Linköping studies in science and technology. Dissertations.
No. 1623

Inertial Navigation and Mapping for Autonomous Vehicles

Martin Skoglund

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

Linköping 2014

Cover illustration: The cover illustration is left intentionally black!

Linköping studies in science and technology. Dissertations.
No. 1623

Inertial Navigation and Mapping for Autonomous Vehicles

Martin Skoglund

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

ISBN 978-91-7519-233-8 ISSN 0345-7524

Printed by LiU-Tryck, Linköping, Sweden 2014

To Maria

Abstract

Navigation and mapping in unknown environments is an important building block for increased autonomy of unmanned vehicles, since external positioning systems can be susceptible to interference or simply being inaccessible. Navigation and mapping require signal processing of vehicle sensor data to estimate motion relative to the surrounding environment and to simultaneously estimate various properties of the surrounding environment. Physical models of sensors, vehicle motion and external influences are used in conjunction with statistically motivated methods to solve these problems. This thesis mainly addresses three navigation and mapping problems which are described below.

We study how a vessel with known magnetic signature and a sensor network with magnetometers can be used to determine the sensor positions and simultaneously determine the vessel's route in an *extended Kalman filter* (EKF). This is a so-called *simultaneous localisation and mapping* (SLAM) problem with a reversed measurement relationship.

Previously determined hydrodynamic models for a *remotely operated vehicle* (ROV) are used together with the vessel's sensors to improve the navigation performance using an EKF. Data from sea trials is used to evaluate the system and the results show that especially the linear velocity relative to the water can be accurately determined.

The third problem addressed is SLAM with inertial sensors, accelerometers and gyroscopes, and an optical camera contained in a single sensor unit. This problem spans over three publications.

We study how a SLAM estimate, consisting of a point cloud map, the sensor unit's three dimensional trajectory and speed as well as its orientation, can be improved by solving a *nonlinear least-squares* (NLS) problem. NLS minimisation of the predicted motion error and the predicted point cloud coordinates given all camera measurements is initialised using EKF-SLAM.

We show how NLS-SLAM can be initialised as a sequence of almost uncoupled problems with simple and often linear solutions. It also scales much better to larger data sets than EKF-SLAM. The results obtained using NLS-SLAM are significantly better using the proposed initialisation method than if started from arbitrary points. A SLAM formulation using the *expectation maximisation* (EM) algorithm is proposed. EM splits the original problem into two simpler problems and solves them iteratively. Here the platform motion is one problem and the landmark map is the other. The first problem is solved using an extended Rauch-Tung-Striebel smoother while the second problem is solved with a quasi-Newton method. The results using EM-SLAM are better than NLS-SLAM both in terms of accuracy and complexity.

Populärvetenskaplig sammanfattning

Vi människor utför dagligen en stor mängd navigerings- och karteringsrutiner utan att ens reflektera över dessa. Allt ifrån att resa mellan hem och arbete till att upptäcka nya spännande miljöer i en ny stad. För navigering i välbekanta miljöer använder vi våra sinnen och den mentala karta av omvärlden som är uppbygd av tidigare upplevelser. Att navigera i okända miljöer är svårare eftersom vi måste strukturera en ny mental karta av alla intryck och samtidigt uppfatta var i kartan vi befinner oss. Lyckligtvis finns en stor mängd hjälpmedel såsom, kartor, kompasser, satellitpositioneringssystem, med mera, som underlättar för oss.

Navigering och kartering i okända miljöer är även en viktig byggsten för obemannade farkosters ökade autonomi eftersom externa positioneringssystem kan vara störningskänsliga eller helt enkelt otillgängliga. Navigering och kartering kräver, ofta omfattande, signalbehandling av farkosternas sensordata för att uppskatta hur den rör sig relativt omvärlden och att samtidigt uppskatta olika egenskaper i omvärlden. Fysikaliska modeller av sensorer, farkosters rörelse och yttre påverkan används tillsammans med statistiskt motiverade metoder för att lösa dessa problem så bra som möjligt. Denna avhandling behandlar huvudsakligen tre navigerings- och karteringsproblem som beskrivs nedan.

Säkerhetskritiska maritima miljöer, såsom hamnar, kylvattenanläggningar vid kärnkraftverk och andra skyddsobjekt kräver ständig övervakning. För att detektera och göra målföljning av främmande fartyg och undervattensfarkoster i dessa miljöer kan man använda sig av sensorer placerade i ett nätverk på havsbotten. Dessa nätverk fungerar endast tillförlitligt om sensorernas position och orientering på havsbotten är känd. Vi studerar hur ett fartyg med känd magnetisk signatur och ett sensornätverk med tre-axliga magnetometrar kan användas för att bestämma sensorernas position och samtidigt bestämma fartygets rutt med ett så kallat *extended Kalman filter* (EKF). Detta är ett så kallat samtidig lokalisering och karteringsproblem, på engelska *simultaneous localisation and mapping* (SLAM), men med omvänd mätrelationen. Analys av hur noggrant sensorernas position kan bestämmas för en given rutt redovisas och det omvända, alltså hur noggrant en rutt kan bestämmas när sensorernas positioner är kända. Dessutom visas med känslighetsanalys att om fartyget är utrustat med positioneringssystem så kan fel i sensororientering och fel i den magnetiska signaturen undertryckas och positonsnoggrannheten för sensorerna förbättras. Systemet utvärderas med hjälp av simuleringar.

Det andra problemet som behandlas är modellering en fjärrstyrd obemannad undervattensfarkost, på engelska *remotely operated vehicle* (ROV), och sensorfusion. ROVar används i en mängd krävande undervattenstillämpningar såsom tunnel- och skrovinspektion, svetsning på oljeriggar där djupet är för stort för vanliga dykare och arkeologiska expeditioner. ROVar har ett begränsat utrymme för navigerings- och karterinssensorer och dessa måste även vara relativt billiga. Dessutom saknas det ofta externa positioneringssystem under vattnet och därför krävs oberoende robusta navigeringsmetoder. Vi studerar hur tidigare bestämda

hydrodynamiska delmodeller för en ROV kan användas tillsammans med farkostens sensorer för att förbättra navigeringsprestanda. De hydrodynamiska modellerna beskriver farkostens rörelse i det omgivande vattnet där rörelsen genereras av de fem propellrarna. Vi redovisar modeller för alla sensorer uttryckta i ett kroppsfixt system och dessa används tillsammans med de hydrodynamiska modellerna i ett EKF. Experimentella data från sjökörningar i Vättern används för att utvärdera systemet och resultaten visar att framför allt linjär hastighet relativt vattnet kan bestämmas noggrant.

Det tredje problemet som behandlas är SLAM med tröghetssensorer (accelerometrar och gyroskop) och optisk kamera i en sammansatt sensormodul. Detta problem sträcker sig över tre publikationer.

I den första publikationen studerar vi hur en SLAM-skattning, bestående av en skalenligt punktmolnskarta, sensormodulens tre-dimensionella bana och hastighet samt dess orientering, kan förbättras genom att lösa ett olinjärt minstakvadratproblem. Den initiala SLAM-skattningen görs med EKF-SLAM och det förbättrade estimated fås genom att minimera det predikterade rörelsefelet över hela banan och de predikterade punktmolnskoordinaterna givet alla kameramätningar och där tröghetssensorerna behandlas som en känd insignal. Resultaten är utvärderade med väldigt tillförlitliga data där sensormodulen är fastspänd i verktygspositionen på en industrirobot och miljön uppmätt för hand.

