0.3	0.32

Radii r_i are expressed in cm. Standard deviations s_a and s_g are expressed in mg and deg s^{-1}, respectively. ± sign indicates 95% confidence intervals of the radii estimated using $s_a\sqrt{\text{trace}\left(\left(\boldsymbol{J}_n^T\boldsymbol{J}_n\right)^{-1}\right)}$ [28].

In typical situations this is acceptable, but this is not the case considering the black box situation. In this case, the computed rotation radii provide a suitable method to decide whether the calibration was successful or not. Consider, for example, that the value of the gravitational acceleration is given in units of g instead of m s^{-2}. As such, this is only going to scale the computed parameters differently, and will be visible only if the specifications of the used sensors are known. However, this error will be immediately seen in the values of the computed rotation radii.

5. Test results

The proposed calibration method was tested with a total of eight handmade IMUs. All IMUs were constructed in the same way, where each individual sensor was supposed to measure parallel to the x, y or z direction seen in figure 1. The test results are divided into three parts. The first part shows the differences in the calibration characteristics. The second part shows the actual difference between a calibrated and uncalibrated IMU as seen by the error models. Finally, the third part shows the differences in actual measurements. While a real navigation test is not included, the difference made by the calibration is clearly shown by the provided results.

The calibrations were performed using a 16-bit AD-converter to store the signals given by the IMU. 12 g accelerometers [29] were used along with 300 deg s^{-1} gyroscopes [30]. For the accelerometers it holds that $M = 30$ (recall that M is the number of the used angular rates to calibrate the accelerometers) and $N = 10$ for the gyroscopes. The used rotation rates were designed to follow approximately the following plan: (0–300) deg s^{-1} with 30 deg s^{-1} increments (up to 1 g), (1–3) g with 0.25 g increments, (3–6) g with 0.5 g increments and (6–12) g with 1 g increments (depending on the actual rotation radius). The size of the IMU was confirmed not to cause significant errors by simulations, and thus it was possible to use the approximation $\delta r \approx 0$.

5.1. Comparison between different IMUs

Let us next present a few characteristics (described in section 4.3) of each test collected in table 2.

Table 3. Numerical example of the calibration function (3).

$\boldsymbol{A}$ (g V^{-1})			$\boldsymbol{b}$ (V)
−6.929	0.048	−0.160	−2.351
−0.132	−7.005	−0.052	−2.369
−0.024	0.160	−7.020	−2.350

Because the measurements are expressed in bits, the most natural unit of parameters s_a and s_g seen in table 2 would also be bits. For clarity, however, they are converted to more meaningful units using the rule: 100 bits correspond to 50 mg for the accelerometers and 1.6 deg s^{-1} for the gyroscopes. These values are based on the scale factors of the sensors reported in [29] and [30].

From table 2, it is noted that the overall characteristics of the all eight IMUs are close to each other. Knowing that the IMUs are constructed by hand and that the computed radius is just an approximate value for any realistic IMU, the computed radii are consistent, with a typical 95% confidence interval of ±3 mm. The typical values for s_a and s_g for noncalibrated sensors are not displayed, since the constant bias alone would cause them to be in the order of 0.1 g and 1 deg s^{-1}, respectively.

5.2. Comparison of the calibration functions

In this section, the computed error models of the IMU number 1 are discussed in detail. This gives a possibility of comparing them against the situation where no calibration can be done. In this case, the values are taken directly from the specifications of the sensors. For clarity, the numerical values represented here are converted to standard units, i.e. voltages, g and degrees per second. Note, however, that the proposed method does not require this to be done.

In table 3, the parameters for the calibrated output of the accelerometers are shown. In case no calibration is done, the matrix $\boldsymbol{A}$ is a diagonal matrix having values −6.667 at the diagonal and $\boldsymbol{b}$ is a vector consisting of values $b_i = -2.350$ (i.e. the used accelerometers are set to show zero when placed on the desk). Note that there is a significant difference between the scale factors given by the manufacturer and the computed ones. This is explained by the fact that the used datalogger had, for some reason, a tendency to underestimate

Table 4. Numerical example of the calibration function (4).

$\boldsymbol{C}$ (deg s^{-1} V^{-1})			$\boldsymbol{E}$ (deg s^{-1} g^{-1})			$\boldsymbol{d}$ (V)
−208.5	−1.652	−1.953	−0.152	0.024	0.137	−2.574
1.051	−207.6	−0.601	0.194	−0.051	−0.163	−2.464
2.553	−0.526	208.4	−0.013	−0.043	0.027	−2.475

voltages. This does not affect the other results considered in this section: this is why the calibration should be done using the same datalogger that is used to store the data in action. The parameters for the calibrated output of the gyroscopes are shown in table 4. In case no calibration is performed, the scale factor is ± 166.667 depending on the sensor and bias value $b_i = -2.500$. As one could expect, the large difference in the scale factors is also observed here.

To give an overall idea about the error models, the diagonal elements of the matrices $\boldsymbol{A}$ and $\boldsymbol{C}$ correspond to the scale factors of each individual sensor, provided, of course, that the sensor readings are picked up in a 'correct' order. The minus sign in the scale factor simply means that the sensor is upside down. The off-diagonal parameters are caused by the misalignment and the cross-correlation of the sensors. If the off-diagonal terms are interpreted as a consequence of the geometrical alignment error alone, we can easily compute the corresponding error angles: by normalizing each column vector of $\boldsymbol{A}$ and $\boldsymbol{C}$, each element of the resulting six vectors is the sine of the respective error angle. For instance, the elements of the first column vector of $\boldsymbol{A}$ correspond to angles -88.89°, -1.09° and -0.20°. The first angle indicates that the corresponding sensor was aligned at an angle 178.89° with respect to the corresponding axis of the IMU.

This kind of a geometrical interpretation cannot be made for the matrix $\boldsymbol{E}$ representing the dependence of the measured angular rate from the linear acceleration seen in table 4. In this particular case, all the elements of the respective matrix are close to each other indicating that the magnitude of the coupling between linear acceleration and measured angular rate is about the same for all axes.

5.3. Comparison of the measurement accuracy

In this section, the gain of the calibration procedure performed to the IMU number 1 is discussed. Note that the accuracy tests are performed on the same rate table that was used to calibrate data. This was the only possibility of generating accurate reference accelerations up to 12 g and reference angular rates up to 300 deg s^{-1}, since we did not have the possibility of exploiting high-accuracy reference sensors. However, the angular rates and accelerations used to demonstrate the accuracy were *not* used in the calibration. Furthermore, the accuracy tests and the calibration routine were run as two separate events. The run-to-run variation of the sensor bias was taken into account by averaging the output before the accuracy test took place.

In the case of the accelerometers, the used sensor error model is identical to the ones used before. Only the calibration method has changed from the basic six-position calibration method to the proposed method. The accuracy of these two methods are compared against the reference accelerations. The positions used for the six-position calibration are the ones used in the proposed method, while the rate table was not rotating. In the case of the gyroscopes, we use an error model different from the ones used in the referred studies. Namely, the g-dependent bias is also taken into account. In this case the calibration method is basically the same than the one seen, for example, in [5]. The accuracy of the two models are compared against the reference angular rates. The results corresponding to the two choices of error models were both computed using the same measurements.

In table 5, the calibrated results of the accelerometer triad are compared to the reference signals in all six positions seen in figure 2 while the rate table was rotated at two different angular rates. The angular rates were selected in such a way that both ends of the accelerometer's dynamic range were covered. The reference signals are seen on the left and the output of the accelerometer triad using the proposed calibration method at the center. The results given by the standard six-position calibration method are seen on the right.

The results given by the standard six-position calibration method are somewhat better whenever the total reference acceleration is close to 1 g. This is expectable since this method uses accelerations up to 1 g, whereas the proposed method uses the whole dynamic range of the accelerometers. With higher accelerations, the proposed calibration method gives a better overall accuracy. This can be verified by computing the value s_a for the standard six-position calibration method, which gives $s_a = 80.2$ mg against a typical value of 32 mg obtained with the proposed method. Hence, the proposed calibration method gives better results than the standard six-position calibration, when considering the whole dynamic range of the sensors. One reason for this is that the six-position calibration method relies on extrapolation when applied to sensors with a range exceeding 1 g. Secondly, small alignment errors are not necessarily observable due to the limited sensitivity of the sensors.

For the gyroscopes, more specific calibration results can be found in table 6. The reference and measured angular rates for each of the six positions are tabulated for two angular rates at both ends of the dynamic range. The reference angular rate is seen on the left, proposed calibration (with g-dependence) output of the gyroscope triad at the center and output of the method proposed in [5] (without g-dependence) on the right.

In this case, the overall performance of the two calibration functions are close to each other. This is quantified by the value $s_g = 0.33$ deg s^{-1} for the method where g-dependence is not considered. This is larger than the corresponding value $s_g = 0.32$ deg s^{-1}, but considering the variation in these values between different IMUs, this is hardly a significant difference. This is explained by the fact that during the test the total acceleration acting upon the IMU was at most 1.2 g, because rotations rates up to 300 deg s^{-1} do not cause centripetal accelerations larger than 0.7 g with the given radii. As table 4 proposes, the difference of the methods should not be that dramatic in these kind of circumstances. Angular velocity of $[0.0 \;\; 1110.3 \;\; 0.0]^T$ deg s^{-1}, on the other hand, causes a total acceleration of ≈ 10 g. In this case the angular velocity given

Table 5. Results of the accuracy test performed for the accelerometers of the IMU number 1. The expected values are the reference accelerations computed with the known values of the rotation radii.

Expected (g)			Measured acceleration (g)					
			Proposed method			Six-position method		
x	y	z	x	y	z	x	y	z
−1.000	−0.007	0.000	−0.991	0.014	−0.013	−1.003	−0.002	0.006
1.000	0.007	0.000	1.008	0.043	0.003	0.998	0.012	0.008
0.000	1.000	−0.007	0.015	1.020	−0.020	0.009	0.995	−0.004
0.000	−1.000	0.007	0.013	−0.978	0.012	−0.002	−1.000	0.019
−0.007	0.000	−1.000	0.005	0.037	−1.004	−0.003	0.018	−0.992
0.007	0.000	1.000	0.038	0.037	0.990	0.025	0.011	1.002
−1.000	−9.096	0.000	−1.007	−9.095	0.135	−1.061	−9.099	0.117
1.000	10.172	0.000	0.988	10.174	0.118	1.024	10.129	0.164
0.000	1.000	−9.517	−0.071	1.004	−9.483	−0.055	1.018	−9.472
0.000	−1.000	9.840	−0.061	−0.998	9.866	−0.100	−1.058	9.881
−9.371	0.000	−1.000	−9.358	−0.196	−1.040	−9.376	−0.144	−0.960
9.921	0.000	1.000	9.927	−0.191	0.957	9.922	−0.293	0.894

Table 6. Results of the accuracy test performed for the gyroscopes of the IMU number 1. The expected values are the reference rotation rates.

Expected (deg s^{-1})			Measured angular rate (deg s^{-1})					
			Proposed method			Method by Syed *et al* [5]		
x	y	z	x	y	z	x	y	z
30.2	0.0	0.0	30.3	0.1	−0.1	30.3	0.3	−0.1
−30.0	0.0	0.0	−30.2	0.0	0.0	−30.1	−0.2	0.0
0.0	−29.9	0.0	−0.2	−29.7	0.1	−0.3	−29.7	0.1
0.0	29.9	0.0	−0.2	29.9	0.1	−0.1	29.8	0.0
0.0	0.0	29.9	0.4	0.2	30.2	0.6	0.1	30.2
0.0	0.0	−29.9	0.4	0.2	−29.8	0.3	0.3	−29.9
305.8	0.0	0.0	306.4	−0.9	−0.5	306.4	−1.0	−0.5
−305.7	0.0	0.0	−305.3	−0.8	−0.3	−305.4	−0.7	−0.3
0.0	−305.6	0.0	−0.2	−305.3	0.7	−0.2	−305.3	0.7
0.0	305.6	0.0	−0.1	306.0	0.8	−0.2	306.0	0.8
0.0	0.0	305.7	−0.7	0.0	305.2	−0.8	0.1	305.1
0.0	0.0	−305.6	−0.4	0.1	−306.0	−0.3	0.0	−306.0

by the proposed method is $[-3.6 \;\; 484.0 \;\; 0.5]^{\mathrm{T}}$ deg s^{-1}, and the angular velocity given by the method proposed by [5] is $[-5.1 \;\; 484.2 \;\; 0.9]^{\mathrm{T}}$ deg s^{-1}. Obviously, the y-components are useless in both cases because the rate exceeds the range of the used sensors. In the other components, however, there is a clear improvement over the method proposed in [5].

6. Conclusions

With the methods described in this paper, it is possible to enhance the overall accuracy of an IMU to a better level than the standard six-position calibration would allow. In the conducted experiments for accelerometers, for example, the standard deviation of the residual error is 32.0 mg against the value of 80.2 mg acquired using the standard calibration. In the case of the gyroscopes, it is demonstrated that under low accelerations, the results are comparable to known calibration methods (residual error 0.32 deg s^{-1} versus 0.33 deg s^{-1}). Under high accelerations, the proposed method can significantly increase the accuracy. This can be achieved without a detailed knowledge about the sensors within the IMU, as long as it is capable of measuring general rotations and accelerations. For example, the IMU could include sensors with different dynamic ranges thus causing the scale factors to be different for each axis. One does not need to know this beforehand, as the proposed method does not require any prior knowledge about the parameters to be computed. Using the provided control quantities, it can be easily verified if the calibration process was successful or not. The proposed method also allows one to compute error models specialized in a certain range of inputs, expanding the use of sensors with a fixed range without the need for recalibration.

As the focus is on consumer-grade IMUs, the used reference angular rates must be provided by a rate table of some sort. However, the design requirements for this are fulfilled without costly and precise equipment, because the calibration process uses only a set of constant angular velocities. A rate table provides a way to perform more complete calibration routines using also the centripetal accelerations as reference accelerations. The rotation radii do not need to be known, as the provided solution method can solve for these along with the actual calibration parameters.

The calibration functions presented here are not the only possible choices, but examples of accurate, but still simple

calibration functions. One can also use a different error model important in a particular application, as the total number of the measurements gives a possibility of doing this. While the proposed method is based on the use of affine functions, the use of non-affine calibration functions could also be possible at least when good initial guesses are available. Another interesting point is how to exploit redundant sensor configurations. Basically, the proposed method can be readily generalized to these situations, since we would only need to consider the rectangular matrices $\boldsymbol{A}$, $\boldsymbol{C}$ and $\boldsymbol{D}$ instead of the square ones considered here. This will, however, introduce questions about the uniqueness of the solution, which need to be examined first.

While the provided calibration functions compensate for the main error sources (scale factor, misalignment, cross correlation and bias errors) of consumer-grade IMUs, they are simple enough to be used in any INS. This is because only simple matrix multiplications and vector additions are required, and these can be applied directly to the raw data. However, as the method is based on a relatively simple calibration process, it cannot compensate for more complicated error sources such as run-to-run variations of the biases. When these kinds of corrections are necessary, they can be applied to the calibrated output of the IMU.

Appendix A. Error analysis of the measured mean angular velocity

Before going on, observe that the rotation angle θ is known to be exactly $2\pi k$, where $k \in \mathbb{N}$ is the number of revolutions. This follows from the type of the measurement explained in section 3, and it plays a central role in the error analysis.

The measured average angular rate $\hat{\omega}_{\text{avg}}$ and the true average angular rate ω_{avg} satisfy

$$\delta\omega_{\text{avg}} = \hat{\omega}_{\text{avg}} - \omega_{\text{avg}}, \tag{A.1}$$

where $\delta\omega_{\text{avg}}$ is the error in the measured mean angular velocity. Similarly for the rotation time T, it holds that

$$\delta T = \hat{T} - T. \tag{A.2}$$

Now, since the mean angular velocity over a certain time period is defined as

$$\omega_{\text{avg}} = \frac{\theta}{T}, \tag{A.3}$$

$\delta\omega_{\text{avg}}$ can be written as

$$\delta\omega_{\text{avg}} = \frac{\theta}{T+\delta T} - \frac{\theta}{T} = -\frac{\theta\delta T}{T\,(T+\delta T)}. \tag{A.4}$$

From (A.3) we know that

$$T = \frac{\theta}{\omega_{\text{avg}}} = \frac{2\pi k}{\omega_{\text{avg}}} \tag{A.5}$$

holds.

In order to quantify $\delta\omega_{\text{avg}}$, we need to quantify δT. For this, we need to assume that

- internal delay of the optical fork sensor is a constant and
- clock drift of the AD converter is negligible.

Provided that these assumptions are correct and that the sampling frequency of the AD converter is f, it holds that

$$\delta T = \frac{t_\epsilon}{f}, \qquad -1 < t_\epsilon < 1. \tag{A.6}$$

That is, the error in the rotation time is at most the time between two adjacent samples given by the AD converter.