I den andra publikationen visar vi hur det olinjära minstakvadratproblemet för SLAM kan initialiseras med en sekvens av nästan frikopplade problem med enkla, och oftast linjära, lösningar. Fördelen med att använda denna metod, istället för EKF-SLAM, är att den kan användas på mycket större dataset. Vi visar även att resultatet från det olinjära minstakvadratproblemet blir bättre med den föreslagna initialiseringsmetoden än om den startas från godtyckliga punkter. Flera steg i metoden utvärderas med Monte Carlo simuleringar och experimentella data.

I den tredje publikationen studeras en alternativ formulering till det olinjära minstakvadratproblemet med hjälp av *expectation maximisation* (EM) algoritmen. EM är en metod som används för att hantera komplexa problem genom att dela upp det i två enklare problem och lösa dessa iterativt. I den föreslagna metoden betraktas sensormodulens bana som det ena problemet och det andra problemet utgörs av punktmolnskartan. EM metoden initialiseras med den ovan beskrivna metoden och resultat på experimentell och simulerad data visar sig ge mindre medel-fel än med den olinjära minstakvadratmetoden. Dessutom är metoden beräkningsmässigt mer effektiv.

Acknowledgments

The quote *"I travel light. But not at the same speed. "* – *Jarod Kintz*, perfectly summarises that my long journey has *finally* reached its end. This would of course not have been possible without the support and encouragement from colleagues, friends and family. My first acknowledgement is directed to my supervisor Prof Fredrik Gustafsson, your inspiring tempo, professional sharpness and relaxed attitude to the never-ceasing flood of tight deadlines is impressive. Thanks for dragging me along this fascinating ride. Thanks! I also like to thank my co-supervisor Prof Thomas B Schön. Thomas, you have a very professional attitude and is an excellent researcher and reviewer with a bunch of cool estimation skills up your sleeve. Thanks! Various parts of this thesis has also been proof-read by Dr Zoran Sjanic and Dr Gustaf Hendeby. Thanks for all the comments and letting me borrow your time! All remaining eRrors are mine! Thanks to Dr Gustav Hendeby and Dr Henrik Tidefelt for help with figures in *Shapes* and LaTeX questions and for swiftly improving the thesis template. Working at the Division of Automatic Control has been a great experience. Thanks to all the nice colleagues and the curious students! Prof Svante Gunnarsson responsively manages the division creating a nice atmosphere, thanks! Our secretary Ninna Stensgård, and her two predecessors Åsa Karmelind and Ulla Salaneck make sure that administrative and practical matters work flawlessly. Thank you!

Thanks to the Industry Excellence Center LINK-SIC founded by Vinnova for financial support. A special thanks Dr Martin Enqvist for managing everything related to LINK-SIC smoothly.

Dr Zoran Sjanic, it's been great working together and I hope we can continue on this quest! Besides being a clever guy, you are also a good friend with a funky sense of humor that I enjoy. Thanks!

Dr Magnus Axholt, your super positive attitude was vital when we worked with HMD calibration. I also really enjoyed learning more about augmented reality and optical systems. I am really impressed by your entrepreneurial skills! Thanks!

Dr Jonas Callmer and I have shared many hours of studies and a bizarre amount of coffee whilst quoting South Park. Unfortunately we don't meet as often since you joined *the dark side* outside university. That will hopefully change when you sell SenionLab for a trillion balubas!

Dr Karl Granström –*master of beer induced discussions* – and I shared office for the first years of the PhD studies. Thanks for all the good times we have had at RT and outside work!

Dr David Törnqvist gave me a smooth start in the group when supervising my Master's Thesis. You have been my next-door neighbour in the corridor, fellow musician, dive mate and a good friend. Thanks David!

Thanks Dr Karl-Johan Thore for being an excellent study companion and friend.

I'm happy that you finally moved to the right side of the city!

Johan Reunanen, you inspired me to move to Linköping to study and I also joined your *vaniljekrem*-gang in Oslo. We got to know each other when studying a preparatory year of Natural Sciences in our hometown Sundsvall and I hope we get to meet more often.

Johan Bjugert experimentally proved that conserving eggs in the freezer is a bad idea! It was fun living together in Ryde on BG's second floor. I hope we get to meet more often now when you are back in Sweden. Do you still prepare an extra serving of Pasta Carbonara for your trumpet?

Thanks to my old friend Jens *struuut!* Södervall for all these years!

Thanks Per *Buffy* Keller for all the fun in Dublin and elsewhere!

Clas Veibäck, it's nice to have an office mate again. Especially with someone who is as eager as I to put on a Maalouf album on our blazing sound system!

Thanks to Henrik Tidefelt and Nina Fjällström for all the fun we have together! Congratulations to the marriage and for letting us be a part of your day!

These years have been accompanied by a lot of great music. Thanks to Miles Davis, Håkan Hardenberger, Ibrahim Maalouf, Wynton Marsalis, Esbjörn Svensson Trio, Goran Kajfeš, Wolfgang Amadeus and Leopold Mozart, Igor Stravinsky, Brad Mehldau, Al Di Meola, Avishai Cohen, Pat Metheny, Keith Jarrett and the great Ludwig van.

Over the years I have made many friends. Dr André Carvalho Bittencourt, Lic Sina Khoshfetrat Pakazad, Dr Patrik Axelsson, Lic Ylva Ljung, Dr Christian Lundquist, Dr Per Skoglar, Dr Daniel Peterson, Dr Christian Lyzell, Dr Johan Sjöberg, Dr Jeroen Hol, Dr Gustaf Hendby, Lic Manon Kok, Dr Emre Özkan, Dr Saikat Saha, Dr Daniel Axehill, Jonas Linder, Dr Fredrik Lindsten, Lic Michael Roth and possibly more people. Thank you for making my life much more enjoyable!!

Outside work I have had particularly many great laughs together with Dr Hanna Fager, Dr Oskar Leufvén, Dr Jonas Callmer and Dr Karl Granström. Thanks!

Thanks to the Falkenberg people for all the fun we have together! Thanks to my late mother Helena for all love, care and the somewhat useful stubbornness I inherited. My brother John and father Bernt, I hope you feel as excited and proud as I do! Thanks for everything!

Finally, thanks and a thousand kisses to Maria, Stella and the unborn *lill-räpen* for all your love and support!

Linköping, September 2014
Martin Skoglund

Contents

Notation

Abbreviations

Abbreviation	Meaning
BA	*bundle adjustment*
CRLB	*Cramér-Rao lower bound*
CCD	*charge coupled devices*
CMOS	*complementary metal oxide semiconductor*
DOF	*degrees of freedom*
DVL	*Doppler velocity log*
EKF	*extended Kalman filter*
EM	*expectation-maximisation*
E-RTS	*extended Rauch-Tung-Striebel smoother*
GNSS	*global navigations satellite system*
GPS	*global positioning system*
IMU	*inertial measurement unit*
IDP	*inverse depth parametrisation*
KF	*Kalman filter*
LiDAR	*light detection and ranging*
MAP	*maximum a posteriori*
MEMS	*micro electrical mechanical systems*
ML	*maximum likelihood*
NLS	*nonlinear least square*
RADAR	*radio detection and ranging*
RANSAC	*random sampling consensus*
ROV	*remotely operated vehicle*
SLAM	*simultaneous localisation and mapping*
SFM	*structure from motion*
SIFT	*scale-invariant feature transform*
SONAR	*sound navigation and ranging*
UAV	*unmanned aerial vehicle*
UUV	*unmanned underwater vehicle*

Symbols and Operators

Notation	Meaning
x_t	State vector
$x_{0:t}$	State trajectory up to time t
y_t	Measurement vector
$y_{1:N}$	Data batch of length N also denoted $y_{1:N} = Y = \{y_1, \ldots, y_N\}$
u_t	Input vector
θ	Parameter vector
$\hat{\theta}$	Estimate of θ
e_t	Measurement noise
w_t	Process noise noise
Q	Process noise variance
R	Measurement variance
$f(x_t)$	Process model
$h(x_t)$	Measurement model
$\mathcal{N}(\mu, \Sigma)$	Gaussian distribution with mean μ and variance Σ
$p(Y\|\theta), p_\theta(Y)$	Data likelihood
$p(\theta\|Y)$	Posterior density
R	Rotation matrix
ω	Angular velocity
$\boldsymbol{p}$	Position
$\boldsymbol{v}$	Velocity
$\boldsymbol{a}$	Acceleration
R^T	Transpose of matrix or vector
q	Unit quaternion $q = [q_0,\ q_1,\ q_3,\ q_4]^T$
det	Determinant
$\times$	Cross product
∇	Gradient operator
$\mathbb{R}$	Set of real numbers
$P([X, Y, Z]^T)$	Projection operator $P([X, Y, Z]^T) = [X/Z, Y/Z]^T$
$\|x\|_{P^{-1}} = \sqrt{x^T P^{-1} x}$	P-weighted norm of vector x
$SO(3)$	Special Orthogonal Group
E	Essential Matrix
$\arg\max_\theta$	Maximising argument with respect to θ
$\arg\min_\theta$	Minimising argument with respect to θ
$\sim$	Distributed according to. Or up to an unknown scale λ according to $a \sim b \iff a = \lambda b$.

Part I

Background

1 Introduction

This thesis is about navigation, mapping and modeling for mobile robotics. The specific applications studied are sensor network localisation, underwater vehicle modeling and batched *simultaneous localisation and mapping* (SLAM) initialisation and estimation methods for inertial sensors and monocular cameras.

This chapter introduces and motivates the research topics in this thesis. It also summarises the research contributions by the author of the appended publications and other publications.

1.1 Motivation

Autonomous systems are becoming increasingly important in today's society. The main drive for industrial automatisation is increased efficiency leading to a larger competitive advantage. In our homes we are enjoying the benefits of many automatised processes such as dishwashers, heat exchangers and robotic lawnmowers. We use these systems because it allows us to do things that are more rewarding, such as building autonomous systems. The design, construction and evaluation of these systems is an iterative process which may involve machines, computers, physical models, mathematical models and a fair bit of experience.

Today, autonomous systems are not only restricted to industries and our homes but are becoming integrated into our normal day lives. For instance, yesterday's fantasies about self-driving cars are now becoming reality[1]. Autonomous robotics see, Figure 1.1, is a special branch in mobile robotics in which the focus is to automatise as many subsystems of the robot as possible. A research-intensive

[1] `http://googleblog.blogspot.se/2014/04/the-latest-chapter-for-self-driving-car.html`

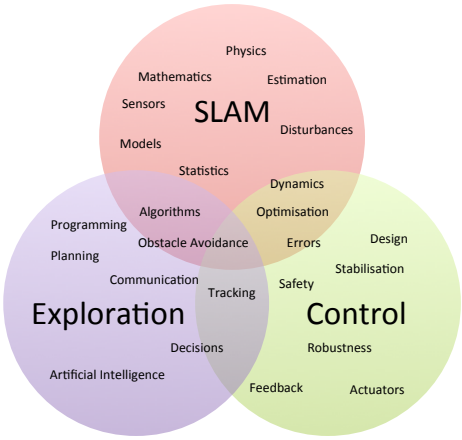


Figure 1.1: *Parts of autonomous robotic systems.*

Figure 1.2: *Left: Combined camera and inertial sensors in a single unit. Prototype made by XSens. Right: The Saab V-200 Skeldar helicopter. By courtesy of Saab AB.*

part of this subsystem is onboard sensor data interpretation which is a cornerstone for making truly autonomous robots without the need for external support systems. The processed sensor data is used for various purposes such as navigation, mapping, control and exploration.

SLAM emerged in the field of mobile robotics as a tool to enhance navigation and to infer properties of the surrounding environment by means of the onboard sensors. As the name suggests, SLAM is the joint problem of localisation and mapping which, on their own, are active research fields. In Figure 1.2 a multisensor unit and an unmanned autonomous vehicle are shown.

Localisation, or navigation, problems assume that the surrounding environment is, to some extent, known a priori. Such information can for instance be; topographic maps, radio beacons, star constellations, magnetic fields, *global naviga-*

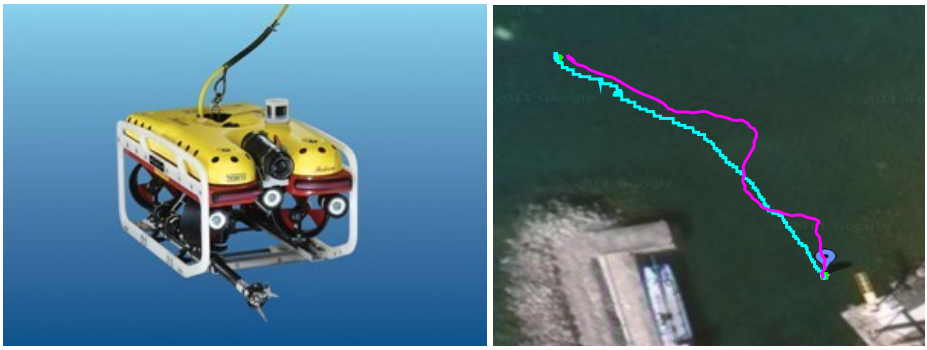


Figure 1.3: *Left: The Saab Seaeye Falcon ROV. By courtesy of Saab AB. Right: Position trajectory based on fusion of the ROV hydrodynamic model and magnetometer in magenta and the cyan curve marks the cable which the ROV roughly followed.*

tion satellite systems (GNSS) satellite ephemeris, which are integrated directly on moving platforms. If localisation of moving objects, i.e., tracking, is done using external systems such as; ground based radar, closed-circuit television, cellular networks, and other types of sensor networks, the location of the sensor nodes needs to be accurately known. In Figure 1.3 a navigation example using a *remotely operated vehicle* (ROV) is shown.

Mapping problems consider estimation of the surrounding environment with examples such as; topographic maps, bathymetric maps, magnetic fields, object shape. It is assumed that all necessary information about the moving platforms is available such that the uncertainty w.r.t., to the estimated maps is negligible. This also covers cases when moving platforms are measured by sensor networks and the node locations need to be calibrated. In Figure 1.4 a small experimental setup and the corresponding SLAM estimate are shown.

SLAM research initially spent a lot of effort on indoor platforms typically equipped with wheel encoders for odometry and line sweeping laser range scanners and these are still common. Today, this field of research is huge and heterogeneous with all kinds of platform and sensor combinations. A small table with some sensors and platforms that have been used in SLAM applications are shown in Table 1.1.

1.2 Contributions

Published and submitted publications which constitute the second part of this thesis are listed below in chronological order together with a description and the contribution of each paper.