Putting these together and taking absolute values, we get

$$|\delta\omega_{\text{avg}}| = \frac{|\theta||\delta T|}{|T||T+\delta T|} < \frac{|\omega_{\text{avg}}|}{f\left|\frac{2\pi k}{\omega_{\text{avg}}} + \frac{t_\epsilon}{f}\right|}. \tag{A.7}$$

When it holds that $2\pi kf > |\omega_{\text{avg}}|$ (which will be the case for any reasonable choice of f and ω_{avg}), (A.7) can be further reduced to

$$|\delta\omega_{\text{avg}}| < \frac{\omega_{\text{avg}}^2}{2\pi kf - |\omega_{\text{avg}}|}. \tag{A.8}$$

Appendix B. Error analysis of the measured acceleration

Let us use similar notation as in appendix A, now treated as vectors. That is, the computed reference acceleration $\hat{\boldsymbol{a}}$ and the true reference acceleration $\boldsymbol{a}$ satisfy

$$\delta\boldsymbol{a} = \hat{\boldsymbol{a}} - \boldsymbol{a}, \tag{B.1}$$

where $\delta\boldsymbol{a}$ is the error in the measured centripetal acceleration. In the following, zero angular acceleration is assumed, $\boldsymbol{\alpha} \equiv 0$. Hence, the acceleration of an arbitrary point $\boldsymbol{r}$ fixed to the rate table is

$$\boldsymbol{a} = \ddot{\boldsymbol{r}} + \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r})\,. \tag{B.2}$$

Note that $\boldsymbol{r}$ describes the position of the origin of the IMU and the position of a certain sensor with respect to the origin of the IMU is denoted by $\delta\boldsymbol{r}$. These vectors rotate with the rate table.

Putting these together, $\delta\boldsymbol{a}$ can be written as

$$\begin{aligned}\delta\boldsymbol{a} &= \hat{\boldsymbol{a}} - \boldsymbol{a}\\ &= \ddot{\boldsymbol{r}} + \delta\ddot{\boldsymbol{r}} + (\boldsymbol{\omega}+\delta\boldsymbol{\omega}) \times [(\boldsymbol{\omega}+\delta\boldsymbol{\omega}) \times (\boldsymbol{r}+\delta\boldsymbol{r})]\\ &\quad - \ddot{\boldsymbol{r}} - \boldsymbol{\omega} \times (\boldsymbol{\omega} \times \boldsymbol{r})\,,\end{aligned} \tag{B.3}$$

where $\delta\ddot{\boldsymbol{r}} \equiv 0$ holds. Exploiting the properties of the cross product, it is possible to derive the following upper limit for $\|\delta\boldsymbol{a}\|$:

$$\begin{aligned}\|\delta\boldsymbol{a}\| &\leqslant \|\boldsymbol{\omega}\|^2 \|\delta\boldsymbol{r}\| + 2\|\boldsymbol{\omega}\| \|\boldsymbol{r}\| \|\delta\boldsymbol{\omega}\|\\ &\quad + 2\|\boldsymbol{\omega}\| \|\delta\boldsymbol{r}\| \|\delta\boldsymbol{\omega}\| + \|\boldsymbol{r}\| \|\delta\boldsymbol{\omega}\|^2 + \|\delta\boldsymbol{r}\| \|\delta\boldsymbol{\omega}\|^2\,.\end{aligned} \tag{B.4}$$

Assuming that $\|\delta\boldsymbol{r}\| / \|\boldsymbol{r}\|$ and $\|\delta\boldsymbol{\omega}\| / \|\boldsymbol{\omega}\|$ are small and using the notation from appendix A, we have

$$\begin{aligned}\frac{\delta a}{a} = \frac{\delta a}{\omega_{\text{avg}}^2 r} &\leqslant \frac{\delta r}{r} + 2\frac{\delta\omega_{\text{avg}}}{\omega_{\text{avg}}} + \mathcal{O}\left(\frac{\delta r}{r}\frac{\delta\omega_{\text{avg}}}{\omega_{\text{avg}}}, \frac{\delta\omega_{\text{avg}}^2}{\omega_{\text{avg}}^2}\right)\\ &\quad + \mathcal{O}\left(\frac{\delta r}{r}\frac{\delta\omega_{\text{avg}}^2}{\omega_{\text{avg}}^2}\right).\end{aligned} \tag{B.5}$$

(B.5) contains terms caused by the physical size of the IMU (δr) and by the measurement error of the angular velocity

($\delta\omega_{\text{avg}}$). Typically, the first term dominates and thus, as an overall 'rule of thumb',

$$\delta a \approx \frac{\delta r}{r} a. \tag{B.6}$$

In practice, this estimate has a tendency to be quite pessimistic. This is because in (B.4), the worst case requires a certain direction for the measurement axis of the respective sensor in addition to orthogonality of $\delta \boldsymbol{r}$ and $\boldsymbol{\omega}$.

References

[1] Titterton D H and Weston J L 2004 *Strapdown Inertial Navigation Technology* 2nd edn (London: Institution of Engineering and Technology/Reston, VA: The American Institute of Aeronautics and Astronautics)

[2] Farrell J A and Barth M 1998 *The Global Positioning System and Inertial Navigation* (New York: McGraw-Hill)

[3] Grewal M S, Henderson V D and Miyasako R S 1991 Application of Kalman filtering to the calibration and alignment of inertial navigation systems *IEEE Trans. Autom. Control* **36** 3–13

[4] Chatfield A B 1997 *Fundamentals of High Accuracy Inertial Navigation* (Reston, VA: The American Institute of Aeronautics and Astronautics)

[5] Syed Z F, Aggarwal P, Goodall C, Niu X and El-Sheimy N 2007 A new multi-position calibration method for MEMS inertial navigation systems *Meas. Sci. Technol.* **18** 1897–907

[6] Fong W T, Ong S K and Nee A Y C 2008 Methods for in-field user calibration of an inertial measurement unit without external equipment *Meas. Sci. Technol.* **19** 085202

[7] Lötters J C, Schipper J, Veltink P H, Olthuis W and Bergveld P 1998 Procedure for in-use calibration of triaxial accelerometers in medical applications *Sensors Actuators* A **68** 221–8

[8] Cho S Y and Park C G 2005 A calibration technique for a redundant IMU containing low-grade inertial sensors *ETRI J.* **27** 418–26

[9] Hwangbo M and Kanade T 2008 Factorization-based calibration method for MEMS inertial measurement unit *IEEE Int. Conf. Robotics and Automation* pp 1306–11

[10] Skog I and Händel P 2006 Calibration of a MEMS inertial measurement unit *XVII IMEKO World Congress (Rio de Janeiro, Brazil, 17–22 September)*

[11] Aslan G and Saranli A 2008 Characterization and calibration of MEMS inertial measurement units *16th European Signal Processing Conf. (Lausanne, Switzerland, 25–29 August)*

[12] Wu Z C, Wang Z F and Ge Y 2002 Gravity based online calibration for monolithic triaxial accelerometers' gain and offset drift *Proc. 4th World Congress on Intelligent Control and Automation (Shanghai, People's Republic of China, 10–14 June)*

[13] Zappa B, Legnani G, van den Bogert A J and Adamini R 2001 On the number and placement of accelerometers for angular velocity and acceleration determination *J. Dyn. Syst.* **123** 552–4

[14] Parsa K, Angeles J and Misra A K 2002 Attitude calibration of an accelerometer array *Proc. IEEE Int. Conf. Robotics and Automation (Washington DC, 11–15 May)* vol **1** pp 129–34

[15] Park S, Tan C-W and Park J 2005 A scheme for improving the performance of a gyroscope-free inertial measurement unit *Sensors Actuators* A **121** 410–20

[16] Parsa K and Lasky T A 2007 Design and implementation of a mechatronic, all-accelerometer inertial measurement unit *IEEE/ASME Trans. Mechatronics* **12** 6

[17] Hung C-Y and Lee S-C 2006 A calibration method for six-accelerometer INS *Int. J. Control Autom. Syst.* **4** 615–23

[18] Cappa P, Patanè F and Rossi S 2008 Two calibration procedures for a gyroscope-free inertial measurement system based on a double-pendulum apparatus *Meas. Sci. Technol.* **19** 055204

[19] Umeda A, Onoe M, Sakata K, Fukushia T, Kanari K, Iioka H and Kobayashi T 2004 Calibration of three-axis accelerometers using a three-dimensional vibration generator and three laser interferometers *Sensors Actuators* A **114** 93–101

[20] Alves J, Lobo J and Dias J 2003 Camera-inertial sensor modeling and alignment for visual navigation *J. Mach. Intell. & Robot. Control* **5** 103–11

[21] Nebot E and Durrant-Whyte H 1999 Initial calibration and alignment of low-cost inertial navigation units for land vehicle applications *J. Robot. Syst.* **16** 81–92

[22] Grewal M S, Weill L R and Andrews A P 2001 *Global Positioning Systems, Inertial Navigation, and Integration* (New York: Wiley)

[23] Bossavit A 1998 *Computational Electromagnetism* (Berlin: Academic)

[24] IEEE Aerospace and Electronic Systems Society 2001 *IEEE Recommended Practice for Precision Centrifuge Testing of Linear Accelerometers* (The American Institute of Aeronautics and Astronautics) www.ieee-aess.org

[25] Bertsekas D P 1995 *Nonlinear Programming* (Nashua, NH: Athena Scientific)

[26] Demmel J W 1997 *Applied Numerical Linear Algebra* (Berkeley, CA: Society for Industrial and Applied Mathematics)

[27] Rosenkrantz W A 1997 *Introduction to Probability and Statistics for Scientists and Engineers* (New York: McGraw-Hill)

[28] Press W H, Flannery B P, Teukolsky S A and Vetterling W T 1992 *Numerical Recipes in C (The Art of Scientific Computing)* 2nd edn (Cambridge: Cambridge University Press)

[29] *VTI SCA620-CHCV1A Datasheet (Revision 2/2)* checked on 22 December 2009

[30] *Analog Devices ADXRS300ABG Datasheet (Revision B)* checked on 22 December 2009

Publication 2

Tuukka Nieminen, Jari Kangas, Saku Suuriniemi and Lauri Kettunen
"Accuracy Improvements by Boundary Conditions for Inertial Navigation"
International Journal of Navigation and Observation (10pp), 2010
doi:10.1155/2010/869127

Hindawi Publishing Corporation
International Journal of Navigation and Observation
Volume 2010, Article ID 869127, 10 pages
doi:10.1155/2010/869127

Research Article

Accuracy Improvement by Boundary Conditions for Inertial Navigation

Tuukka Nieminen, Jari Kangas, Saku Suuriniemi, and Lauri Kettunen

Tampere University of Technology, Department of Electronics, Unit of Electromagnetics, P.O. Box 692, 33101 Tampere, Finland

Correspondence should be addressed to Tuukka Nieminen, tuukka.nieminen@tut.fi

Received 16 September 2009; Revised 8 January 2010; Accepted 12 May 2010

Academic Editor: Paul Cross

The term *inertial navigation* is often automatically associated with the term *initial value problem*. However, there are many applications where it is possible to end up with a *boundary value problem* (BVP) as well. We show that in case of a BVP, the *finite element method* that incorporates *boundary conditions* can be efficiently used to compute position and velocity estimates not prone to accumulation of errors. For further accuracy enhancements, a method of combining inertial measurements with additional constraints is proposed. This way, we can model sensor errors, known to limit the accuracy of the system. The capabilities of the proposed methods are demonstrated with real-life examples.

1. Introduction

Typically, inertial measurements are made to have estimates of current position and velocity in real time. The set of equations used to compute the position and velocity estimates out of the actual measurements depends greatly on the application in hand. Equations for a general navigation application are presented, for example, in [1] and [2–5]. In this study, however, we shall concentrate on the growing market of *consumer applications* employing inertial sensors within a suitable price (and quality) range. Thus, we can simplify the equations needed to compute the *position and velocity estimates*. This is so, because in this case the errors are more likely to be determined by the limited accuracy of the sensors rather than the accuracy of the used equations.

In terms of the simplified equations, we basically need to solve the following problem: given $\mathbf{a}(t) : \mathbb{R} \rightarrow \mathbb{R}^3$, find $\mathbf{r}(t) : \mathbb{R} \rightarrow \mathbb{R}^3$ such that $\mathbf{r}$ satisfies

$$\ddot{\mathbf{r}}(t) = \mathbf{a}(t). \tag{1}$$

In (1), "¨" represents the second time derivative, $\mathbf{r}(t)$ is the position of the object in a suitable coordinate frame, $\mathbf{a}(t)$ represents the acceleration of the object in the same coordinate frame, and t is time. The velocity of the object ($\dot{\mathbf{r}}$), given in the same coordinate frame as $\mathbf{a}(t)$ and $\mathbf{r}(t)$, is denoted as $\mathbf{v}(t)$. Notice that the form (1) follows from the more general navigation situation by neglecting the rotation and "curvature" of the Earth.

Now, from the theory of ordinary differential equations (ODEs), we know that we need exactly two independent constraints to unambiguously solve a 2nd order ODE (1). Particularly, if these two independent constraints are both given at time $t = T_0$, we have an *initial value problem* (IVP) [6]: given $\mathbf{a}(t)$, $\mathbf{v}(T_0)$, *and* $\mathbf{r}(T_0)$, find $\mathbf{r}(t)$ such that

$$\begin{aligned} \ddot{\mathbf{r}}(t) &= \mathbf{a}(t), \\ \mathbf{v}(T_0) &= \mathbf{v}_0, \\ \mathbf{r}(T_0) &= \mathbf{r}_0 \end{aligned} \tag{2}$$

hold. Problems that are not IVPs based on the above definition are called *boundary value problems* (BVPs). A special case of a BVP, convenient to our purposes, is stated as follows: given $\mathbf{a}(t)$, one of the constraints

$$\mathbf{v}(T_0) = \mathbf{v}_0 \quad \text{or} \quad \mathbf{r}(T_0) = \mathbf{r}_0, \tag{3}$$

and one of the constraints

$$\mathbf{v}(T_1) = \mathbf{v}_1 \quad \text{or} \quad \mathbf{r}(T_1) = \mathbf{r}_1, \tag{4}$$

find $\mathbf{r}(t)$ for $T_0 \leq t \leq T_1$ such that it satisfies

$$\ddot{\mathbf{r}}(t) = \mathbf{a}(t),$$
$$\mathbf{v}(T_0) = \mathbf{v}_0, \quad \text{or} \quad \mathbf{r}(T_0) = \mathbf{r}_0, \tag{5}$$
$$\mathbf{v}(T_1) = \mathbf{v}_1 \quad \text{or} \quad \mathbf{r}(T_1) = \mathbf{r}_1.$$

We know for a fact that a problem of the form (5) has a unique solution whenever the position is fixed at least at one boundary: This knowledge comes from the fact that (7) is a one-dimensional case of a certain class of partial differential equations called Poisson problems, properties of which are well known [7].

Notice that there is a natural reason why problems (2) and (5) are distinguished. Namely, the solution methods for IVPs and BVPs differ substantially from each other [6, 8]. This is also where this paper differs from numerous research articles considering inertial navigation: previously, the problems have been given in the form (2), whereas we consider problems of the form (5).

The key assumption of our approach is the following: *inertial measurements related to a certain time period, including a set of boundary values and possibly some additional constraints, are all available when processed.* The length of the time period (i.e., *domain*) $[T_0, T_1]$ in (5) can be anything from fractions of a second to some minutes. There are different ways we may end up with a BVP of the form (5): firstly, the underlying problem may naturally be a BVP. Secondly, we may have an IVP, which we can pose as a BVP. This, of course, requires that we also know something about the result at the end of the interesting event and that we can afford to wait for the results until the end of the event. In the first case, we generally do not have any additional constraints we could use, but in the second case we end up with a BVP and at least one additional constraint. We will discuss examples of both cases in the next section.

The case where we only know the velocity at both ends is, however, special and deserves some attention. According to the discussion above, we do not have a unique solution for this kind of a BVP although it fits the definition of our model BVP (5). It is neither a valid IVP according to (2). As it turns out, it is possible to solve also this without loss of generality. This is based on the fact that we can always fix the position at one boundary without changing the "shape" of the solution. Then, we will come up with a well-defined BVP with an additional constraint.

The overall scope of this paper is to show how to treat inertial navigation problems that are naturally (or knowingly) posed as a BVP of the form (5). Attitude computations are not considered, and where necessary, attitude is assumed to be available. The form of problem (5) allows us to consider it as a set of three one-dimensional (1D) problems, rather than one three-dimensional (3D) problem. Notice that two boundary values per a dimension are required to obtain a unique solution. On the other hand, there is no need for the boundary values for different dimensions to be of the same type.

This paper is organized as follows: in Section 2, preliminaries are discussed. In Sections 3 and 4, we will show how to exploit 1D finite element method (FEM) to solve the underlying BVP with various possible choices of boundary conditions. At this stage, we assume that no additional constraints are given and that the measurements are exact. In Section 5, we face the reality with faulty measurements and exploit linear additional constraints to enhance the accuracy of the results. The underlying BVP is treated as exact, but the measurements are corrected using a linear sensor error model. As a result, we get two systems of linear equations: an exact one for the BVP and possibly an overdetermined one for the parameters of the sensor error model. We will solve these together to yield the corrected results. Finally, two real-life examples are discussed in Sections 6 and 7 before the conclusions in Section 8.