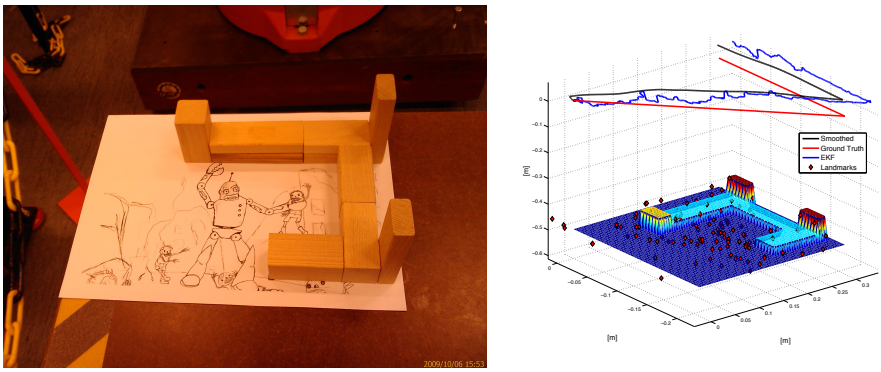


Figure 1.4: *Left: Image of an environment used for SLAM experiments using camera and inertial sensors. Right: SLAM results using EKF and (smoothed) nonlinear least-squares.*

Sensor\Platform	Flying	Ground	Pedestrian	Vessel	Submarine
Camera	**1**	**2**	**3**	**4**	**5**
Stereo Camera	**6**	**7**	**8**		**9**
RADAR	**10**	**11**		**12**	
SONAR		**13**		**14**	**15**
IMU	**16**	**17**	**18**	**19**	**20**
LiDAR	**21**	**22**		**23**	

Table 1.1: *Some SLAM publications using various platforms and sensors.* ***1*** *(Karlsson et al., 2008; Caballero et al., 2009; Bryson and Sukkarieh, 2009; Lupton and Sukkarieh, 2009, 2008; Bryson and Sukkarieh, 2008),* ***2*** *(Wang and Dissanayake, 2012; Thrun and Montemerlo, 2006; Callmer et al., 2008),* ***3*** *(Davison et al., 2007; Davison, 2003; Eade, 2008; Klein and Murray, 2007),* ***4*** *(Callmer et al., 2011),* ***5*** *(Kim, 2013),* ***6*** *(Jung and Lacroix, 2003),* ***7*** *(Konolige and Agrawal, 2008; Cummins and Newman, 2010),* ***8*** *(Lupton and Sukkarieh, 2012; Karlsson and Bjärkefur, 2010; Strasdat et al., 2011; Jung and Taylor, 2001),* ***9*** *(Eustice et al., 2006; Mahon et al., 2008),* ***10*** *(Sjanic and Gustafsson, 2010),* ***11*** *(Marck et al.; Gerossier et al., 2009), - (),* ***12*** *(Callmer et al., 2011; Mullane et al., 2010),* ***13*** *(Choi et al., 2005), - (),* ***14*** *(Wrobel, 2014),* ***15*** *(Newman et al., 2003; Ribas et al., 2006),* ***16*** *(Bryson et al., 2010),* ***17*** *(Wijesoma et al., 2006),* ***18*** *(Lupton and Sukkarieh, 2012),* ***19*** *(Han and Kim, 2013),* ***20*** *(Kim and Eustice, 2009),* ***21*** *(Fossel et al., 2013),* ***22*** *(Bosse and Zlot, 2008),* ***23*** *(Han and Kim, 2013).*

1.2.1 Included Contributions

Silent Localization of Underwater Sensors Using Magnetometers

> J. Callmer, M. Skoglund, and F. Gustafsson. Silent localization of underwater sensors using magnetometers. *EURASIP Journal on Advances in Signal Processing*, 2010, 2010. doi: 10.1155/2010/709318. URL `http://dx.doi.org/10.1155/2010/709318`. Article ID 709318.

Paper A presents a method for localisation of underwater sensors equipped with triaxial magnetometers using a friendly vessel with known magnetic characteristics.

Underwater sensor networks can be used to detect and track surface vessels but more importantly, underwater vessels which may pose threats in security critical maritime environments. For these networks to function properly the location of each sensor need to be accurately known which can be difficult to obtain in fast deployment scenarios. We here demonstrate how the sensor positions and the vessel trajectory can be estimated simultaneously in an *extended Kalman filter* (EKF) using only magnetometer measurements from the sensors. *Cramér-Rao lower bound* (CRLB) analysis shows the attainable node localisation accuracy for a given trajectory and the attainable tracking performance for a given network. The CRLB analysis could thus serve as a guide on sensor network design. Sensitivity analysis indicates that when using GNSS measurement of the vessel trajectory, errors in sensor orientation and magnetic dipole strength are suppressed and the localisation accuracy is enhanced.

The results are evaluated using simulated data. This is also the only publication in this thesis without real, experimental, data. This inexpensive solution to the difficult sensor localisation problem may also be used for re-localisation of sensors if conditions have changed.

This is joint work primarily between the first and the second author who produced the ideas, theory, implementation and most of the writing.

A Nonlinear Least Squares Approach to the SLAM Problem

> Z. Sjanic, M. A. Skoglund, F. Gustafsson, and T. B. Schön. A nonlinear least squares approach to the SLAM problem. In *Proceedings of the IFAC World Congress*, volume 18, Milan, Italy, 28-2 Aug./Sept. 2011.

Paper B presents an algorithm for visual/inertial SLAM based on the *maximum a posterior* (MAP) estimate of the whole 6 *degrees-of-freedom* (DOF) trajectory (including velocities) and 3D map, solved using *nonlinear least-squares* (NLS) optimisation.

An initial estimate is acquired using EKF-SLAM with inertial data as input to the motion model. *Scale invariant feature transform* (SIFT) image features are used for tracking and data association and the *inverse depth parametrisation* (IDP) is used for landmark parametrisation since it is known to handle depth uncertainty, due to EKF linearisation errors, better than using the direct 3D parametrisation.

The inherent sparsity structure of the full SLAM problem is efficiently utilised in the NLS solver and the result is a metrically correct position, orientation, velocity and landmark map estimate. The proposed algorithm is evaluated on experimental data using a sensor platform mounted on an industrial robot enabling accurate ground truth reference.

The results from this algorithm can, for instance, be used on unmanned aerial vehicles to compute detailed terrain maps necessary for safe landing or in underwater localisation and mapping and other GNSS denied environments.

This is joint work primarily between the first and the second author who produced the ideas, theory, experiments, implementation and most of the writing.

Parts of this work has also been used in Nilsson (2010); Nilsson et al. (2011) where the sensor is a forward looking *infrared* (IR) camera attached to a car, and in Karlsson and Bjärkefur (2010) where the authors use stereo camera and laser camera in an indoor environment.

Modeling and Sensor Fusion of a Remotely Operated Underwater Vehicle

> M. A. Skoglund, K. Jönsson, and F. Gustafsson. Modeling and sensor fusion of a remotely operated underwater vehicle. In *Proceedings of the 15th International Conference on Information Fusion (FUSION), Singapore, 9-12 July 2012*, pages 947–954. IEEE, 2012.

Paper C presents how a complex hydrodynamic model for a *remotely operated vehicle* (ROV) can be used to robustify navigation based on onboard sensors.

ROV's are small and relatively cheap *unmanned underwater vehicles* (UUV). They are used in situations such as: tunnel or hull inspections; deep sea missions beyond human diver depths; mine hunting and mine disposal. Due to their limited payload capacity, and lack of external underwater localisation support, the onboard navigation systems have to be robust. The paper shows how previously estimated model structures are put together in a large model describing the full 6-DOF motion of the ROV, including angular and linear velocities. The model potentially provides an independent source of vehicle speed and angular rate. We also provide models for all the onboard sensors, expressed in the body frame of the ROV, which are fused with the hydrodynamic model in an EKF. The results are based on data from the field tests performed in the Master's thesis work of Jönsson (2010) which the first author supervised. We show that, in particular, the vehicle speed can be accurately predicted as compared with the doppler speedometer.

This is joint work primarily between the first and the second author. The second author provided initial models and experiments. The first author did most of the implementation and most the writing.

Initialisation and Estimation Methods for Batch Optimisation of Inertial/Visual SLAM

> M. A. Skoglund, Z. Sjanic, and F. Gustafsson. Initialisation and estimation methods for batch optimisation of inertial/visual SLAM. Submitted to IEEE Transactions on Aerospace and Electronic Systems, June 2014.

Paper D presents a complete initialisation method for inertial/visual batch SLAM which can be used to obtain metrically correct map and navigation state estimates.

Batch formulations of inertial/visual SLAM are nonlinear and nonconvex problems which need a good initial estimate to converge to the global optimum. In Paper B the full SLAM problem was initialised using EKF-SLAM which is not a feasible solution for large data sets. In this paper we present a multi-step algorithm that solves a series of almost uncoupled problems. The combination of rotation estimation and appearance based data association using only vision together with visual/inertial methods leads to almost linear formulations which can be solved easily. The initialisation method is demonstrated on both simulated data and a small feasibility study on experimental data using an industrial robot, to get access to ground truth, is also performed.

This is joint work primarily between the first and the second author who produced the ideas, theory, experiments, implementation and most of the writing.

EM-SLAM with Inertial/Visual Applications

> Z. Sjanic, M. A. Skoglund, and F. Gustafsson. EM-SLAM with inertial/visual applications. Submitted to IEEE Transactions on Aerospace and Electronic Systems, June 2014.

Paper E presents an *expectation-maximisation* (EM) approach to a *maximum likelihood* (ML) batch formulation of inertial/visual SLAM.

The EM algorithm introduces a set of so-called latent, or hidden, variables. By doing so, the problem can be split into two, hopefully, simpler problems. The first problem is the expectation, the so-called E-step, with respect to the conditional density of the hidden variables. The second problem, known as the M-step, is to maximise the result obtained in the E-step with respect to the unknown parameters. These two problems are solved iteratively until some convergence criterion is met. The EM-SLAM algorithm proposed here considers the platform motion as hidden variables and the landmark map as the unknown parameters. The E-step is solved using an *extended Rauch-Tung-Striebel smoother* (E-RTS) with constant size state vector which becomes computationally cheap. The M-step is solved efficiently with a quasi-Newton method having fewer variables than the full NLS formulation in which both the state sequence and the map are seen as parameters in an ML fashion. EM-SLAM is compared with NLS-SLAM both in terms of performance and complexity. The proposed method is evaluated in real experiments and also in simulations on a platform with a monocular cam-

era attached to an inertial measurement unit. The initial estimate for both the parameters and the hidden variables are obtained using the method in Paper D. It is demonstrated to produce lower *root mean square error* (RMSE) than with a standard Levenberg-Marquardt solver of NLS problem, at a computational cost that increases considerably slower.

This is joint work primarily between the first and the second author who produced the ideas, theory, experiments, implementation and most of the writing.

1.2.2 Additional Publications

This section contains published work not included in this thesis.

1.2.3 Optical See-Through Head Mounted Display Calibration

Augmented reality (AR) is used to denote visual information overlaid, augmented, on human vision. *Optical see-through head mounted displays* (OSTHMD) are wearable AR systems, which means that the OSTHMD displayed graphics moves with the user, see Figure 1.5. To augment the real world with graphics in such a system three main problems need to be solved. First, the pose (position and orientation) of the OSTHMD with respect to the inertial frame needs to be known. Second, the calibration problem which constitutes the static transformations of the relative position and rotation of the semitransparent graphics screen with respect to the user's eye and an inertial frame. Finally, models of the environment where the graphics should be superimposed have to be created. In the two articles, the first two of the three problems are addressed. The pose of the user's head, and consequently the OSTHMD, is provided using a visual tracking system along with a visual landmark coordinate. The calibration problem is solved using the data from the tracking system and measurements acquired by the subject through a bore-sighting exercise. The user aligns graphics (a crosshair) displayed on the screen with a measured point (a diode) in the inertial frame, see Figure 1.5, and several such alignments are collected from different screen points and the subject's locations in order to excite the parameter space.

We adopt the theoretical framework for camera calibration founded in the computer vision and photogrammetry domains to OSTHMD calibration. The calibration problem itself is rather ill-posed since the measurements are few compared to the parameter space and the *signal-to-noise ratio* (SNR) is low. The work in these two publications reflects some labour intense engineering/research where several months were spent on the AR system which consists of several hardware components interacting through several software layers.

This is joint work primarily between the first and the second author who produced the ideas, theory, experiments and implementation. While the first author did most of the writing.

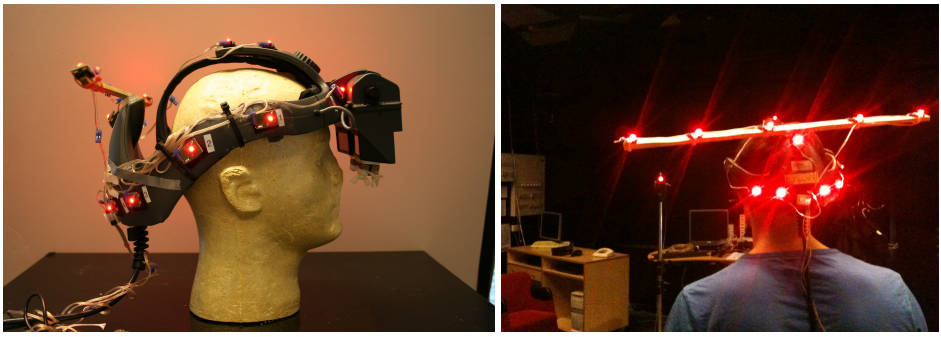


Figure 1.5: *Left: A Kaiser ProView 50ST Optical see-through head mounted display equipped with IR diodes for the PhaseSpace IMPULSE motion tracking system placed on a mannequin. Right: Example of a bore-sighting exercise with the reindeer-looking OSTHMD system. Note the diode in the far back of the room which the subject (myself) aims at.*

Optical See-Through Head Mounted Display Direct Linear Transformation Calibration Robustness in the Presence of User Alignment Noise

> M. Axholt, M. Skoglund, S. D. Peterson, M. D. Cooper, T. B. Schön, F. Gustafsson, A. Ynnerman, and S. R. Ellis. Optical see-through head mounted display direct linear transformation calibration robustness in the presence of user alignment noise. In *Proceedings of the 54th Annual Meeting of the Human Factors and Ergonomics Society*, volume 54, pages 2427–2431, San Francisco, CA, USA, 27-1 Sept./Oct. 2010. Human Factors and Ergonomics Society.

The abstract from the paper is included below.