2. Preliminaries

Let us first motivate the chosen approach by means of examples: an application suitable to our approach is ski jumping, which has been the prime motivation of this study [9]. As an inertial navigation problem, it includes the use of consumer grade sensors with knowledge of the boundary values (in this case, position at both ends of the event). See Section 7 for further details.

It turns out that especially *sports applications* tend to have properties that are well suited to the considered approach: the included actions are often periodic, in such a way that the certain short-term action (e.g., a single step) repeats several times or some longer-term action a few times (e.g., a single lap or a single jump as discussed earlier) during a certain event. In these kinds of applications, it is natural to encounter problems of the form (5) rather than (2): for example, in long jump (considering only the jump part), at time T_0 ("take off"), the velocity of the shoe is known but position is not and at time T_1 ("landing"), the position of the shoe when it hits the surface of the sand can be accurately measured but the velocity is unknown.

For a more general view, even a GPS-assisted inertial navigation system can be considered as a series of separate navigation periods with given boundary values rather than a single event with additional constraints given at certain time instances. This is an example of an IVP, which can be posed as a BVP.

2.1. Some Remarks. There are two main classes of numerical methods for BVP's. One class includes so-called shooting methods and the other class methods of weighted residuals such as FEM [8, 10]. We concentrate on the latter because of its property to minimize an error norm over the whole integration interval rather than minimizing only the local error [7]. Another tempting property is that it concerns the position directly, giving us a possibility to more easily handle various types of additional constraints we will encounter later on.

In many applications, inertial measurements are not the only source of information. In practice, however, the number or the type of these additional constraints—combined with the different kinds of a solution method—does not suggest

the use of the traditional filtering (see Section 5 for details). For these situations, we will introduce a computationally cheap and easily exploitable method to enhance the accuracy of the position and velocity estimates. The proposed method is based on sensor error modeling and is characterized by the following assumptions.

(i) Sensor errors are modeled as constant errors. While the behavior of a certain sensor error is in real life a stochastic process, one is usually able to fairly model it at least momentarily as a constant error. A suitable mathematical tool to characterize this is the *Allan variance* [11].

(ii) Considering consumer grade sensors, causes of the most significant errors are usually known (e.g., bias and scale factor error, both changing from turn-on to turn-on [2]).

(iii) In particular, the sensor noise is *not* modeled. This is a conscious modeling decision to prevent unnecessary "smoothing" of data.

(iv) Additional constraints are treated as "exact". That is, the overall error in the additional constraints is assumed to be smaller than the error caused by the simplification of the navigation equations.

When additional constraints are available, problems resembling the ones considered here have previously been resolved using the means of *fixed interval smoothing* (or "Kalman smoother", if the type of the filter is fixed) [12–15]. Considering inertial navigation, the most used methods of solving the problems are the *two-filter smoother* [16] and *Rauch-Tung-Striebel smoother* [17], used for example in [15, 18–20]. In both cases, the problem itself is posed as an IVP and the basic idea is to run the filter in the forward direction as a "predictor" in phase one and then to run the filter in the backward direction while combining these two results to yield the corrected result in phase two. Between the proposed method and the fixed interval smoothing method, the fundamental difference is that in the proposed method the dynamics model is based on the BVP formulation and in the previous approaches, on the IVP formulation with additional smoothing.

Comparing the fixed interval smoothing technique to the proposed method, in addition to the previously mentioned points, the most significant differences are as follows.

(i) As fixed interval smoothing is run in both directions, the filter needs two process models, which can be problematic [21]. As the proposed method is based on the BVP formulation, it does not make a distinction between forward and backward directions.

(ii) In the filtering approach, each additional constraint is assumed to be attached to a single time instance [2], while the proposed method does not make such a restriction (see Section 5 for details).

(iii) Fixed interval smoothing is a two-phase method requiring numerous computations per time step [2, 15], whereas the proposed approach is a single-phase method with only few computations per an unknown.

Due to the significant differences in these two approaches, we will in this context concentrate on the proposed method. Obviously, there are situations where the two methods could both be used. Comparison of the methods in such a situation is interesting, although not addressed in this paper.

Finally, recall that problem (5) can be considered as a set of three 1D problems. Thus, let us focus on the 1D case for a while. Details of the more prevalent 3D case are considered later on. Notice that because of this, the symbols will be changed a bit: $r_0, v_0 \in \mathbb{R}$ whereas $\mathbf{r}_0, \mathbf{v}_0 \in \mathbb{R}^3$ and so on. In the following treatment, it is assumed that the accelerometer samples represent an instantaneous value of the specific force. In other words, it is assumed that the sensor output is not processed in any way before the "navigation computer."

3. Solution of a BVP

The main goal of this section is to form a linear system of equations of the form $\mathbf{Ax} = \mathbf{b}$ for the position estimates $\mathbf{x}$. These are now expressed as a vector containing the position at each discrete time instant (referred to as x_i), where the vector $\mathbf{b}$ is a function of $a(t)$. In the following treatment, the total number of the samples is N, $t_i \in [T_0, T_1]$ is the value of time instant $i \in \mathbb{Z}$ $(1 \le i \le N)$, and $h_i = t_{i+1} - t_i$.

Now, let us rephrase problem (5) as a variational equation

$$\int_{T_0}^{T_1} \ddot{r}u \, \mathrm{d}t = \int_{T_0}^{T_1} au \, \mathrm{d}t \quad \forall u \in U, \tag{6}$$

where U is a Hilbert space [7]. Second derivative of r can be eliminated by integrating the left-hand side of (6) by parts, which yields

$$\int_{T_0}^{T_1} \ddot{r}u \, \mathrm{d}t = \Big/_{T_0}^{T_1} \dot{r}u - \int_{T_0}^{T_1} \dot{r}\dot{u} \, \mathrm{d}t = \int_{T_0}^{T_1} au \, \mathrm{d}t \quad \forall u \in U, \tag{7}$$

where the notation "$\Big/$" stands for substitution. By evaluating the substitution term and rearranging (7), we get

$$\int_{T_0}^{T_1} \dot{r}\dot{u} \, \mathrm{d}t = -\int_{T_0}^{T_1} au \, \mathrm{d}t + \dot{r}(T_1)u(\mathrm{T}_1) - \dot{r}(T_0)u(T_0) \quad \forall u \in U. \tag{8}$$

While (8) is otherwise in a convenient form for our purposes, it is not well suited for practical computations, because U is an infinite-dimensional space. Thus, let us approximate U with a finite-dimensional space spanned by

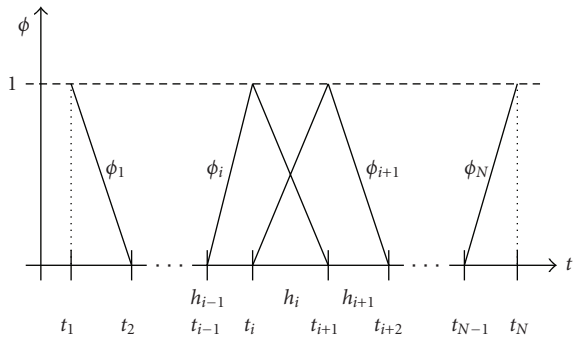

Figure 1: Lowest order basis functions ϕ_i.

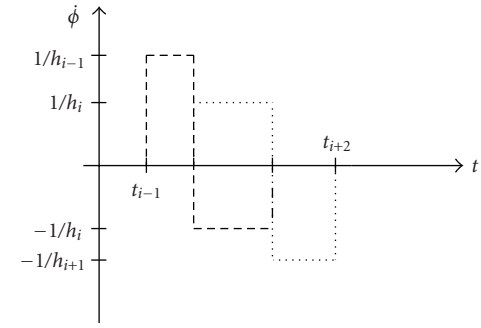

Figure 2: Time derivative of the lowest order basis functions ϕ_i (dashed line) and ϕ_{i+1} (dotted line).

piecewise affine basis functions ("affine function = linear function + a constant")

$$\phi_i = \begin{cases} 0, & t < t_{i-1}, \\ \dfrac{(t - t_{i-1})}{h_{i-1}}, & t_{i-1} \le t < t_i, \\ 1 - \dfrac{(t - t_i)}{h_i}, & t_i \le t < t_{i+1}, \\ 0, & t \ge t_{i+1}, \end{cases} \tag{9}$$

often referred to as the "hat" functions. Functions (9) for few values of i are shown in Figure 1. Also, let us use the same basis functions ϕ to discretize u and r. This choice is often referred to as the Galerkin method [7]. Note that it is also possible to choose basis functions ϕ_i different from the lowest order approximation used here, when considered necessary.

Given the basis functions ϕ_i, notice that the position r can be approximated as a piecewise affine function

$$r \approx \sum_{i=1}^{N} x_i \phi_i. \tag{10}$$

From (8), we see that the derivatives of the basis functions (9) are also needed. It holds that

$$\dot{\phi}_i = \begin{cases} 0, & t < t_{i-1}, \\ \dfrac{1}{h_{i-1}}, & t_{i-1} < t < t_i, \\ -\dfrac{1}{h_i}, & t_i < t < t_{i+1}, \\ 0, & t > t_{i+1}. \end{cases} \tag{11}$$

From Figure 2, one can verify that function $\dot{\phi}_i$ is a piecewise constant function, discontinuous at points t_{i-1}, t_i, and t_{i+1}. From (8) we see that we need to *integrate* a similar term with discontinuities at the nodes over the domain. In other words, these discontinuities do not matter, which is a well-known fact from integral theory.

In total, we are now in the position to discretize equation (8), which yields

$$\sum_{i=1}^{N} x_i \int_{T_0}^{T_1} \dot{\phi}_i \dot{\phi}_j \, \mathrm{d}t = -\int_{T_0}^{T_1} a\phi_j \, \mathrm{d}t + \dot{r}(T_1)\phi_j(T_1) - \dot{r}(T_0)\phi_j(T_0) \quad \forall j = 1, 2, \ldots, N. \tag{12}$$

Now that we have discretized the problem, we are getting closer to the equation $\mathbf{Ax} = \mathbf{b}$ stated as our goal. In fact, (12) is a system of linear equations. Thus, let us start by assembling the matrix $\mathbf{A}$. Knowing that the index i in (12) refers to a certain column and j to a certain row, the elements of $\mathbf{A}$ are given as

$$\mathbf{A}[i, j] = \int_{T_0}^{T_1} \dot{\phi}_i \dot{\phi}_j \, \mathrm{d}t \quad \forall i, j = 1, 2, \ldots, N, \tag{13}$$

which are easy to compute by substitution of (11) into (13). In practice, $\mathbf{A}$ is going to have only few nonzero elements all of them at the diagonal, subdiagonal, and superdiagonal (assuming the "obvious" indexing). For equally spaced nodes with step size h, for example, we have elements $2/h$ at the diagonal and $-1/h$ at the sub-and superdiagonal.

Our next task is to compute the vector $\mathbf{b}$. As seen from (12), the task is to integrate term $a\phi_j$ over the interval $[T_0, T_1]$. To do this, we fit a piecewise affine function to $a(t)$ between every node as seen in Figure 3 with dashed line, which yields

$$\int_{t_{j-1}}^{t_{j+1}} a\phi_j \, \mathrm{d}t = \frac{h_{j-1}}{6}\left[a\left(t_{j-1}\right) + 2a\left(t_j\right)\right] + \frac{h_j}{6}\left[2a\left(t_j\right) + a\left(t_{j+1}\right)\right] \tag{14}$$

for every $j \in [2, N-1]$. For nodes $j = 1$ and $j = N$, we get

$$\int_{t_1}^{t_2} a\phi_j \, \mathrm{d}t = \frac{h_1}{6}[2a(t_0) + a(t_1)], \qquad \int_{t_{N-1}}^{t_N} a\phi_j \, \mathrm{d}t = \frac{h_{N-1}}{6}[a(t_{N-1}) + 2a(t_N)], \tag{15}$$

respectively.

Finally, let us consider how to apply the different types of boundary values into (12). At first, notice that terms $\phi_j(T_1)$ and $\phi_j(T_0)$ seen in (12) will be nonzero (evaluating to value one) only for the values $j = 1$ and $j = N$, respectively. If $\dot{r}(T_0) = v_0$ or $\dot{r}(T_1) = v_1$ of (5) is given, the respective boundary condition is called a *Neumann* boundary condition [7]. These can be applied directly by adding the given values into $\mathbf{b}$.

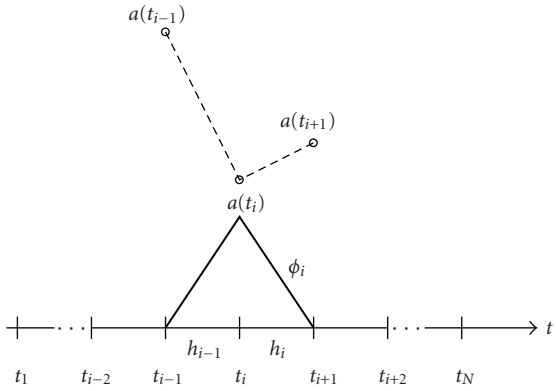

Figure 3: Basis function ϕ_i along with the corresponding acceleration values.

The other possibility for the boundary values is to have $r(T_0) = r_0$ or $r(T_1) = r_1$ fixed thus having a *Dirichlet* boundary condition [7]. As the value at the corresponding boundary is already known, it does not need to be solved. Thus, we can move all terms depending on it to $\mathbf{b}$ reducing the number of unknown terms by one. For the reduced system, the corresponding term of the last two terms of (12) will be zero, as mentioned above. Recall that it is also possible to have a Dirichlet condition on one boundary and a Neumann condition on the other boundary.

We have now means to assembly an equation of the form

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{16}$$

for the (1D) position of the object. Depending on the type of the applied boundary conditions, the number of unknowns n is equal to $N-1$ or $N-2$. Matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$ is known to be symmetric and positive definite [22]. With the chosen basis functions, $\mathbf{A}$ is also a tridiagonal matrix. In practice, these properties guarantee that (16) can be solved with linear time complexity [8]. In other words, when doubling the amount of unknowns N, the time needed to solve the system is also doubled (approximately), which is certainly not true for a general system of linear equations.

From the position data $\mathbf{x}$, it is now a straightforward task to compute the velocity using a suitable numerical differentiation formula. Since the position data is relatively smooth due to the "double integration" process, we have not experienced any problems in computing the derivative with an adequate accuracy.

4. Generalization to 3D

In this section, we will generalize the method described in the previous section to the 3D case. For this, we introduce a coordinate transformation matrix $\mathbf{C}(t)$ used to transform the specific force measurements $\mathbf{f}^{\mathrm{b}}(t)$ into the accelerations $\mathbf{a}^{\mathrm{n}}(t)$ represented in a suitable *navigation* frame as follows:

$$\mathbf{a}^{\mathrm{n}}(t) = \mathbf{C}(t)\mathbf{a}^{\mathrm{b}}(t). \tag{17}$$

An element-wise representation of $\mathbf{C}(t)$ is

$$\mathbf{C}(t) = \begin{bmatrix} c_{11}(t) & c_{12}(t) & c_{13}(t) \\ c_{21}(t) & c_{22}(t) & c_{23}(t) \\ c_{32}(t) & c_{32}(t) & c_{33}(t) \end{bmatrix}. \tag{18}$$

In this paper, we will assume that an estimate of $\mathbf{C}(t)$ for each time instant is available. Note that $\mathbf{C}(t)$ depends only on the attitude, which is independent of the position and velocity, when the assumptions considered in the first section are valid [2]. Furthermore, let vector $\mathbf{g}^n$ be the acceleration due to the gravity represented in the navigation frame. With these notions, the specific force measurements f made by the accelerometers can be "converted" into accelerations as

$$\begin{aligned} a_1^{\mathrm{n}} &= c_{11}f_1^{\mathrm{b}} + c_{12}f_2^{\mathrm{b}} + c_{13}f_3^{\mathrm{b}} + g_1^{\mathrm{n}}, \\ a_2^{\mathrm{n}} &= c_{21}f_1^{\mathrm{b}} + c_{22}f_2^{\mathrm{b}} + c_{23}f_3^{\mathrm{b}} + g_2^{\mathrm{n}}, \\ a_3^{\mathrm{n}} &= c_{31}f_1^{\mathrm{b}} + c_{32}f_2^{\mathrm{b}} + c_{33}f_3^{\mathrm{b}} + g_3^{\mathrm{n}}, \end{aligned} \tag{19}$$

where the time dependencies have not been explicitly stated.