The correct spatial registration between virtual and real objects in optical see-through augmented reality implies accurate estimates of the user's eyepoint relative to the location and orientation of the display surface. A common approach is to estimate the display parameters through a calibration procedure involving a subjective alignment exercise. Human postural sway and targeting precision contribute to imprecise alignments, which in turn adversely affect the display parameter estimation resulting in registration errors between virtual and real objects. The technique commonly used has its origin in computer vision, and calibrates stationary cameras using hundreds of correspondence points collected instantaneously in one video frame where precision is limited only by pixel quantization and image blur. Subsequently the input noise level is several order of magnitudes greater when a human operator manually collects correspondence points one by one. This paper investigates the effect of human alignment noise on view parameter estimation in an optical see-through head mounted display to determine how well astandard camera calibration method performs at greater noise levels than documented in computer vision literature. Through Monte-Carlo simulations we show that it is particularly difficult to estimate the user's eyepoint in depth, but

that a greater distribution of correspondence points in depth help mitigate the effects of human alignment noise

Parameter Estimation Variance of the Single Point Active Alignment Method in Optical See-Through Head Mounted Display Calibration

> M. Axholt, M. A. Skoglund, S. D. O'Connell, M. D. Cooper, S. R. Ellis, and A. Ynnerman. Parameter estimation variance of the single point active alignment method in optical see-through head mounted display calibration. In *Proceedings of the IEEE Virtual Reality Conference*, pages 27–34, Singapore, Republic of Singapore, Mar. 2011.

The abstract from the paper is included below.

The parameter estimation variance of the *single point active alignment method* (SPAAM) is studied through an experiment where 11 subjects are instructed to create alignments using an *optical see-through head mounted display* (OSTHMD) such that three separate correspondence point distributions are acquired. Modeling the OSTHMD and the subject's dominant eye as a pinhole camera, findings show that a correspondence point distribution well distributed along the user's line of sight yields less variant parameter estimates. The estimated eye point location is studied in particular detail. The findings of the experiment are complemented with simulated data which show that image plane orientation is sensitive to the number of correspondence points. The simulated data also illustrates some interesting properties on the numerical stability of the calibration problem as a function of alignment noise, number of correspondence points, and correspondence point distribution.

Insights from Implementing a System for Peer-Review

> C. Lundquist, M. A. Skoglund, K. Granström, and T. Glad. Insights from implementing a system for peer-review. *IEEE Transactions on Education*, 56(3):261–267, 2013.

Finally, the following paper about undergraduate teaching and peer-review has been published.

1.3 Thesis Outline

The thesis is divided into two parts, with background material in the first part and with edited versions of published papers in the second part. The first part consists of material introducing and explaining the background to the publications. The publications on their own include detailed background material.

The first part of the thesis is organised as follows. Chapter 2 presents model structures for motion, sensors and computer vision. Chapter 3 introduces the sensor fusion concept which is used for both filtering and smoothing. The connection to optimisation is also explained. Chapter 4 gives a brief overview of SLAM estimation methods and explains the concepts used in the publications. The first

part of the thesis ends with Chapter 5, which presents conclusions and discusses future work.

The second part of the thesis consist of edited versions of the papers. The papers are only edited to comply with the layout of the thesis template.

2
Models

In this chapter the most important model structures which are used in the publications in Part II are outlined. The models describe rigid body kinematics and dynamics, the pinhole camera, inertial measurements, magnetometers and three different landmark parametrisations for camera measurement models. This is the basis for inference in state-space, and other parametric, models which are described in the next chapter.

In a navigation context the state vector may include:

- Position $\boldsymbol{p}$, velocity $\boldsymbol{v}$, acceleration $\boldsymbol{a}$ and possibly higher order derivatives.
- Unit quaternion q parametrising orientation, angular velocity $\boldsymbol{\omega}$ and higher order derivatives.

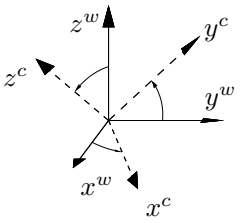
For localisation and tracking problems the state vector may include:

- Stationary point targets as landmark coordinates m.
- Stationary sensors $\boldsymbol{p}$ which may also include sensor orientation q and other quantities.
- Non-stationary targets possibly constrained to different motion models.
- Extended targets, stationary or non-stationary.
- Binary correspondence variables c relating measurements to target identities and possibly other metadata.

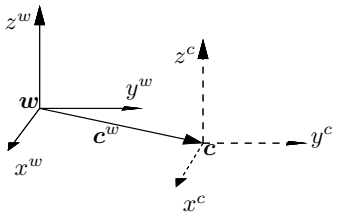
In case navigation and localisation are estimated jointly it is natural to include the appropriate quantities mentioned above in a large state vector.

2.1 Kinematics

Kinematics describes the motion of bodies without considering the forces causing the motion. Kinematic transformations between coordinate frames consist of length preserving translations and rotations as illustrated in Figure 2.1.



(a) *A rotation is a displacement of the axes while the origin is kept fix.*



(b) *A translation is a displacement of the origin while the axes are kept aligned.*

Figure 2.1: *Rigid body transformations.*

2.1.1 Translational Kinematics

A translation is a displacement of the body origin $\boldsymbol{p}$ while the axes are kept aligned. The translation is a vector, $\boldsymbol{p}^w$, from the origin of the frame w to the coordinate $\boldsymbol{p}$ expressed in the w-frame and it is also called the position. The translational motion equations can be derived from the time-derivative of a translation

$$\ddot{\boldsymbol{p}} = \dot{\boldsymbol{v}} = \boldsymbol{a}, \tag{2.1}$$

where $\boldsymbol{v}$ is the velocity and $\boldsymbol{a}$ is the acceleration. It is of course possible, and sometimes necessary, to introduce higher order terms such as $\dot{\boldsymbol{a}}$ which is known as jerk, but most often it can be considered to be noise i.e., $\dot{\boldsymbol{a}} = \boldsymbol{w}_a$. In (2.1) there is no indication of the dimension in which the motion occurs. This is however no restriction since the motion is independent in each dimension. The corresponding matrix form is

$$\begin{bmatrix} \dot{\boldsymbol{p}} \\ \dot{\boldsymbol{v}} \\ \dot{\boldsymbol{a}} \end{bmatrix} = \begin{bmatrix} 0 & I & 0 \\ 0 & 0 & I \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{v} \\ \boldsymbol{a} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ I \end{bmatrix} \boldsymbol{w}_a, \tag{2.2a}$$

$$\Longleftrightarrow$$

$$\dot{\boldsymbol{x}} = A\boldsymbol{x} + B\boldsymbol{w}_a, \tag{2.2b}$$

where the matrices 0 and I are of appropriate dimensions. Assuming that the input $\boldsymbol{w}_a$ is constant over the sampling interval, T, the ordinary differential equa-

tion (2.2) is solved as

$$x_{t+1} = e^{AT} x_t + \int_0^T e^{A\tau} \, \mathrm{d}\tau B \boldsymbol{w}_a, \tag{2.3}$$

which results in

$$\begin{bmatrix} \boldsymbol{p}_{t+1} \\ \boldsymbol{v}_{t+1} \\ \boldsymbol{a}_{t+1} \end{bmatrix} = \begin{bmatrix} I & TI & \frac{T^2}{2} I \\ 0 & I & TI \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \boldsymbol{p}_t \\ \boldsymbol{v}_t \\ \boldsymbol{a}_t \end{bmatrix} + \begin{bmatrix} \frac{T^3}{6} I \\ \frac{T^2}{2} I \\ I \end{bmatrix} \boldsymbol{w}_a. \tag{2.4}$$

2.1.2 Rotational Kinematics

Rotational kinematics is more complicated than its translational counterpart and this is due to the fact that rotation representations are nonlinear. This means that some ordinary operations defined on vector spaces, such as addition, are not defined here. Proper rotations are linear maps that preserve length and the handedness of the basis. There are many parametrisation alternatives to rotation matrices and perhaps the most common ones are Euler angles and the unit quaternion. For a thorough overview of rotation parameterisations see e.g., Shuster (1993). In this thesis, rotation in 3D is the typical case considered and as a common denominator for all parameterisations is that they can be transformed into a corresponding rotation matrix $\mathsf{R} \in SO(3)$.