Now, following from (19) and the right-hand side of (12), we have an equation

$$\begin{aligned} \mathbf{b}_i[j] &= -\int_{T_0}^{T_1} a_i^{\mathrm{n}}\phi_j \,\mathrm{d}t \\ &= -\int_{T_0}^{T_1} \left[c_{i1}f_1^{\mathrm{b}} + c_{i2}f_2^{\mathrm{b}} + c_{i3}f_3^{\mathrm{b}} + g_i^{\mathrm{n}}\right]\phi_j \,\mathrm{d}t \end{aligned} \tag{20}$$

for the j th component of vector $\mathbf{b}_i \ \forall\, i = [1,2,3]$. Thus, the linear equation for the 3D position is

$$\begin{bmatrix} \mathbf{A}_1 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_3 \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \mathbf{x}_3 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \\ \mathbf{b}_3 \end{bmatrix}. \tag{21}$$

As discussed in the previous section, matrix $\mathbf{A}_i$ is a tridiagonal matrix. Then, also the system matrix of (21) is not only tridiagonal, but a block diagonal matrix. For instance, if boundary conditions are fixed in the same way for all dimensions, it follows that $\mathbf{A}_1 = \mathbf{A}_2 = \mathbf{A}_3$ holds.

With the equations derived in this section, it is possible to solve the 3D BVP using the presented FEM method. The solution of this more general problem can also be found with linear time complexity, as in the 1D case.

5. Using a Number of Additional Constraints to Model Sensor Errors

So far we have constructed a method to compute position and velocity data with certain boundary conditions. We will now propose a way to exploit a number of additional constraints concerning the velocity or the position information to estimate sensor errors. For this, let us now precisely define the term "additional constraint": *an additional constraint is a linear equation that bounds the position or the velocity of the object at an arbitrary number of time instances* $t_i \in [T_0, T_1]$.

At simplest, the previous definition could simply mean that $v_1(t_i)$ is fixed. On the other hand, a less intuitive equation $\sum_{i=1}^{N}[r_1(t_i) + r_2(t_i)] = z_1$ is also a valid constraint. This way, we can for example exploit constraints like $x_i = x_j$ for any i and j without saying anything about the absolute position at these points. It could be hard to exploit these kind of constraints properly in traditional filtering problems.

In this section, we will use constraints systematically to enhance accuracy. To make the concepts presented in this section clear, let us first present the equations for the 1D case.

5.1. Treating Additional Constraints. Let us first assume that $\mathbf{A} \in \mathbb{R}^{N\times N}$ holds. That is, given a Dirichlet boundary condition, we add an auxiliary equation to the system instead of removing the corresponding boundary value from the system. This is a necessary procedure when exploiting additional constraints.

In general, all linear constraints can be represented in the form

$$\mathbf{D}\mathbf{x} = \mathbf{e}, \tag{22}$$

where $\mathbf{D} \in \mathbb{R}^{p\times N}$ and $\mathbf{e} \in \mathbb{R}^{p}$ hold, where p is the number of the constraint equations. For a practical example closely related to the example 1, consider that (5) is solved with Dirichlet boundary values. If the Neumann boundary values are also known and the samples are equally spaced, one can construct (22) as

$$\mathbf{D} = \begin{bmatrix} -\frac{3}{2h} & \frac{2}{h} & -\frac{1}{2h} & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & \frac{1}{2h} & -\frac{2}{h} & \frac{3}{2h} \end{bmatrix} \tag{23}$$

$$\mathbf{e} = \begin{bmatrix} v_0 \\ v_1 \end{bmatrix}. \tag{24}$$

Matrix $\mathbf{D}$ is formed using a three-point differentiation method based on polynomial interpolation [10] to determine Neumann boundary values from the displacement data.

These constraints could be exploited just by computing the linear least squares problem constructed by adding these additional equations to (16). Usually, however, it is favorable to use the additional constraints to *model sensor errors*, at least when one has any knowledge of the types of the errors included in the measurements. For more information on the assumptions related to the sensor error modeling, recall Section 2.1. In this text, we will present a linear sensor error model.

5.2. Sensor Error Model. In typical situations, bias offset (i.e., the sensor shows nonzero output when no forces are acting upon it) and scaling factor are the two terms which have to beknown very accurately in order to have any realistic position or velocity estimate as a solution. Because of the run-to-run variations of these errors, they cannot be assumed to be constant between two separate events. Thus, they should be treated as unknowns. Other typical errors are caused by misalignment of the axes, changing temperature and drifting bias.

As a practical example used in the example 1 later in Section 6, let us model the sensor errors as follows:

$$\hat{f}(t) = sf(t) + b, \tag{25}$$

where $\hat{f}(t)$ is the corrected specific force measurement, s is some constant scaling factor, and b is some constant bias term correcting the erroneous measurement. By replacing the measured $f(t)$ treated in the previous section with the corrected acceleration $\hat{f}(t)$, it is easy to form an equation of the form

$$\mathbf{b} = \mathbf{F}\mathbf{l} \tag{26}$$

for vector $\mathbf{b}$ seen in (16). Matrix $\mathbf{F}$ is in this case an $N \times 2$ matrix, whose elements are formed by computing the integrals $-\int_{T_0}^{T_1} a\phi_j$ and $-\int_{T_0}^{T_1} \phi_j$ from (12). Vector $\mathbf{l}$ is simply $[\, s \; b \,]^T$.

Let us now replace the right-hand side of (16) with

$$\mathbf{b}_0 + \mathbf{b} = \mathbf{b}_0 + \mathbf{F}\mathbf{l}, \tag{27}$$

where $\mathbf{b}_0$ depends only on the given boundary conditions (which are treated as exact) and corresponding boundary values. This distinction is necessary, since one should not modify the given boundary values by applying error modeling on them.

In general, a problem of linear constraints used to model linear sensor errors can be stated as

$$\begin{cases} \mathbf{A}\mathbf{x} = \mathbf{b}_0 + \mathbf{F}\mathbf{l}, \\ \mathbf{D}\mathbf{x} = \mathbf{e}, \end{cases} \tag{28}$$

where $\mathbf{F} \in \mathbb{R}^{N\times q}$, $\mathbf{l} \in \mathbb{R}^{q}$ hold, and q is the number of modeled sensor error terms.

Now, since $\mathbf{A}$ is known to be invertible, one gets an equation

$$\mathbf{D}\mathbf{A}^{-1}\mathbf{F}\mathbf{l} = \mathbf{e} - \mathbf{D}\mathbf{A}^{-1}\mathbf{b}_0 \tag{29}$$

for $\mathbf{l}$. Note that the matrix $\mathbf{D}\mathbf{A}^{-1}\mathbf{F}$ has dimension $p \times q$, that is, it is typically a very small matrix compared to $\mathbf{A}$. With $\mathbf{l}$ known, it is easy to solve for $\mathbf{x}$.

In the case where $p > q$ (more constraints than model parameters), one can also take reliability of different measurements into account by solving a *weighted least squares problem* (WLS) [22]. In general, (29) has a unique solution $\mathbf{l}$ only when $p = q$ holds and the row rank of $\mathbf{D}$ is full. Typically, making sure that $p \geq q$ and that the row rank of $\mathbf{D}$ is at least q, (29) has either a unique or a least squares solution for $\mathbf{l}$. In each case, the upper equation of (28) is treated as an exact equation.

5.3. Additional Constraints in 3D. Let us now consider the use of additional constraints in a 3D case. Using the

notions form the previous section and (22) as an example, constraints can be represented as

$$\mathbf{D}\begin{bmatrix}\mathbf{x}_1\\ \mathbf{x}_2\\ \mathbf{x}_3\end{bmatrix} = \mathbf{e}. \tag{30}$$

In addition to the sensor errors discussed in the 1D example, it is now also possible to model errors like attitude errors, unknown value of the acceleration due to the gravity, and cross-correlation of different sensors. A particularly useful method is to treat the acceleration due to the gravity as an unknown three-dimensional vector, which is then subtracted from the specific force measurements given in the global coordinates. This reduces the systems sensitivity to initial attitude errors.

As an example, let us now derive the equations for sensor error of (25) for each axis in addition to the unknown acceleration due to the gravity discussed above. Thus, we have a total of 9 unknown model parameters. As one could expect, we must take the effects of the coordinate transformation matrix into account when deriving the necessary equations. By plugging (25) into (21), we have

$$\begin{aligned}\mathbf{b}_i[j] = &-\int_{T_0}^{T_1}\Big[c_{i1}\left(s_1 f_1^{\mathrm{b}} + b_1\right) + c_{i2}\left(s_2 f_2^{\mathrm{b}} + b_2\right)\\ &- c_{i3}\left(s_3 f_3^{\mathrm{b}} + b_3\right) + g_i^{\mathrm{n}}\Big]\phi_j\,\mathrm{d}t\\ = &-s_1\int_{T_0}^{T_1} c_{i1} f_1^{\mathrm{b}}\phi_j\,\mathrm{d}t - b_1\int_{T_0}^{T_1} c_{i1}\phi_j\,\mathrm{d}t\\ &- s_2\int_{T_0}^{T_1} c_{i2} f_2^{\mathrm{b}}\phi_j\,\mathrm{d}t - b_2\int_{T_0}^{T_1} c_{i2}\phi_j\,\mathrm{d}t\\ &- s_3\int_{T_0}^{T_1} c_{i3} f_3^{\mathrm{b}}\phi_j\,\mathrm{d}t - b_3\int_{T_0}^{T_1} c_{i3}\phi_j\,\mathrm{d}t\\ &- g_i^{\mathrm{n}}\int_{T_0}^{T_1}\phi_j\,\mathrm{d}t.\end{aligned} \tag{31}$$

The several integrals in (31) are scalars for each $j \in [1, 2, \ldots N]$, easily evaluated with (14) and (15), since no unknown terms appear inside the integrals. Let us now gather the results into vectors

$$\mathbf{p}_{klmn}[j] = -\int_{T_0}^{T_1} c_{kl}^{m}\left(f_l^{\mathrm{b}}\right)^{n}\phi_j\,\mathrm{d}t \tag{32}$$

for all $j \in [1, 2, \ldots, N]$. With these notions, we get, for example

$$\begin{aligned}\mathbf{p}_{kl11}[j] &= -\int_{T_0}^{T_1} c_{kl} f_l^{\mathrm{b}}\phi_j\,\mathrm{d}t,\\ \mathbf{p}_{kl1}[j] &= -\int_{T_0}^{T_1} c_{kl}\phi_j\,\mathrm{d}t,\\ \mathbf{p}[j] &= -\int_{T_0}^{T_1}\phi_j\,\mathrm{d}t,\end{aligned} \tag{33}$$

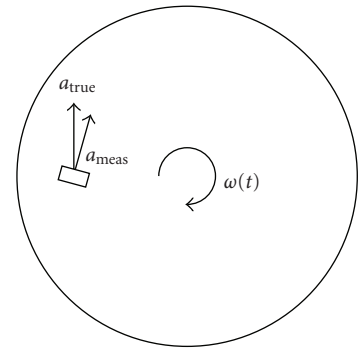

Figure 4: Measurement setup of example 1.

where only the relevant indices are shown. Thus, we get size

$$\mathbf{F} = \begin{bmatrix}\mathbf{p}_{1111} & \mathbf{p}_{111} & \mathbf{p}_{1211} & \mathbf{p}_{121} & \mathbf{p}_{1311} & \mathbf{p}_{131} & \mathbf{p} & \mathbf{0} & \mathbf{0}\\ \mathbf{p}_{2111} & \mathbf{p}_{211} & \mathbf{p}_{2211} & \mathbf{p}_{221} & \mathbf{p}_{2311} & \mathbf{p}_{231} & \mathbf{0} & \mathbf{p} & \mathbf{0}\\ \mathbf{p}_{3111} & \mathbf{p}_{311} & \mathbf{p}_{3211} & \mathbf{p}_{321} & \mathbf{p}_{3311} & \mathbf{p}_{331} & \mathbf{0} & \mathbf{0} & \mathbf{p}\end{bmatrix}, \tag{34}$$

$$\mathbf{l} = \begin{bmatrix}s_1 & b_1 & s_2 & b_2 & s_3 & b_3 & g_1^{\mathrm{n}} & g_2^{\mathrm{n}} & g_3^{\mathrm{n}}\end{bmatrix}^{\mathrm{T}}. \tag{35}$$

For the nine unknown model parameters we need at least nine constraints. This can be covered, for example, with the knowledge of the velocity of the object at three different points. In total, we have equations identical to (28), with dimension N replaced by $3N$.

6. Example 1

To demonstrate the use of the proposed method, an example using readily available consumer grade accelerometers [23] was created. An accelerometer was mounted on a horizontally rotating rate table, whose angular velocity (rotation rate) can be controlled. The accelerometer was mounted in such a way that its measurement direction was *approximately* the same as the direction of the tangential acceleration caused by angular acceleration of the rate table, as seen in Figure 4.

The aim of this example was not to get position and velocity as accurately as possible, but to compare different solution methods in a situation where one needs to get reasonable position and velocity estimates regardless of the fact that the measurement contains significant errors. The angular velocity of the rate table was set to follow function illustrated in the Figure 5 a certain number of times. With this kind of a setup, the true acceleration, velocity, and displacement of the mounted sensor were known up to the accuracy of the motor rotating the rate table. The errors caused by the rate table were observed using the angular rate output of the motor and found to be negligible compared to other error sources.

In first test, the rate table was set to repeat exactly the function represented in Figure 5 ten times, leading to seven full revolutions (or 10.4 meters) in $T = 20$ seconds. In the other test, the function of the same form was scaled in such a way that 50 repeats lead to total of 62 full revolutions (or 92.3 meters) in $T = 100$ seconds. The sampling frequency of

Table 1: Comparison of different methods computing object's velocity and position. "I" stands for time-stepping, "II" for FEM, and "III" for FEM with sensor error modeling.

	Test 1			Test 2		
Vel. error [m/s]	t_1	t_2	t_3	t_1	t_2	t_3
I	0.07	0.10	0.12	33.24	65.47	97.17
II	0.01	0.04	0.06	−0.67	0.50	1.15
III	0.04	0.05	0.04	0.33	0.54	0.44
Pos. error [m]	t_1	t_2	t_3	t_1	t_2	t_3
I	0.10	0.49	1.00	510.96	1992.99	4434.70
II	−0.31	−0.29	−0.15	−42.34	−43.42	−16.59
III	−0.10	−0.07	−0.05	−6.78	−4.72	−0.61

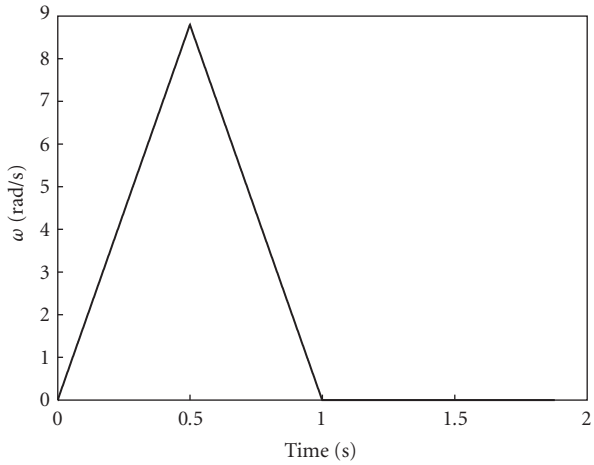

Figure 5: Angular velocity of the rate table.

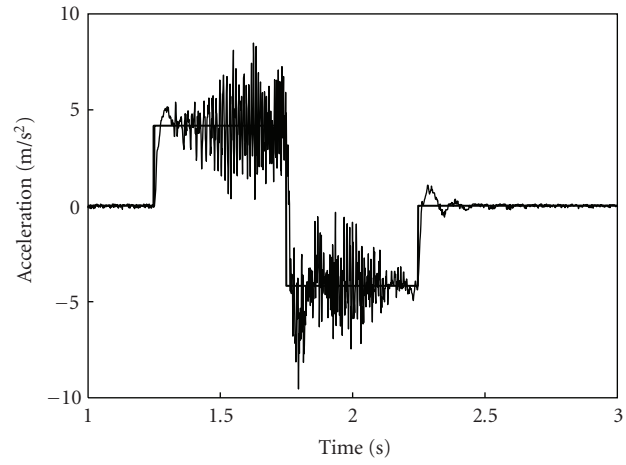

Figure 6: Ideal (thick line) and measured acceleration (thin line).

the sensor was set to 1000 Hz and an accurate (16 bit) analog to digital converter was used. Accelerations were measured using a consumer grade 12 g accelerometer [23].

The raw accelerometer data was mapped to accelerations with the scale factor given by the manufacturer. From this acceleration data, position and velocity were computed by a number of different methods:

(i) "Traditional" IVP (double integration with trapezoid rule),

(ii) FEM with Dirichlet boundary conditions (BCs),

(iii) FEM with Dirichlet BCs combined with additional Neumann BCs to supply the error model (25).

In methods I and II, the bias error of the sensor was estimated by averaging the output while the sensor was at rest. This was done in order to make method I (and to some extent, method II) comparable to the method III by reduction of the large bias error. This is rarely possible in general and only method III can be used to reliably detect any remaining bias (example 2 in Section 7 is an example of this case). As a reference, the ideal (ideal driving motor) and the measured accelerations (accelerometer bias removed) of the first spin are plotted in Figure 6.

Table 1 shows the main results of the tests. In each test, the computed results were compared to the known value in three separate points by computing the difference between the computed and the real value. Point t_1 was located at $T/3$, t_2 at $T/2$, and t_3 at $2T/3$.

As seen in Table 1, method I seems to increase the error with increasing time, as expected. Method II on the contrary, thanks to the basic property of the variational technique, does not increase the error but distributes it over the whole time period. The difference between these two methods is clearly seen in the velocity and position errors of the longer test (test 2), where method II gives much better estimates than method I. In each test, method III clearly outperforms methods I and II, which is expected due to the provided two additional constraints.

Figures 7 and 8 demonstrate the differences between velocity and position estimates given by methods I and II during test 1. Estimates given by method III coincide with the reference plots. Figure 7 shows only the last spin of the test 1 for better view of the differences.

7. Example 2

This example considers the computation of the velocity and position of a ski jumper during a single jump. As compared to the previous example, this is a more realistic and general inertial navigation problem with six degrees of freedom.

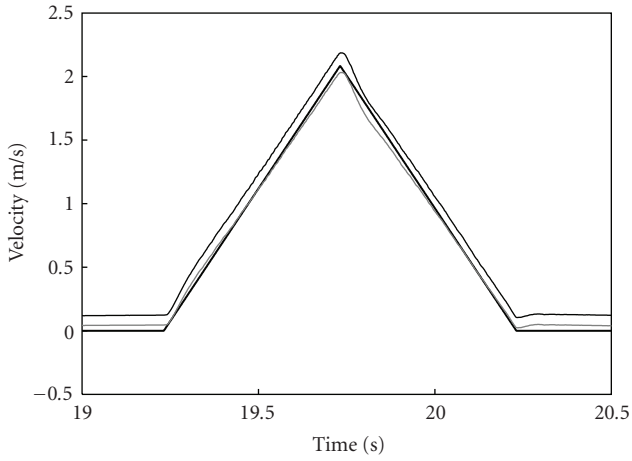

Figure 7: Velocity estimates given by methods I (thin black line) and II (thin gray line) compared to the ideal velocity (thick line).

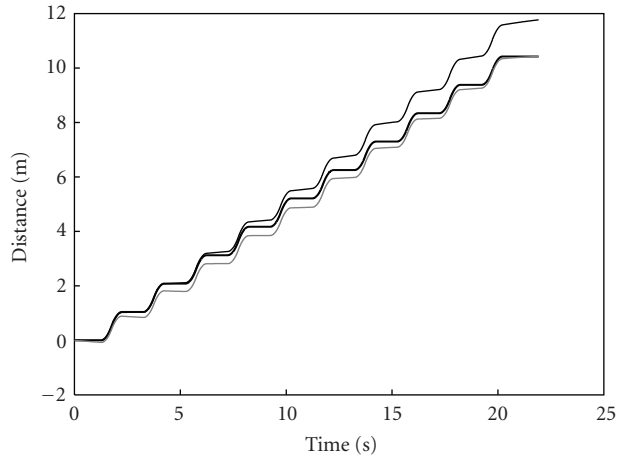

Figure 8: Position estimates given by methods I (thin black line) and II (thin gray line) compared to the ideal position (thick solid line).

Computation of the attitude of the object is based on the data given by three standard consumer grade gyroscopes [24]. The position and velocity were then measured with three standard consumer grade accelerometers [23] and computed with the proposed method with a sensor error model similar to the one presented in Section 5.3. The needed additional constraints contained information about

(i) the location of the jumper at five points evenly spread on the inrun hill (in Figure 9, the part of thick black line with negative x- and positive y-coordinates)

(ii) the trajectory of the jumper after the landing, which should coincide with the linearization of the landing hill (in Figure 9, the part of thick black line with x-coordinates greater than 100 m).

In Figure 9, two-dimensional trajectories of two jumpers are plotted along with the known profile of the hill. At first, the trajectories follow the profile of the hill until the jumpers

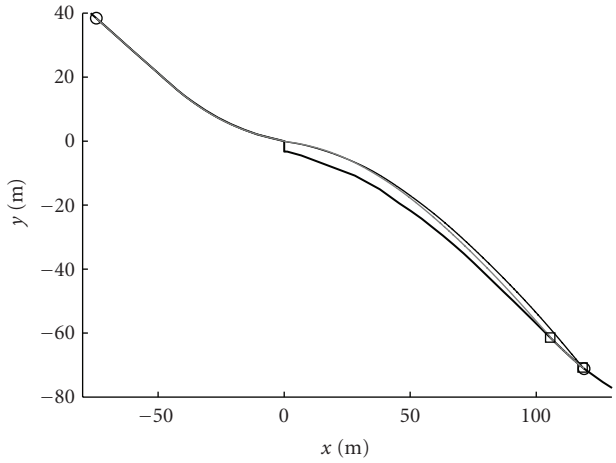

Figure 9: Computed two-dimensional trajectories of two independent events (thin gray and thin black lines) along with the known profile of the hill (thick black line).

take off and eventually land at some point of the landing hill. Drawn circles at both ends of the trajectories demonstrate the start and the end of the navigation period and the given Dirichlet boundary conditions. Drawn squares represent the landing points, in this case quite different for the two events, agreeing well with the recorded jump lengths. Notice that the trajectories are computed using two distinct measurement systems, each containing unique error sources.

Figure 10 shows that the vertical component of the velocity of the same two jumps is plotted as a function of horizontal displacement. Notice how the absolute values of the vertical velocity substantially differ while the small changes in the velocity are practically identical. One might first consider the small changes as typical stochastic errors caused by the inertial navigation system, especially when dealing with low performance sensors.

Given that two independent measurement systems show the same variations in the velocity at the same locations, it is evident that there is actually only a negligible amount of stochastic errors present. Instead, the small variations are caused by deterministic sources, namely, in this particular application the uneven inrun hill.

Unfortunately, the estimates cannot be compared with a reference trajectory, because such data are not available. Thus, we cannot give the exact amount of error present in the position and velocity estimates. We do however claim that the achieved accuracy is something one does not typically expect from consumer grade inertial sensors.

8. Conclusion

The work was motivated by applications, where it is natural to encounter BVPs instead of IVPs. In many cases, it is also possible to formulate an IVP as a BVP, given that the results are not required in real time.

Finite element method is utilized to solve inertial navigation problems formulated as BVPs. As a result, we get a linear system of equations for the position estimates, whose

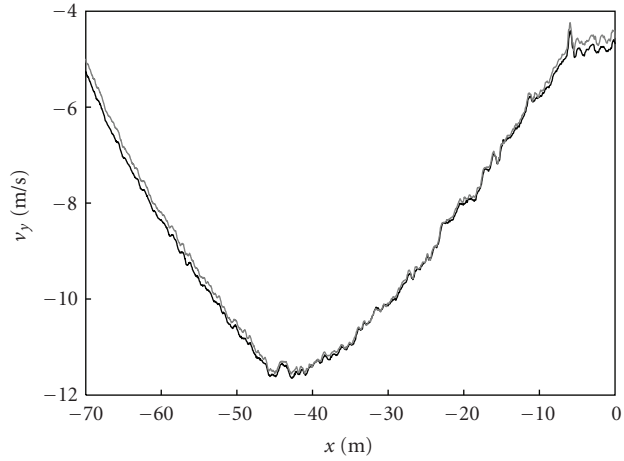

FIGURE 10: Vertical velocity (v_y) as a function of horizontal displacement (x) of the two independent measurements.

solution can be found with linear time complexity. It is demonstrated that solving a BVP rather than an "equivalent" IVP gives more accurate results.

For further accuracy enhancements, an efficient way of combining inertial measurements with possible additional constraints is created. This gives us a possibility to model constant sensor errors, known to limit the achievable accuracy of the system. While the error model significantly enhances the accuracy of the system, it is kept computationally simple and easily adoptable.

In practice, the accuracy improvements allow us to exploit inertial sensors of certain performance level in more challenging applications. For this, it is necessary to see that the concept of inertial navigation does not invariably imply an IVP, but a BVP as well. Then, the use of FEM will provide an efficient way to compute position and velocity estimates not prone to the accumulation of errors.

In larger scale, the current paper serves as an introduction to the idea of formulating inertial navigation problems as BVPs. As a consequence, further studies are needed to address problems to which the presented tools do not provide an obvious solution. These include, for example, stochastic errors, reliability of the possible additional constraints (as compared to the accuracy of the IMU), and coupling of position and attitude errors.

References

[1] I. Y. Bar-Itzhack, "Navigation computation in terrestrial strapdown inertial navigation systems," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 13, no. 6, pp. 679–689, 1977.

[2] D. H. Titterton and J. L. Weston, *Strapdown Inertial Navigation Technology*, Institution of Engineering and Technology, London, UK, 2nd edition, 2004.

[3] A. B. Chatfield, *Fundamentals of High Accuracy Inertial Navigation*, American Institute of Aeronautics and Astronautics, Reston, Va, USA, 1997.

[4] M. S. Grewal, L. R. Weill, and A. P. Andrews, *Global Positioning Systems, Inertial Navigation, and Integration*, John Wiley & Sons, New York, NY, USA, 2001.

[5] J. A. Farrell and M. Barth, *The Global Positioning System & Inertial Navigation*, McGraw–Hill, New York, NY, USA, 1999.

[6] L. F. Shampine, I. Gladwell, and S. Thompson, *Solving ODEs with Matlab*, Cambridge University Press, Cambridge, UK, 2003.

[7] S. Larsson and V. Thome, *Partial Differential Equations with Numerical Methods*, Springer, New York, NY, USA, 2003.

[8] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C (The Art of Scientific Computing)*, Cambridge University Press, Cambridge, UK, 2nd edition, 1992.

[9] M. Virmavirta, T. Nieminen, T. Tarhasaari, and L. Kettunen, "High precision inertial measurement for tracking the trajectories of ski jumpers "Smart Boot Project"," in *Proceedings of the 13th Congress European College of Sport Science*, p. 572, Estoril, Portugal, July 2008.

[10] D. Kincaid and W. Cheney, *Numerical Analysis: Mathematics of Scientific Computing*, BROOKS/COLE, Florence, Ky, USA, 3rd edition, 2002.

[11] N. El-Sheimy, H. Haiying, and N. Xiaoji, "Analysis and modeling of inertial sensors using allan variance," *EEE Transactions on Instrumentation and Measurement*, vol. 57, no. 1, pp. 140–149, 2008.

[12] A. Gelb, *Applied Optimal Estimation*, MIT Press, Cambridge, Mass, USA, 1974.

[13] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*, Chapman & Hall / CRC, London, UK, 2004.

[14] J. L. Crassidis and J. L. Junkins, *Optimal Estimation of Dynamic Systems*, Chapman & Hall / CRC, London, UK, 2004.

[15] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, Artech House, Norwood, Mass, USA, 2008.

[16] D. Fraser and J. Potter, "The optimum linear smoother as combination of two optimum linear filters," *IEEE Transactions on Automatic Control*, vol. 14, no. 4, pp. 387–390, 1969.

[17] H. E. Rauch, F. Tung, and C. T. Striebel, "Maximum likelihood estimates of linear dynamic systems," *AIAA Journal*, vol. 3, no. 8, pp. 1445–1450, 1965.

[18] K. Gade, "Navlab, a generic simulation and postprocessing tool for navigation," *European Journal of Navigation*, vol. 2, no. 4, pp. 21–59, 2004.

[19] Y. Yang, Z. Jin, W. Tian, and F. Qian, "Application of fixed interval smoothing to gps/dr integrated navigation system," in *Proceedings of the IEEE Intelligent Transportation Systems*, vol. 2, pp. 1027–1031, October 2003.

[20] A. B. Willumsen and ∅. Hegrenæs, "The joys of smoothing," in *Proceedings of the IEEE Bremen: Balancing Technology with Future Needs (OCEANS '09)*, pp. 1–7, Bremen, Germany, May 2009.

[21] M. Klaas, M. Briers, N. de Freitas, A. Doucet, S. Maskell, and D. Lang, "Fast particle smoothing: if i had a million particles," in *Proceedings of the 23rd International Conference on Machine Learning (ICML '06)*, vol. 148, pp. 481–488, Pittsburgh, Pa, USA, 2006.

[22] J. W. Demmel, *Applied Numerical Linear Algebra*, SIAM, Philadelphia, Pa, USA, 1st edition, 1997.

[23] VTI. *VTI SCA620-CHCV1A Datasheet*, 2006. Revision 2/2, Checked on 22.12.2009.

[24] Analog Devices. *AD ADXRS300ABG Datasheet*, 2004. Revision B, Checked on 22.12.2009.

Publication 3

Tuukka Nieminen, Jari Kangas and Lauri Kettunen
"Use of Tikhonov Regularization to Improve the Accuracy of Position Estimates in Inertial Navigation"
International Journal of Navigation and Observation (10pp), 2011
doi:10.1155/2011/450269

Hindawi Publishing Corporation
International Journal of Navigation and Observation
Volume 2011, Article ID 450269, 10 pages
doi:10.1155/2011/450269

Research Article

Use of Tikhonov Regularization to Improve the Accuracy of Position Estimates in Inertial Navigation

Tuukka Nieminen, Jari Kangas, and Lauri Kettunen

Institute of Electromagnetics, Tampere University of Technology, P.O. Box 692, 33101 Tampere, Finland

Correspondence should be addressed to Tuukka Nieminen, tuukka.nieminen@tut.fi

Received 26 November 2010; Accepted 12 February 2011

Academic Editor: Abbas Mohammed

Inertial navigation problems are often understood as *initial value problems*. However, there are many applications where *boundary value problems* naturally arise. In these situations, it has been shown that the *finite element method* can be efficiently used to compute accurate position and velocity estimates. We will propose that finite element method complemented with *Tikhonov regularization*—a basic tool for inverse problems—is a powerful combination for further accuracy improvements. The proposed method provides a straightforward way to exploit prior information of various types and is subject to rigorous optimality results. Use and accuracy of the proposed method are demonstrated with examples.

1. Introduction

The term *inertial navigation* is often associated with the term *initial value problem* (IVP). This kind of an association is justified in many cases, but not always: there are situations, where an inertial navigation problem may just as naturally be posed as a *boundary value problem* (BVP) [1]. In case of a BVP, all measurements within the interesting time interval are exploited, whereas an IVP only exploits measurements up to a certain time within the time interval.

We will discuss BVPs concerning the position of the object, given the accelerations in the navigation frame. Instead of the "full equations", where rotation of Earth is taken into account, we consider "simplified equations", where this is neglected [1]. In this case, to obtain the position ($\mathbf{p}$) of the object, we need to solve the following problem: given $\mathbf{a}(t) : \mathbb{R} \to \mathbb{R}^3$, find $\mathbf{p}(t) : \mathbb{R} \to \mathbb{R}^3$ such that $\mathbf{r}$ satisfies

$$\frac{d^2\mathbf{p}(t)}{dt^2} = \mathbf{a}(t). \tag{1}$$

The problem of resolving accelerations $\mathbf{a}(t)$ given the specific force and angular rate measurements is a different matter, not considered herein. What makes (1) a BVP is that the necessary constraints are not given at same time [2]. More specifically, we will consider two-point BVPs where the necessary constraints are given at the beginning and at the end of the interesting time interval [3].

In [1], a custom-made FEM model is derived to treat two-point BVPs of the form (1) with various choices of boundary conditions. The resulting system of linear equations is shown in [1] to be of the form

$$\begin{bmatrix} \mathbf{A}_x & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_y & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{A}_z \end{bmatrix} \begin{bmatrix} \mathbf{r}_x \\ \mathbf{r}_y \\ \mathbf{r}_z \end{bmatrix} = \begin{bmatrix} \mathbf{b}_x \\ \mathbf{b}_y \\ \mathbf{b}_z \end{bmatrix}. \tag{2}$$

In (2), $\mathbf{r}_x, \mathbf{r}_y, \mathbf{r}_z \in \mathbb{R}^N$, and $\mathbf{r}_x$, for example, is a vector containing the x-coordinate of the object at N different time instances. Vectors $\mathbf{b}_x$, $\mathbf{b}_y$, and $\mathbf{b}_z$ are determined by $\mathbf{a}(t)$. Submatrices $\mathbf{A}_x$, $\mathbf{A}_y$, and $\mathbf{A}_z$ are symmetric, positive definite and with the basis functions considered in [1], tridiagonal $N \times N$ matrices. In the following, a shorthand notation for (2) is used. It is written in the form

$$\mathbf{A}\mathbf{r} = \mathbf{b} + \boldsymbol{\epsilon}, \tag{3}$$

where vector $\boldsymbol{\epsilon} \in \mathbb{R}^{3N}$ is added to emphasize the presence of *measurement errors*, generally of unknown properties. Based on the above discussion, the matrix $\mathbf{A} \in \mathbb{R}^{3N\times 3N}$ is known to be nonsingular. For a 2D problem, the dimensions of

system (3) are $2N$ and, respectively, for 1D, N. Unless otherwise noted, the following discussion assumes dimensions to be $3N$.