Definition 2.1 (Special Orthogonal Group SO(3)). A matrix $\mathsf{R} \in \mathbb{R}^{3\times3}$ belongs to the special orthogonal group SO(3) if and only if it holds

$$\mathsf{R}^T \mathsf{R} = I, \tag{2.5a}$$

$$\det \mathsf{R} = 1. \tag{2.5b}$$

Figure 2.1a describes a rotation from the w frame to the c frame. This can be parametrised by the rotation matrix R^{cw}. The inverse rotation direction using (2.5a) is $(\mathsf{R}^{cw})^{-1} = (\mathsf{R}^{cw})^T$, hence $(\mathsf{R}^{cw})^T = \mathsf{R}^{wc}$.

Spatial rotations intuitively only have three degrees of freedom yet they still need at least five parameters (Hopf, 1940) in order to represent a global description see Stuelpnagel (1964) for a discussion. Rotations matrices are impractical for several reasons e.g., when rotations are estimated since it is difficult to design estimators directly on SO(3) and it is costly to use nine parameters. Luckily there are many alternatives for rotation parametrisation and an appealing option is the unit quaternion. Quaternions were invented by Hamilton (1844) as a tool to extend the imaginary numbers. The unit length quaternion is widely used in aerospace industry, mechanics, computer graphics, among others, since it allows a computationally efficient and singularity free rotation representation. The unit quaternion is a vector of real numbers $q \in \mathbb{R}^4$. A definition is given in Kuipers (2002, page 104), who splits the vector a scalar part, q_0, and a vector, $\boldsymbol{q} = e_1 q_1 +$

$e_2q_2 + e_3q_3$, where e_1, e_2, e_3 is the standard orthonormal basis in $\mathbb{R}^3$. With Kuipers definition the full vector representation is

$$q = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} q_0 \\ \boldsymbol{q} \end{bmatrix}. \tag{2.6}$$

Quaternions can be used to represent a rotation in $\mathbb{R}^3$ but in order to do so they must be constrained to the unit sphere $q \in S^3$, i.e., $q^T q = 1$, hence the name unit quaternion. Let

$$q = \begin{bmatrix} \cos\delta \\ \sin\delta \boldsymbol{n} \end{bmatrix} \tag{2.7}$$

which is a unit quaternion describing a rotation of an angle 2δ around the unit vector $\boldsymbol{n} \in \mathbb{R}^3$. Then a rotation using a unit quaternion is

$$\tilde{x}^b = q^{ba} \odot \tilde{x}^a \odot q^{ab} \tag{2.8}$$

where $\tilde{x}^\star = [0, x^\star]$ are the vectors' quaternion equivalents and $\odot$ denotes the quaternion multiplication, see e.g., Törnqvist (2008); Hol (2011). The unit quaternion can also be used to construct a rotation matrix, see for instance Shuster (1993, page 462), where the superscript ^{ab} is omitted for the sake of readability,

$$\mathsf{R}(q) = \begin{bmatrix} (q_0^2 + q_1^2 - q_2^2 - q_3^2) & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & (q_0^2 - q_1^2 + q_2^2 - q_3^2) & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & (q_0^2 - q_1^2 - q_2^2 + q_3^2) \end{bmatrix} = \tag{2.9}$$

$$= \begin{bmatrix} (2q_0^2 + 2q_1^2 - 1) & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) \\ 2(q_1q_2 - q_0q_3) & (2q_0^2 + 2q_2^2 - 1) & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & (2q_0^2 + 2q_3^2 - 1) \end{bmatrix}, \tag{2.10}$$

where the last equality is established using $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$. It should be noted that the unit quaternion is not a global parametrisation because the q and $-q$ describe the same rotation i.e., $\mathsf{R}(q) = \mathsf{R}(-q)$ which is easily verified from (2.9). This is seldom a practical problem since a solution with e.g., positive q_0 component can be chosen.

A local parametrisation which is popular within the computer vision community is the so-called exponential coordinates, see e.g., Ma et al. (2003). It uses a three-dimensional vector $\omega \in \mathbb{R}^3$ such that for any $\mathsf{R} \in SO(3)$ there exists $\mathsf{R} = e^{[\omega]_\times}$ where

$$[\omega]_\times = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \tag{2.11}$$

is the matrix form of the cross product. It is should be noted that any vector of the form $\omega + 2\pi k\omega$, where k being an integer, have the same rotation matrix which also means that for a given rotation matrix there are infinitely many exponential coordinates. The rotation matrix can be efficiently computed from the

exponential parameters using Rodrigues' formula

$$\mathsf{R} = e^{[\omega]_\times} = I + \frac{[\omega]_\times}{\|\omega\|}\sin(\|\omega\|) + \frac{[\omega]_\times^2}{\|\omega\|^2}\left(1 - \cos(\|\omega\|)\right), \tag{2.12}$$

which follows from the Taylor expansion of the exponential function, see Ma et al. (2003, Therorem 2.9) for a proof.

A differential equation for rotations can be established by using some properties of rotation matrices. Using the definition of SO(3) and taking the time derivative of both sides of (2.5a) as

$$\frac{\mathrm{d}}{\mathrm{d}t}(\mathsf{R}\mathsf{R}^T) = \frac{\mathrm{d}}{\mathrm{d}t}I, \quad \Longleftrightarrow \tag{2.13}$$

$$\dot{\mathsf{R}}\mathsf{R}^T + \mathsf{R}\dot{\mathsf{R}}^T = 0, \tag{2.14}$$

which implies that the matrix products are skew-symmetric

$$\dot{\mathsf{R}}\mathsf{R}^T = -(\dot{\mathsf{R}}\mathsf{R}^T)^T. \tag{2.15}$$

We can thus define the right hand side as

$$\dot{\mathsf{R}}\mathsf{R}^T = [\omega]_\times, \tag{2.16}$$

and since $R^T R = I$ we have that

$$\dot{\mathsf{R}} = [\omega]_\times \mathsf{R}. \tag{2.17}$$

By solving the differential equation (2.17) with ω constant

$$\mathsf{R}(t) = e^{[\omega]_\times t}\,\mathsf{R}(0). \tag{2.18}$$

the exponential coordinates are obtained. Similarly, the differential form of the unit quaternion parametrisation is given by

$$\dot{q}_t = \frac{1}{2}\underbrace{\begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix}}_{S(\omega)}\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \frac{1}{2}\underbrace{\begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix}}_{\widetilde{S}(q)}\begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix}, \tag{2.19}$$

which has the nice property of being bilinear. Note that due to the algebraic constraint the differential form of the unit quaternion is a *differential algebraic equation* (DAE). For a complete derivation of these relations see e.g., Törnqvist (2008); Hol (2011). The unit quaternion is thus solved as

$$\dot{q} = \frac{1}{2}S(\omega)q \Longrightarrow \tag{2.20}$$

$$q(t) = e^{\frac{1}{2}S(\omega)t}q(0) \tag{2.21}$$

where it is assumed that the angular velocity is constant. Defining the skew-symmetric matrix $S = \frac{1}{2}S(\omega)t$, $q_1 = q(t)$ and $q_0 = q(0)$ the quaternion length

remains unchanged

$$\|q_1\|_2^2 = \|e^S q_0\|_2^2 = q_0^T (e^S)^T e^S q_0 = q_0^T e^{S^T+S} q_0 = q_0^T q_0 = \|q_0\|_2^2, \tag{2.22}$$

since $S = -S^T$. The matrix exponential of the orientation dynamics can further be simplified