The goal of this paper is to estimate $\mathbf{r}$ as accurately as possible, given $\mathbf{A}$, $\mathbf{b}$, and *additional information* of some form. Sometimes, also some properties of $\boldsymbol{\epsilon}$ can be known but in general, we do not make such an assumption. We will be using the term "additional information" throughout this paper, so let us now clarify this concept: "additional information" refers to all available information that can be represented in the linear form

$$\mathbf{D}\mathbf{r} = \mathbf{e} + \boldsymbol{\theta}, \tag{4}$$

where $\mathbf{D} \in \mathbb{R}^{M\times 3N}$ and $\mathbf{e} \in \mathbb{R}^M$. In (4), the term $\boldsymbol{\theta}$ denotes uncertainty, just like $\boldsymbol{\epsilon}$ in (3). The integer M is independent of N and indicates how many individual equations the additional information contains. Notice that properties of $\mathbf{D}$ are generally unknown.

In [1], problems of the form (3) are discussed. There, it is assumed that $\boldsymbol{\epsilon}$ contains mainly deterministic error components, caused by deficient sensors. Then, one can exploit provided additional information of type (4) by modeling errors of these deficient sensors, which results in smaller position and velocity errors via reduction of $\|\boldsymbol{\epsilon}\|_2$ (where index 2 refers to the 2-norm). While this approach was shown to work well in many cases, it is adequate only for situations where the types of the most significant sensor errors are known. In practice, the approach proposed in [1] also limits the attainable accuracy in situations where M is significantly larger than the number of parameters in the chosen error model. The motivation of this paper is to present a new technique to exploit additional information, without the limitations of the technique proposed in [1].

The technique proposed herein is based on *Tikhonov regularization* [4–6], independently developed also by Phillips [7, 8]. As the name suggests, the basic idea of the method is to *regularize* the solution $\mathbf{r}$ of (3). This is required in order to reduce the effects of the error term $\boldsymbol{\epsilon}$. Tikhonov regularization has been studied extensively during the last few decades, mainly in the field of *inverse problems* [8]. Thus, the properties of the method are well known. In the statistical literature, Tikhonov regularization is known as *ridge regression* [8–10].

We apply a tool of inverse problem theory to inertial navigation, because *inertial navigation problems are, in fact, inverse problems*. That is, we are basically resolving the time-parametrized trajectory of an object based on erroneous observations of specific force and angular rate. This reasoning is supported also by the fact that inertial navigation problems are particularly sensitive to measurement errors, which is the hallmark of inverse problems.

This paper is organized as follows: first, we will discuss the background of this study in Section 1.1 and make some remarks about this study in Section 1.2. In Section 2, Tikhonov regularization is presented. Some practical aspects of the proposed method are discussed in Section 3. Sections 4 and 5 are devoted to numerical examples and the conclusions are presented in Section 6.

1.1. Background. Problems resembling the ones considered here have previously been resolved using the means of *fixed interval smoothing* [11–14]. The most frequently used methods of solving these problems are *two-filter smoother* [15] and *Rauch-Tung-Striebel smoother* [16]. These are exploited, for example, in [14, 17–19]. In both cases, differential equation (1) is realized as an IVP. The basic idea of these methods is to run the filter in the forward direction as a "predictor" in phase one and then to run the filter in the backward direction as a "corrector" in phase two. In this approach, each additional constraint is assumed to be associated to a single time instance.

The proposed BVP formulation does not make a distinction between forward and backward directions. Instead of advancing one step at a time, the BVP formulation considers the whole time period, and the solution is obtained in one phase. In addition to the different realization of (1), the key difference is the form of (4), which allows additional information to involve an arbitrary number of time instances. This makes it possible and straightforward to exploit wide-ranging types of additional information. Another particularly useful feature of the proposed method is its capability of finding a reasonable balance between (3) and (4) without knowledge of $\|\boldsymbol{\epsilon}\|_2$ and $\|\boldsymbol{\theta}\|_2$. This property is discussed in Section 2.3.

The similarity between fixed interval smoothing and the BVP approach is that they can exploit information with varying reliability and have similar optimality results. Some optimality results of the proposed method are presented in Section 2.2. Although there are situations, where both methods could be exploited, we will not compare these methods quantitatively.

1.2. Basic Assumptions and Notation. We assume that some additional information is always available, that is, $M > 0$. If this is not the case, the presented method will reduce back to the one considered already in [1]. We will insist on additional information given in the form (4). A more general setting can be found, for example, in [20].

From time to time, we will need to print the entries of matrices and vectors. They are denoted $A_{i,j}$ (for matrix $\mathbf{A}$) and b_j (for vector $\mathbf{b}$), where i and j are positive integers within the allowed range. Symbols i and j are reserved only for this use. Similarly, notation t_i refers to the ith time instance, where the time starts at t_1 and ends at t_N. Step size is defined as $h_i = t_{i+1} - t_i$, where $1 \le i \le N-1$. Without loosing generality, for simplicity we treat the time step as if it was a constant. Finally, a normally distributed random variable x with expectation (or mean) value $E(x)$ and variance σ^2 is denoted as $x \sim \mathcal{N}(E(x), \sigma^2)$.

2. An Introduction to Tikhonov Regularization

Tikhonov regularized solution of (3) and (4) is the solution of

$$\mathbf{r}_\lambda = \underset{r}{\operatorname{argmin}} \|\mathbf{A}\mathbf{r} - \mathbf{b}\|_2^2 + \lambda \|\mathbf{D}\mathbf{r} - \mathbf{e}\|_2^2, \tag{5}$$

where $\lambda > 0$ is called the *regularization parameter*. It is used to weight (3) with respect to (4). Value $\lambda = 1$, for example,

indicates that both equations are weighted equally. With $\lambda \to 0$, $\|\mathbf{A}\mathbf{r}_\lambda - \mathbf{b}\| \to 0$ and the regularized solution $\mathbf{r}_\lambda$ approaches the solution of (3) (with $\boldsymbol{\epsilon} = \mathbf{0}$). Respectively, with $\lambda \to \infty$, the regularized solution $\mathbf{r}_\lambda$ approaches a solution satisfying (4).

Let us point out that in the literature, Tikhonov regularization is seldom presented in the form (5). Most often, it is assumed that the dimensions of $\mathbf{A}$ and $\mathbf{D}$ are equal. Moreover, in many formulations, $\mathbf{D}$ is set to be a difference operator or the identity matrix ($\mathbf{I}$) and/or $\mathbf{e} = \mathbf{0}$ [8, 21, 22].

2.1. Existence and Uniqueness of the Solution. The questions regarding problem (5) are

(1) is there a solution to (5)?

(2) is the solution unique?

(3) how can one find the solution?

Answer to question one follows easily from the fact that the cost function of (5) is nonnegative for every $\lambda > 0$, indicating that the minimization problem has at least one solution. To answer questions two and three, let us denote the value of the cost function by

$$Q_\lambda(\mathbf{r}) = \|\mathbf{A}\mathbf{r} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{D}\mathbf{r} - \mathbf{e}\|_2^2. \tag{6}$$

As known, function $Q_\lambda(\mathbf{r})$ has a local extremum in a critical point of (6), in other words, at a point where the Gâteaux derivative

$$\frac{\mathrm{d}}{\mathrm{d}u} Q_\lambda(\mathbf{r} + u\mathbf{w})\Big|_{u=0} = 0 \quad \forall \mathbf{w} \neq \mathbf{0}, \tag{7}$$

where $u \in \mathbb{R}$ and $\mathbf{w} \in \mathbb{R}^{3N}$. To find critical point(s) of (6), let us substitute (6) into (7). For the first term of (6), we get

$$\begin{aligned} &\frac{\mathrm{d}}{\mathrm{d}u}\|\mathbf{A}(\mathbf{r} + u\mathbf{w}) - \mathbf{b}\|_2^2\Big|_{u=0} \\ &\quad = \frac{\mathrm{d}}{\mathrm{d}u}\langle \mathbf{A}\mathbf{r} + u\mathbf{A}\mathbf{w} - \mathbf{b}, \mathbf{A}\mathbf{r} + u\mathbf{A}\mathbf{w} - \mathbf{b}\rangle\Big|_{u=0} \\ &\quad = 2\langle \mathbf{A}\mathbf{r} - \mathbf{b}, \mathbf{A}\mathbf{w}\rangle, \end{aligned} \tag{8}$$

where $\langle \cdot, \cdot \rangle$ denotes an inner product. Similarly for the latter term of (6),

$$\frac{\mathrm{d}}{\mathrm{d}u}\|\mathbf{D}(\mathbf{r} + u\mathbf{w}) - \mathbf{e}\|_2^2\Big|_{u=0} = 2\langle \mathbf{D}\mathbf{r} - \mathbf{e}, \mathbf{D}\mathbf{w}\rangle. \tag{9}$$

By substituting these into (7) and dividing by 2, we have

$$\langle \mathbf{A}\mathbf{r} - \mathbf{b}, \mathbf{A}\mathbf{w}\rangle + \lambda\langle \mathbf{D}\mathbf{r} - \mathbf{e}, \mathbf{D}\mathbf{w}\rangle = 0. \tag{10}$$

By exploiting the properties of the inner product, it follows that

$$\left\langle \left(\mathbf{A}^{\mathrm{T}}\mathbf{A} + \lambda\mathbf{D}^{\mathrm{T}}\mathbf{D}\right)\mathbf{r} - \left(\mathbf{A}^{\mathrm{T}}\mathbf{b} + \lambda\mathbf{D}^{\mathrm{T}}\mathbf{e}\right), \mathbf{w}\right\rangle = 0. \tag{11}$$

Since this has to hold for all $\mathbf{w} \neq \mathbf{0}$, we finally get

$$\mathbf{r}_\lambda = \left(\mathbf{A}^{\mathrm{T}}\mathbf{A} + \lambda\mathbf{D}^{\mathrm{T}}\mathbf{D}\right)^{-1}\left(\mathbf{A}^{\mathrm{T}}\mathbf{b} + \lambda\mathbf{D}^{\mathrm{T}}\mathbf{e}\right). \tag{12}$$

Critical point of (6) is unique as long as the matrix $(\mathbf{A}^{\mathrm{T}}\mathbf{A} + \lambda\mathbf{D}^{\mathrm{T}}\mathbf{D})$ is nonsingular. This is true if and only if the null spaces (i.e., kernels) of matrices $\mathbf{A}$ and $\mathbf{D}$ intersect trivially: $\ker(\mathbf{A}) \cap \ker(\mathbf{D}) = \{\mathbf{0}\}$ [8]. Given that $\mathbf{A}$ is nonsingular, this is the case independently of $\mathbf{D}$. Thus, the critical point of (6) is always unique. Because the minimization problem (5) is known to have a solution, the found critical point must be located at the global minimum of the cost function $Q_\lambda(\mathbf{r})$.

Alternatively, the solution of (5) can be obtained by finding the least squares solution of

$$\begin{bmatrix} \mathbf{A} \\ \lambda^{1/2}\mathbf{D} \end{bmatrix}\mathbf{r}_\lambda = \begin{bmatrix} \mathbf{b} \\ \lambda^{1/2}\mathbf{e} \end{bmatrix}, \tag{13}$$

which is equivalent to (12). For later reference, system (12) is called *normal equations* and (13) *augmented system*.

2.2. Optimality Results of r_λ. Let us now examine the proposed method from the Bayesian point of view. For this, consider the special case of (5), where $M = 3N$ and $\mathbf{D} = \mathbf{I}$:

$$\mathbf{r}_\lambda = \underset{r}{\operatorname{argmin}}\|\mathbf{A}\mathbf{r} - \mathbf{b}\|_2^2 + \lambda\|\mathbf{r} - \mathbf{e}\|_2^2. \tag{14}$$

Furthermore, consider that $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma_{\boldsymbol{\epsilon}}^2\mathbf{I})$ and $\mathbf{r} \sim \mathcal{N}(\mathbf{e}, \sigma_{\boldsymbol{\theta}}^2\mathbf{I})$, where $\sigma_{\boldsymbol{\epsilon}}^2$ and $\sigma_{\boldsymbol{\theta}}^2$ represent variances of $\boldsymbol{\epsilon}$ and $\boldsymbol{\theta}$, respectively. Here, $\mathbf{r}$ is considered to be a random variable. Then, according to [23], the maximum a posteriori estimator of $\mathbf{r}$ is the solution of

$$\arg\min_r \|\mathbf{A}\mathbf{r} - \mathbf{b}\|_2^2 + \frac{\sigma_{\boldsymbol{\epsilon}}^2}{\sigma_{\boldsymbol{\theta}}^2}\|\mathbf{r} - \mathbf{e}\|_2^2. \tag{15}$$

That is, the choice $\lambda = \sigma_{\boldsymbol{\epsilon}}^2/\sigma_{\boldsymbol{\theta}}^2$ yields the best estimate of $\mathbf{r}$ in the sense that the likelihood of $\mathbf{r}$ is maximized.

Similar optimality results are known also for more general situations, where for example, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma_{\boldsymbol{\epsilon}}^2\mathbf{C}\mathbf{C}^{\mathrm{T}})$. For an extensive discussion of these situations, see [8, 24–26]. To the best of our knowledge, it is an open question whether similar results apply in case of the general form of (5).

2.3. The Choice of λ. In this section, we will provide an introduction to the most relevant systematical methods of finding λ.

According to [8], parameter-choice methods can be divided into two classes depending on their assumptions about the error norm $\|\boldsymbol{\epsilon}\|_2$:

(1) methods based on knowledge of $\|\boldsymbol{\epsilon}\|_2$,

(2) methods that do not require $\|\boldsymbol{\epsilon}\|_2$.

Out of these two classes of methods, our main interest is in class two. The reason for this is that knowledge of $\|\boldsymbol{\epsilon}\|_2$ is, in general, a too strict requirement to make. For reference, the most widespread parameter-choice method of class one is the *discrepancy principle* [8].

In class two, there are three popular and widely used methods. These are the *quasioptimality criterion* [8, 27], *generalized cross-validation* [8, 28], and *L-curve criterion* [8, 29, 30]. Let us next take a closer look at the L-curve method, which will be used here to choose the value of λ.

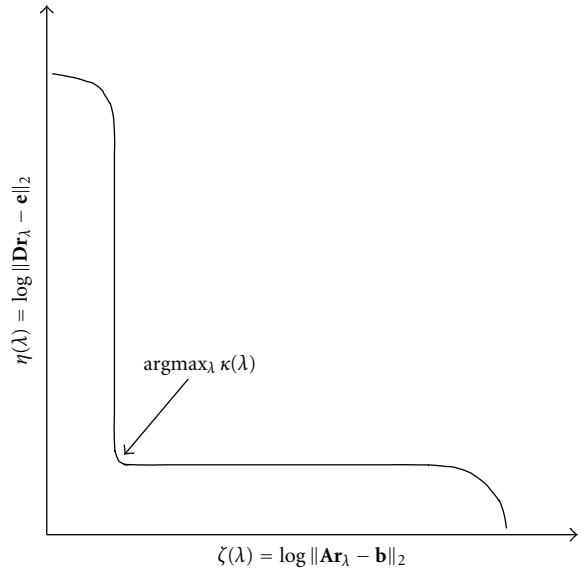

Figure 1: Generic form of the L-curve.

The reason for this choice is that the L-curve criterion is, in fact, accessible and a quite intuitive tool for the selection of λ. In the heart of this method is the shape of a certain curve (see Figure 1), which is parametrized as

$$[\zeta(\lambda), \eta(\lambda)] = [\log \|\mathbf{A}\mathbf{r}_\lambda - \mathbf{b}\|_2, \log \|\mathbf{D}\mathbf{r}_\lambda - \mathbf{e}\|_2]. \quad (16)$$

The idea of the L-curve criterion is to choose λ in such a way that there is a "good balance" between the values of the two norms in (5). In other words, balance between $\|\boldsymbol{\epsilon}\|_2$ and $\|\boldsymbol{\theta}\|_2$. Specifically, the L-curve criterion chooses the value of λ maximizing the curvature

$$\kappa(\lambda) = \frac{\dot{\zeta}\ddot{\eta} - \ddot{\zeta}\dot{\eta}}{\left[\left(\dot{\zeta}\right)^2 + \left(\dot{\eta}\right)^2\right]^{3/2}} \quad (17)$$

of the curve (16). In (17), the dots represent first $(\dot{\cdot})$ and second $(\ddot{\cdot})$ derivatives of ζ and η with respect to λ. The location of the point with maximum curvature is demonstrated in Figure 1. In practice, a suitable one dimensional optimization routine, such as quasi-Newton, can be used to find this point. For a more complete discussion of the reasoning behind the L-curve method, see, for example, [8]. It would also be interesting to compare existing parameter-choice methods in navigation applications, but this does not fit the scope of this paper.

3. Practical Aspects of the Proposed Method

In this section, we will discuss some practical aspects of Tikhonov regularization with L-curve criterion, when it is applied to inertial navigation. First, we will give some insight on the choice of **D** and then, discuss the numerical aspects of the problem.

3.1. The Choice of D. As mentioned in the introduction, **D** is determined by the application at hand. Indeed, an application with M position constraints, for example, results in different realization of **D** than an application with M velocity constraints would. Both of these are examples of constraints that bound position estimates only locally. Average velocity over longer period of time, trajectory of the object, or possible symmetry in the computed result are examples of constraints bounding an arbitrary number of time instances.

We will now present some examples to demonstrate exploitation of possible constraints. For readability and notational convenience, these examples are one dimensional. It is, however, straightforward to generalize the presented examples to the three dimensions. In the examples, only the nonzero elements of the corresponding row of **D** and **e** are presented:

(I) (position constraint) $r_j \sim \mathcal{N}(E(r_j), \sigma_i^2)$:

$$D_{i,j} = \frac{1}{\sigma_i^2}, \qquad e_i = \frac{E\left(r_j\right)}{\sigma_i^2}, \quad (18)$$

(II) (velocity constraint) $\dot{r}_j \sim \mathcal{N}(E(\dot{r}_j), \sigma_i^2)$:

$$D_{i,j-1} = \frac{-1}{(2h\sigma_i^2)}, \qquad D_{i,j+1} = \frac{1}{(2h\sigma_i^2)}, \qquad e_i = \frac{E\left(\dot{r}_j\right)}{\sigma_i^2}, \quad (19)$$

(III) position vector **r** is symmetric when mirrored around the middle element (where N is odd) with pointwise variance σ_i^2 (here, $j = N - i + 1$):

$$D_{i,i} = \frac{-1}{\sigma_i^2}, \qquad D_{i,j} = \frac{1}{\sigma_i^2}, \quad \forall i \in \left[1, \frac{N}{2}\right) \qquad e_i = 0. \quad (20)$$

In these examples, variance σ_i^2, $\forall i \leq M$, is used to give each constraint a suitable weight, similar to the weighted least squares method [31], assuming, of course, that θ_i are independent.

The fact that $\|\boldsymbol{\epsilon}\|_2$ is unknown and L-curve criterion is used to find λ gives rise to an interesting observation regarding the variance. In addition to its traditional interpretation, σ_i^2 can be treated as a measure of "relative" reliability of each constraint with respect to other constraints. The L-curve criterion is then used to find the "absolute" variances, given by $\lambda\sigma_i^2$.

In this section, we have only considered the situation where the M equations within (4) have different weights. Since (3) contains the measured accelerations and the required boundary conditions [1], the N equations within (3) can also have different significance. In this kind of a situation, one can introduce a diagonal weighting matrix $\mathbf{W}$, $W_{i,i} > 0$ and multiply both sides of (3) with it. In this paper, however, it is assumed that $\mathbf{W} = \mathbf{I}$.

3.2. Computational Aspects. The overall performance of the proposed method is determined by whether λ has to be solved or not. Namely, if we do not know it, it must be estimated as well, which requires that (5) must be solved for multiple different values of λ. The worse our initial guess for λ is, the harder it gets to find a reasonable value for it. While this "brute force" strategy might be a reasonable choice if the computations are made on a modern desktop computer, it is probably not a practical thing to do on a platform with limited resources. This is why we suggest that when possible, a suitable value of λ should be estimated beforehand, using either simulations or preferably, test measurements. After this, we can expect reasonable accuracy in similar situations, at least where the same type of measurement equipment is used. In example two, we use this strategy to avoid the re-evaluation of λ for each measurement run.

Independently of λ, problem (5) must be solved at least once. In order to avoid possible accuracy and/or performance issues, some attention should be paid. Although normal equations (12) and the augmented system (13) are mathematically equivalent, they are quite different from the numerical solution point of view. Namely, solving the system with normal equations is known to be inaccurate due to round-off errors when forming matrices $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ and $\lambda\mathbf{D}^{\mathrm{T}}\mathbf{D}$. The augmented system, on the other hand, can be accurately solved for with QR or SVD decomposition. There are specific solvers available also for sparse augmented systems [32].

Considering the computational complexity, the normal equations seem quite attractive: with a tridiagonal and symmetric matrix $\mathbf{A}$ [1], it is easy to see that $\mathbf{A}^{\mathrm{T}}\mathbf{A} = \mathbf{A}^2$ will be a symmetric matrix with a bandwidth of five. The structure of the regularization matrix $\mathbf{D}$ is, however, arbitrary in general. Yet in many situations, it will also be a banded matrix, like in the case of constraints (18) and (19). In case of (20), the situation looks worse, but significantly smaller bandwidth can be obtained by reordering the equations. In case of the normal equations, the resulting system has dimensions $3N \times 3N$ (for general three-dimensional problem) *independently* of M.

Consider a problem leading to a reasonably small bandwidth $p \in \mathbb{N}$ in case of the normal equations. For these situations, we propose an iterative refinement-based solution method similar to the *corrected seminormal equations* (CSNE) method [32]. Traditionally, CSNE is based on the use of a numerical approximation of the matrix $\widetilde{\mathbf{A}} = \mathbf{A}^{\mathrm{T}}\mathbf{A} + \lambda\mathbf{L}^{\mathrm{T}}\mathbf{L}$, computed with QR decomposition. In this case ($p \ll 3N$), we can, however, compute elements of $\widetilde{\mathbf{A}}$ analytically with negligible round-off errors. Thus, no QR decomposition is required and the corresponding algorithm for finding $\mathbf{r}$ is

$\mathbf{r}_{\mathrm{new}} \longleftarrow \mathbf{0}, \mathbf{r}_{\mathrm{old}} \longleftarrow \mathbf{1}$

While $\|\mathbf{r}_{\mathrm{new}} - \mathbf{r}_{\mathrm{old}}\| > \delta$ **do**

$\mathbf{r}_{\mathrm{old}} \longleftarrow \mathbf{r}_{\mathrm{new}}$

$\boldsymbol{\delta}_{\mathbf{r}} \longleftarrow \begin{bmatrix} \mathbf{b} \\ \sqrt{\lambda}\mathbf{e} \end{bmatrix} - \begin{bmatrix} \mathbf{A} \\ \sqrt{\lambda}\mathbf{D} \end{bmatrix} \mathbf{r}_{\mathrm{old}}$

$\mathbf{r}_{\mathrm{update}} \longleftarrow \widetilde{\mathbf{A}}^{-1}([\,\mathbf{A}^{\mathrm{T}} \ \sqrt{\lambda}\mathbf{D}^{\mathrm{T}}\,]\boldsymbol{\delta}_{\mathbf{r}})$

$\mathbf{r}_{\mathrm{new}} \longleftarrow \mathbf{r}_{\mathrm{old}} + \mathbf{r}_{\mathrm{update}}$

end While

with stopping criterion δ [32]. Notice that the solution of the involved matrix equation can be carried out with linear time complexity, as it requires $3Np^2 + 24Np + 3N$ flops [31]. In many occasions, one refinement step is enough, and more than three refinement steps are seldom required. For problems with "large" p, this approach is not practical, and solvers designed for augmented systems should be used instead.

4. Example 1

In this section, we will work out a simple one dimensional example demonstrating the use of the Tikhonov regularization method. The object of this example is to compare three different solution strategies, all of which are based on the finite element method:

(I) tikhonov regularization based on finite element method with the position fixed at both ends,

(II) finite element method with the position fixed at both ends,

(III) finite element method with the position fixed at both ends, where measurement error is modeled as described in [1].

4.1. Setting Up the Example. Let us generate the reference position $p(t)$, $t \in [0, 10]$ s as a fifth-order polynomial satisfying the following six conditions:

$$\begin{aligned} p(0) &= 0\,\mathrm{m}, \quad \dot{p}(0) = 0\,\mathrm{m/s}, \\ p(5) &= -10\,\mathrm{m}, \quad \dot{p}(5) = 0\,\mathrm{m/s}, \\ p(10) &= 10\,\mathrm{m}, \quad \dot{p}(10) = 0\,\mathrm{m/s}. \end{aligned} \tag{21}$$

From the resulting $p(t)$, it is an easy task to compute the reference velocity $v(t)$ and acceleration $a(t)$. Erroneous "measurement" $\hat{a}(t)$, containing scale factor error s and bias term μ with additional white noise, is generated as follows:

$$\hat{a}(t) = (1 + s)a(t) + \mathcal{N}(\mu, \sigma^2). \tag{22}$$

In Figure 2, one realization of $\hat{a}(t)$ ($s = 0.05$, $\mu = -0.50$ and $\sigma^2 = 1$), sampled with a frequency of 50 Hz ($N = 501$), is presented with the reference acceleration $a(t)$.

Now, suppose that along with $\hat{a}(t)$, we know the position and velocity of the object at points $t = 0$ seconds and $t = 10$ seconds. Thus, we have the following problem: find $\hat{p}(t)$,

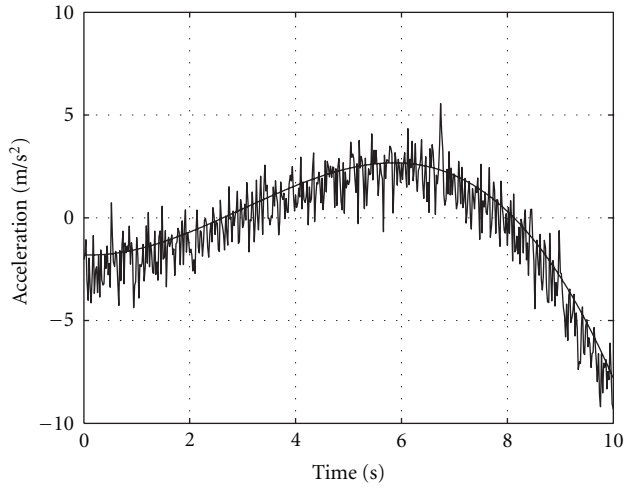

Figure 2: Actual acceleration (thick line) and "measured" acceleration (thin line) as a function of time.

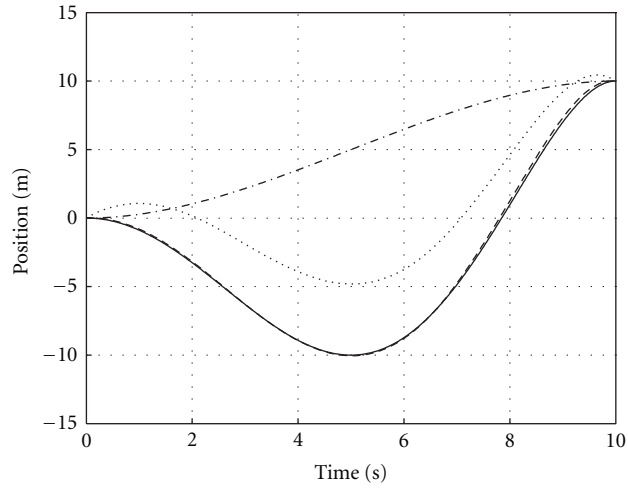

Figure 3: Actual position (solid line), estimated position computed with method I (dashed line, $\lambda = 3.611$), estimated position computed with method II (dotted line) and used prior (dash-dotted line) as a function of time.

$0 \leq t \leq 10$ such that

$$\begin{aligned} \ddot{\hat{p}}(t) &= \hat{a}\,(t), \\ \hat{p}(0) &= 0\,\text{m}, \\ \dot{\hat{p}}(0) &= 0\,\text{m/s}, \\ \hat{p}(10) &= 10\,\text{m}, \\ \dot{\hat{p}}(10) &= 0\,\text{m/s}. \end{aligned} \tag{23}$$

The problem here is, of course, that we do not know $a(t)$, but only the erroneous measurement $\hat{a}(t)$. Thus, ultimately, the goal is to minimize the effects of the measurement error using the given two additional boundary conditions.

To exploit method I, we must first generate a suitable prior, which can then be used to enhance the accuracy of the solution. For this, let us compute a polynomial of the lowest order that satisfies the given four conditions at the boundaries and use that as our prior. In this case, a suitable choice for the regularization matrix $\mathbf{D}$ would be $N \times N$ identity matrix. However, to emphasize the fact that we have no trust in our prior between the two boundaries, we will zero out all but, say, 10 first and last rows of $\mathbf{D}$. Thus, in this example, $M = 20$. Method II follows from the method I by setting $\lambda = 0$. For method III, we use the error model presented in [1], modeling constant scale factor and bias errors using the velocity of the object at $t = 0$ and $t = 10$. Knowing the form of the generated measurements, this model will yield the exact solution when no noise is applied. The idea here is, however, to test the performance of the method III in case of noisy measurements.

4.2. Results. Table 1 summarizes the test results, which were run with a total of nine different combinations of error parameters s and μ. Each test was run several times with different realizations of random noise in order to show the average performance of the tested methods. As an overall comment, the tested methods are not particularly sensitive to noise. The accuracy of the method III is, however, dependent on the value of σ^2. This was expected, since method III uses only two measurements to determine the two error parameters, which gets more inaccurate as the noise level increases. The value of λ minimizing the position error is presented only for reference, as there is no practical way to find it in real situations. Value of λ_L, however, can be found for each measurement using the L-curve criterion. During the test, the value of λ minimizing the velocity error was also recorded, but the difference to the value minimizing the position error was not significant.

Based on the results seen in Table 1, the L-curve method seems to work reasonably well. Indeed, in some cases it is even able to predict the optimal value of λ (or at least obtain position error levels very close to the minimum). On average, Tikhonov regularization with λ_L works much (five to ten times) better than the method II. It can also fail, meaning that the resulting error is larger than the one given by method II. This happened only when the optimal value of λ was zero, and the results of method II were optimal. The L-curve method, on the other hand, performed consistently among the test situations. When compared with method III, the difference is not as clear, although on average, Tikhonov regularization does slightly better. As mentioned above, this is due to the large value of σ^2. In situations with lower noise and known main error sources, a combination of methods I and III is also a reasonable strategy. This is due to the fact that the error modeling can reduce such components of error (such as bias) with which Tikhonov regularization is not intended to work with.

Figures 3 and 4 illustrate the performance of methods I and II in terms of position and velocity. The measurement was the one seen in Figure 2 with parameters $s = 0.05$ and $\mu = -0.5$. Value $\lambda = \lambda_L$ was used to compute the regularized solution. The figures also demonstrate the used prior, only first and last 0.2 seconds of which were actually

Table 1: Simulation results for various values of parameters s and μ. Terms σ_{I}, σ_{II} and σ_{III} refer to the standard deviation of the position error (in meters) in case of solution methods I, II, and III. The value of the regularization parameter obtained via the L-curve method is represented as λ_L and λ represents the value of the regularization parameter minimizing the position error.

s	μ	σ^2	$\mathrm{argmin}_\lambda \sigma_{\mathrm{I}}(\lambda)$	λ	$\sigma_{\mathrm{I}}(\lambda_L)$	λ_L	σ_{II}	σ_{III}
−0.05	−0.5	1	1.137	4.200	1.137	4.501	5.310	1.236
−0.05	0.0	1	0.453	0.000	1.024	1.438	0.453	0.568
−0.05	0.5	1	0.109	0.149	0.995	5.663	4.040	1.413
0.00	−0.5	1	0.813	4.073	0.813	4.255	4.781	0.723
0.00	0.0	1	0.282	0.000	0.737	1.330	0.282	0.693
0.00	0.5	1	0.101	0.221	0.729	5.185	4.353	1.015
0.05	−0.5	1	0.286	3.407	0.286	3.611	3.823	0.757
0.05	0.0	1	0.139	1.230	0.260	1.581	0.666	1.634
0.05	0.5	1	0.132	0.678	0.336	5.243	4.881	0.758

$\hat{a}(t) = (1+s)a(t) + \mathcal{N}(\mu, \sigma^2)$.

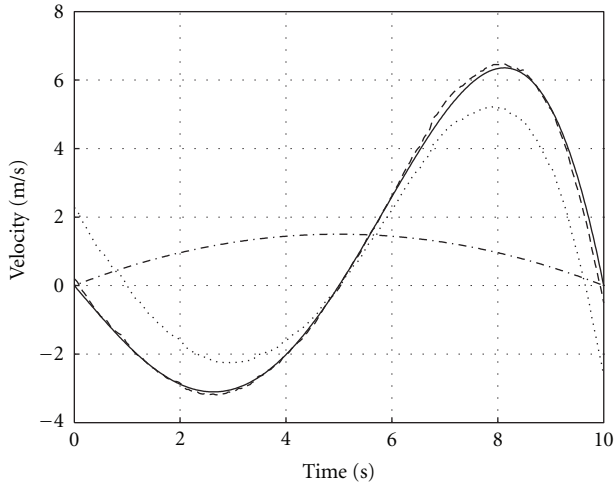

Figure 4: Actual velocity (solid line), estimated velocity computed with method I (dashed line, $\lambda = 3.611$), estimated velocity computed with method II (dotted line) and used prior (dash-dotted line) as a function of time.

exploited. Clearly, comparing to method II, the accuracy of the proposed method is remarkably better. In terms of the standard deviation, the proposed method is about 13 times more accurate than method II. When comparing to method III, the proposed method is about 2.5 times more accurate, which is also a significant improvement.

In Table 2, the computational complexity of the proposed method is demonstrated. The number of samples (N) and constraints (M) was increased by increasing the sampling frequency, keeping the problem otherwise unchanged. The value of λ was kept fixed. As the solution methods were not optimized for maximum performance, the differences in the solution times may not be fully comparable. Especially in case of the LSQR method, default tolerance (relative accuracy of $1e-6$) was used. Despite the "large" number of unknowns, the solution times are small for any solution method. Based on the results, one can expect linear time complexity with respect to $N + M$ for similar problems. In the last case with $N = 128000$, the efficiency of the CSNE method drops as the number of iterations increases to seven. This is due to the fact that the conditioning of the system is comparable to N^2. Otherwise, the CSNE method seems to provide good efficiency with a reasonably small cost in the accuracy.

The main purpose of this example was to demonstrate the performance of the proposed method in situations where the measurement error is biased (due to error term μ) and not normally distributed (due to error term s). That is, in situations where the presented optimality results are not valid. The results indicate that the proposed method is useful also in these situations, commonly encountered in practice. Moreover, the tests indicate that the proposed method can be very efficient with negligible computing times even for large-scale problems with over 100000 unknowns. This also guarantees that when required, also the value of λ can be determined with reasonably small computational cost.

5. Example 2

In this section, we will demonstrate the use of the regularization method with actual measurement data obtained in a real-world situation. The data was obtained for use with a television program of NHK (Japan Broadcasting Corporation) and presented here with due permission. The primary goal was to measure the accelerations to which the passengers were subjected in a car of a roller coaster (The roller coaster in question is called Insane, located at Gröna Lund, Stockholm, Sweden.) In this example, we present a way to estimate the velocity and trajectory of the car with knowledge about the trajectory of the car. It is important to notice that only the trajectory is known, not its time parametrization.

5.1. Measurement Setup. The car moved on a planar trajectory of length ≈ 250 m and rotated freely around its rotation axis (orthogonal to the plane) during the ride. In other words, the problem has three degrees of freedom: one for the rotation and two for the displacements. The trajectory is not known accurately, but roughly estimated using the

Table 2: Computing times as N and M increase. All times are average computing times over 100 cycles and include only the solution phase. Tests were performed with Matlab 7.9.0 running on a 32-bit operating system. The used computer has a 2 GHz processor and 2 GB of memory. CSNE stands for the method described in Section 3.2, "\" for the "backslash" operator, and LSQR for the iterative least-squares problems solution routine implemented in Matlab. "It" indicates the number of iterations made by CSNE and $\delta/N^{1/2}$ indicates the "size" of the last correction step.

			CSNE		"\"	LSQR
N	M	It	T (ms)	$\delta/N^{1/2}$	T (ms)	T (ms)
500	20	1	1	$6.4e-9$	2	5
1000	40	1	2	$3.4e-8$	4	7
2000	80	1	4	$3.1e-7$	8	12
4000	160	1	8	$5.5e-6$	16	20
8000	320	1	16	$8.7e-5$	34	36
16000	640	1	34	$4.1e-4$	76	63
32000	1280	2	103	$1.7e-5$	148	117
64000	2560	2	212	$4.0e-4$	302	271
128000	5120	7	1106	$3.1e-4$	635	618

length and an image of the trajectory. Because of this, the estimated trajectory is known to contain significant errors, which makes it a good example of situations with a rather vague prior.

The data was collected using an inertial measurement unit (IMU) consisting of the following sensors:

(i) $3 \times \pm 75^\circ$/sec Silicon Sensing CRS10 [33],

(ii) $3 \times \pm 300^\circ$/sec AD ADXRS300ABG [34],

(iii) $3 \times \pm 3$ g VTI SCA610-CC5H1A [35],

(iv) $3 \times \pm 12$ g VTI SCA620-CHCV1A [36].

The IMU is calibrated as described in [37]. Only the sensors of the lower dynamic range were exploited unless the input exceeded this range. In this case, only the sensors with the higher dynamical range were exploited. The sample rate was 1000 Hz, and the total number of unknown displacements was about 200000 in each measurement. Based on the construction of the roller coaster, the initial and the final angle were known to be identical. This allowed the angle to be computed with high accuracy. Thus, the accelerometers were the major source of error.

5.2. Exploitation of the Estimated Trajectory. In this section, the methods used to exploit the estimated trajectory are presented. Notice that only the trajectory is estimated, not the time when the car is at a certain point. It is assumed that the horizontal coordinates of the trajectory are stored to vector $\mathbf{X} \in \mathbb{R}^P$ and the vertical coordinates, respectively, to vector $\mathbf{Y} \in \mathbb{R}^P$. The number of the points ($P \in \mathbb{N}$) is assumed to be high enough and the trajectory data "smooth" enough such that the local tangent of the trajectory can be reliably estimated using two adjacent points of the trajectory.

Given a horizontal coordinate r_j, $j \in [1, N]$ such that $X_{i-1} \leq r_j < X_i$, $i \in [1, P]$, the corresponding vertical coordinate r_{N+j} satisfies the equation

$$(Y_i - Y_{i-1})r_j - (X_i - X_{i-1})r_{N+j} = X_{i-1}Y_i - X_iY_{i-1}. \quad (24)$$

Depending on the trajectory, it is possible that the index i is ambiguous, meaning that there can be more than one vertical coordinate corresponding to a single horizontal coordinate. This is the case also here, as can be verified from Figure 5. Thus, it is also required to choose correct i. It is also possible that no suitable index i is found, meaning that the value of r_j is out of range. In such a situation, the trajectory is not exploited for j in question. With these notions, the corresponding elements of the regularization term are

$$\begin{aligned} D_{j,j} &= Y_i - Y_{i-1}, \\ D_{j,N+j} &= X_i - X_{i-1}, \qquad \forall\, j \in [1, N]. \\ e_j &= X_{i-1}Y_i - X_iY_{i-1}, \end{aligned} \quad (25)$$

5.3. Solution Process. The problem was solved in two phases. In the first phase, only few points of the trajectory were exploited. These were the points corresponding to the local extrema of the horizontal coordinates of the stored trajectory. The goal of this phase was to produce a reasonable initial guess for $\mathbf{r}$, which could then be used to obtain better results when linearizing the trajectory. For this, valid estimates of the indices j when the car crosses these points were required. In this phase, we used a rather high value $\lambda = 100$ to minimize the amount of values r_j that were out of range in the next phase. After this phase, the measurement vector $\mathbf{b}$ was updated by setting $\mathbf{b} \leftarrow \mathbf{A}\mathbf{r}_\lambda$.

In phase two, the trajectory of the car was exploited throughout the track. This was done by linearizing the estimated trajectory at each time instance in the neighborhood of the horizontal position obtained in the first phase, as explained in Section 5.2. Variance of each constraint was set to one, as we had no knowledge of the possible differences in the reliability of these estimates. For the first run of the car, the value of the regularization parameter was obtained using the L-curve criterion. The same value of λ was then used also for the second and third run.

5.4. Results. In Figure 5, the computed three trajectories are compared to the used prior. In each case, the car starts

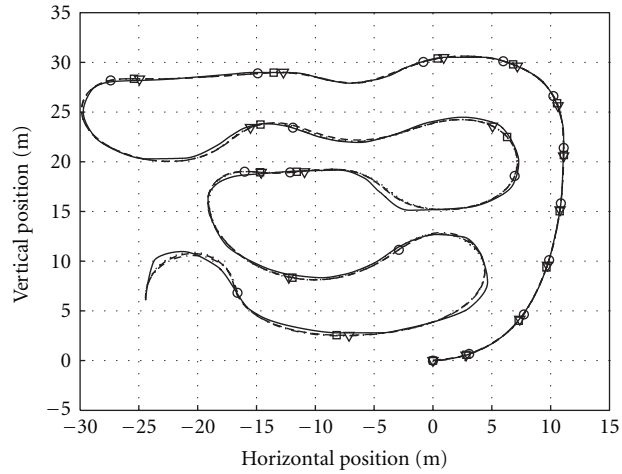

Figure 5: Actual trajectory (solid line), estimated trajectory of run 1 (dashed line), run 2 (dash-dotted line), and run 3 (dotted line). Small squares, circles, and triangles indicate the position of the car with three second increments, respectively, for runs 1, 2, and 3. Regularization parameter $\lambda = 0.352$ for all three runs.

at point $[0, 0]$ leaving to the upright direction. The end point was set to the last local extremum of the horizontal coordinates of the estimated trajectory, because the time instance corresponding to the actual stopping point could not be specified. Notice how the three computed trajectories are almost identical, while they follow the provided prior only approximately. Given that no information about the reliability of different points of the trajectory is provided, this indicates that the L-curve method provides useful estimates of the regularization parameter. Since this happens for all three runs, determining the value of λ in advance, as discussed before, is a plausible technique.

The small squares, circles, and triangles seen in Figure 5 indicate the position of the car with three second increments, respectively, for runs 1, 2, and 3. As seen in the figure, the distances between these marks change significantly, indicating that the used method "allows" the velocity of the car to change between different runs. For the first ≈ 50 meters of the trajectory, however, the distance between the marks does not change much. This is due to the fact that during these first meters, the cars were lifted up essentially the same way for each run. After the highest point of the trajectory, the car moved freely along the track "driven" only by gravity and affected by the moment of inertia, determined by the placements of the passengers. As seen in Figure 5, the run represented with the circles, for example, attained significantly higher velocities than the other two runs.

6. Conclusion

A BVP formulation of inertial navigation problems is further investigated using [1] as a starting point. It is suggested that the possible additional information can be taken into account by exploiting Tikhonov regularization, a basic tool of inverse problems. In addition to typical additional information, such as a momentary position or velocity, it allows one to exploit more general forms of information. These include, but are not limited to, average velocity over longer period of time, trajectory of the object, and possible symmetry in the computed result. In the provided examples, significant accuracy improvements—up to an order of magnitude—over the basic FEM solution without any additional information are obtained.

It is also demonstrated that the proposed method can be viewed as a Bayesian estimator, yielding the maximum a posteriori likelihood estimator in case of unbiased and uncorrelated measurement errors. In addition to this, the provided examples show that the obtained accuracy improvement is significant, even in cases where these assumptions are not met.

A method for choosing the value of the regularization parameter is provided and demonstrated to work in a real-world example. This method, called the L-curve criterion, does not require any prior knowledge of the measurement errors. Thus, it can be used in many real-world situations, where measurement error is always present, but seldom reliably characterized. For performance-critical situations, it is possible to determine a suitable value for the regularization parameter in advance and use this value to obtain good results.

In some applications, it is possible and advantageous to combine the method based on sensor error modeling [1] with the method proposed herein. Sensor error modeling is used first to eliminate the modeled deterministic measurement errors, and Tikhonov regularization is then used to minimize the effects of the remaining stochastic components of the measurement error.

Acknowledgments

The authors would like to thank D.Tech. Saku Suuriniemi and Professor Stefan Kurz for proofreading and improving the clarity of the contents of this paper. They would also like to thank Olli Särkkä (M.S. degree) for his invaluable assistance while obtaining the measurements of Example 2.

References

[1] T. Nieminen, J. Kangas, S. Suuriniemi, and L. Kettunen, "Accuracy improvements by boundary conditions for inertial navigation," *International Journal of Navigation and Observation*, vol. 2010, Article ID 869127, 10 pages, 2010.

[2] L. F. Shampine, I. Gladwell, and S. Thompson, *Solving ODEs with Matlab*, Cambridge University Press, Cambridge, UK, 2003.

[3] U. M. Asher and L. R. Petzold, *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, SIAM, Philadelphia, Pa, USA, 1998.

[4] A. N. Tikhonov, "On the stability of inverse problems," *Doklady Akademii Nauk SSSR*, vol. 39, no. 5, pp. 195–198, 1943.

[5] A. N. Tikhonov, "Solution of incorrectly formulated problems and the regularization method," *Doklady Akademii Nauk SSSR*, vol. 151, pp. 501–504, 1963.

[6] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill Posed Problems*, VH Winston, Washington, Wash, USA, 1977.

[7] D. L. Phillips, "A technique for the numerical solution of certain integral equations of the first kind," *Journal of the ACM*, vol. 9, no. 1, pp. 84–97, 1962.

[8] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*, SIAM, Philadelphia, Pa, USA, 1998.

[9] A. E. Hoerl, "Application of ridge analysis to regression problems," *Chemical Engineering Progress*, vol. 58, pp. 54–59, 1962.

[10] D. W. Marquardt, "Generalized inverses, ridge regression, biased linear estimation, and nonlinear estimation," *Technometrics*, vol. 12, no. 3, pp. 591–612, 1970.

[11] A. Gelb, *Applied Optimal Estimation*, The M.I.T. Press, Cambridge, Mass, USA, 1974.

[12] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*, John Wiley & Sons, New York, NY, USA, 1st edition, 2001.

[13] J. L. Crassidis and J. L. Junkins, *Optimal Estimation of Dynamic Systems*, Chapman & Hall/CRC, Boca Raton, Fla, USA, 2004.

[14] P. D. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*, Artech House, Boston, Mass, USA, 2008.

[15] D. C. Fraser and J. E. Potter, "The optimum linear smoother as a combination of two optimum linear filters," *IEEE Transactions on Automatic Control*, vol. 14, no. 4, pp. 387–390, 1969.

[16] H. E. Rauch, F. Tung, and C. T. Striebel, "Maximum likelihood estimates of linear dynamic systems," *AIAA Journal*, vol. 3, no. 8, pp. 1445–1450, 1965.

[17] K. Gade, "NavLab, a generic simulation and post-processing tool for navigation," *European Journal of Navigation*, vol. 2, no. 4, pp. 51–59, 2004.

[18] Y. Yang, Z. Jin, W. Tian, and F. Qian, "Application of fixed interval smoothing to gps/dr integrated navigation system," in *Proceedings of Intelligent Transportation Systems*, vol. 2, pp. 1027–1031, October 2003.

[19] A. B. Willumsen and Ø. Hegrenæs, "The joys of smoothing," in *IEEE Bremen: Balancing Technology with Future Needs (OCEANS '09)*, pp. 1–7, May 2009.

[20] T. Roths, M. Marth, J. Weese, and J. Honerkamp, "A generalized regularization method for nonlinear ill-posed problems enhanced for nonlinear regularization terms," *Computer Physics Communications*, vol. 139, no. 3, pp. 279–296, 2001.

[21] G. H. Golub, P. C. Hansen, and D. P. O'Leary, "Tikhonov regularization and total least squares," *SIAM Journal on Matrix Analysis and Applications*, vol. 21, no. 1, pp. 185–194, 2000.

[22] M. Vauhkonen, D. Vadâsz, P. A. Karjalainen, E. Somersalo, and J. P. Kaipio, "Tikhonov regularization and prior information in electrical impedance tomography," *IEEE Transactions on Medical Imaging*, vol. 17, no. 2, pp. 285–293, 1998.

[23] E. Demidenko, *Mixed Models: Theory and Applications*, John Wiley & Sons, Hoboken, NJ, USA, 2004.

[24] A. Tarantola, *Inverse Problem Theory*, SIAM, Philadelphia, Pa, USA, 2005.

[25] J. N. Franklin, "Well-posed stochastic extensions of ill-posed linear problems," *Journal of Mathematical Analysis and Applications*, vol. 31, no. 3, pp. 682–716, 1970.

[26] D. D. Jackson, "The use of a priori data to resolve non-uniqueness in linear inversion," *Geophysical Journal of the Royal Astronomical Society*, vol. 57, no. 1, pp. 137–157, 1979.

[27] M. Hanke and T. Raus, "A general heuristic for choosing the regularization parameter in ill-posed problems," *SIAM Journal of Scientific Computing*, vol. 17, no. 4, pp. 956–972, 1996.

[28] G. H. Golub, M. Heath, and G. Wahba, "Generalized cross-validation as a method for choosing a good ridge parameter," *Technometrics*, vol. 21, no. 2, pp. 215–223, 1979.

[29] H. W. Engl and W. Grewer, "Using the l-curve for determining optimal regularization parameters," *Numerische Mathematik*, vol. 69, no. 1, pp. 25–31, 1994.

[30] P. C. Hansen, "Analysis of discrete illposed problems by means of the l-curve," *SIAM Review*, vol. 34, no. 4, pp. 561–580, 1992.

[31] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore, Md, USA, 3rd edition, 1996.

[32] Å. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, Pa, USA, 1996.

[33] "Silicon Sensing CRS10 Datasheet," 2007. Revision 2, Checked on 23.09.2010.

[34] "Analog Devices ADXRS300ABG Datasheet," 2004. Revision B, Checked on 22.12.2009.

[35] "VTI SCA610-CC5H1A Datasheet," 2005. Revision 3, Checked on 23.09.2010.

[36] "VTI SCA620-CHCV1A Datasheet," 2006. Revision 2/2, Checked on 22.12.2009.

[37] T. Nieminen, J. Kangas, S. Suuriniemi, and L. Kettunen, "An enhanced multi-position calibration method for consumer-grade inertial measurement units applied and tested," *Measurement Science and Technology*, vol. 21, no. 10, Article ID 105204, 2010.

Publication 4

Tuukka Nieminen, Jari Kangas, Saku Suuriniemi and Lauri Kettunen

"A non-recursive fixed-interval smoothing -based approach to attitude estimation"

Submitted to *IEEE Transactions on Aerospace and Electronic Systems*

Tampereen teknillinen yliopisto
PL 527
33101 Tampere

Tampere University of Technology
P.O.B. 527
FI-33101 Tampere, Finland

ISBN 978-952-15-3053-1
ISSN 1459-2045