$$q_t = e^{\frac{1}{2}S(\omega)T_s} q_0 = \left(\cos\left(\frac{\|\omega\| T_s}{2} \right) I + \frac{\sin\left(\frac{\|\omega\| T_s}{2} \right)}{\|\omega\|} S(\omega) \right) q_0 \approx \left(I + \frac{T_s}{2} S(\omega) \right) q_0 \tag{2.23}$$

where the approximation is based on the small angle approximation which is good if the sampling rate T_s is high. It is now simple to introduce noise and bias as

$$q_t = \left(I + \frac{T_s}{2} S(\omega + \boldsymbol{b}_\omega + w_\omega) \right) q_0 = \left(I + \frac{T_s}{2} S(\omega + \boldsymbol{b}_\omega) \right) q_0 + \frac{T_s}{2} \widetilde{S}(q_0) w_\omega \tag{2.24}$$

using the relation (2.19). It is important to note that the nonlinear constraint $q^T q = 1$ need to be handled correctly in any estimator. In the case of filtering there are mainly two options. The first is to introduce a fictitious measurement $y = 1 - q^T q$ which should be zero and the second option is to normalise the updated quaternion as $q := q/\|q\|$ which is a projection onto the unit sphere.

Using unit quaternions for filtering may turn out problematic since the process noise covariance and the true state covariance is singular due to the over-parametrisation. Also, if a smoothed estimate is sought problems are unavoidable using unconstrained formulations, see Shuster (2003). Thus, there is room for improving the smoothing approach using unit quaternions as in Paper B.

A practical option when quaternions are simulated in continuous time is to introduce a feedback

$$\dot{q} = \frac{1}{2} S(\omega) q + \frac{\gamma}{2} (1 - q^T q) q, \tag{2.25}$$

which drives the quaternion to unit length, here γ is a proportional positive gain. This approach was used in the hydrodynamic model in Paper C.

2.1.3 Rigid Body Kinematics

Combining translational and rotational kinematics can be done in several ways and is very much a design task that depends on the system at hand. A straightforward model is given by the combination of the constant acceleration model and (2.2) and the unit quaternion (2.19) as

$$\begin{bmatrix} \dot{\boldsymbol{p}} \\ \dot{\boldsymbol{v}} \\ \dot{\boldsymbol{a}} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 & I & 0 & 0 \\ 0 & 0 & I & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} S(\omega) \end{bmatrix} \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{v} \\ \boldsymbol{a} \\ q \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ I \\ 0 \end{bmatrix} \boldsymbol{w}_a, \tag{2.26}$$

and it is also quite common to extend the model with states for angular acceleration, and possibly bias states for gyroscopes and accelerometers and this is done in Paper D.

Another model is obtained by considering acceleration and angular rates to be input to the system

$$\begin{bmatrix} \dot{\boldsymbol{p}} \\ \dot{\boldsymbol{v}} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} 0 & I & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{p} \\ \boldsymbol{v} \\ q \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ I & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \boldsymbol{a} + \boldsymbol{b}_a \\ \frac{1}{2} S(\omega + \boldsymbol{b}_\omega) q \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ I & 0 \\ 0 & \frac{1}{2} \widetilde{S}(q) \end{bmatrix} \begin{bmatrix} \boldsymbol{w}_a \\ \boldsymbol{w}_\omega \end{bmatrix}. \tag{2.27}$$

This significantly reduces the state space at the cost of limited noise modeling and fault detection capabilities. It should be noted that the biases are not observable using only accelerometers and gyroscopes in here, but must be inferred from other observations, e.g., camera measurements. Models with inertial sensors as input signals are used in Paper B and Paper E. In Paper A the kinematics of a vessel is described by a coordinated turn model which essentially constrains the object to follow a circular path. Such models are rather flexible and may be used to simulate more advanced models. It may be convenient to express some parts of a motion model in the body fixed frame since forces and torques are often naturally represented in this frame. This is the case in Paper C when modelling of a ROV is considered since the expressions for linear and rotational velocities due to external forces become much simpler.

2.2 Rigid Body Dynamics

Rigid body dynamics in classical mechanics investigates the motion of objects as caused by forces and torques. The fundamental equations describing these relations are given by Euler's axioms of motion. The first axiom is

$$\left(\frac{d\,(m\dot{p})}{dt} \right)_i = \sum_{j=1}^{N} F_j, \tag{2.28}$$

which gives the relation of a body's acceleration $\ddot{p}$ in an inertial frame i with mass m which is due to external forces F_j and p is the centre of mass. This is the straightforward extension of Newton's second law of motion for particles to rigid bodies expressed as conservation of linear momentum. The corresponding law for angular momentum, known as Euler's second law, states that the change of angular momentum of the body is equal to all external torques about the origin

$$\left(\frac{d\,(J\omega)}{dt} \right)_i = \sum_{j=1}^{N} r_j \times F_j, \tag{2.29}$$

where J is the inertia matrix, ω is the angular velocity of the rigid body, r_j are vectors that point from the center of rotation to the points where the external forces F_j are applied. It is useful to express Euler's laws in body referenced velocities since the mass and inertia are then constant. This can be done considering the derivative of vectors in a rotating reference frame, b, using the relation

$$\left(\frac{dr}{dt} \right)_i = \left(\frac{dr}{dt} \right)_b + \omega \times r. \tag{2.30}$$

in which the body referenced derivative is the same if the inertial frame is not rotating, i.e., $\omega = 0$. Applying (2.30) to (2.28) results in

$$m\ddot{p}^b = \sum_{j=1}^{N} F_j - \omega \times m\dot{p}^b, \tag{2.31}$$

where it is assumed that the mass is constant. Similarly, the expression for angular velocity (2.29) is

$$J\dot{\omega}^b = \sum_{j=1}^{N} r_j \times F_j - \omega^b \times J\omega^b. \tag{2.32}$$

For a complete derivation, see e.g., Fossen (2011).

2.3 Inertial Sensors

Inertial sensors measure specific force and angular velocity. These have great importance for many navigation systems since they provide an independent source for computing the position and orientation of a moving platform relative to some initial pose by integrating the accelerations twice and the angular velocity once, respectively. It may sometimes be the only means of navigation when other systems are unavailable. For instance, GNSS are prohibitive in many places such as underwater environments or indoors, and are also subject to jamming and spoofing. Magnetometers are passive sensors that can be used to calculate the orientation w.r.t., the local magnetic field. However, this field is rather weak and easily disturbed by ferrous objects. Inertial sensors are not subject to such external errors but rather errors due to the internal workings of the sensors. These contribute to the integration drift which means that the true pose of the sensor will deteriorate when the measurements are integrated. This drift is roughly inversely proportional to the price of the sensors. In this work we will only consider *micro electrical mechanical systems* (MEMS) type IMUs due to their relatively low cost and small size. There is a rich body of literature on navigation in general and modeling of IMU's in particular, see for instance Titterton and Weston (1997); Britting (1971). In all applications here the IMU's are rigidly mounted to the moving platform, in a so-called strap-down configuration, and hence sensor readings are naturally referenced in this frame.

2.3.1 Gyroscopes

Commonly, MEMS gyroscopes are based on measuring the rotation induced Coriolis force acting on a vibrating structure. A gyroscope mounted on a moving body, outputs gyroscope signals

$$y_{\omega_t} = \omega_t + e_{\omega_t}, \tag{2.33}$$

where ω_t is the angular velocity of the body w.r.t., an inertial frame and e_{ω_t} is noise. Due to unmodeled effects in the sensors, a bias, b_{ω_t}, could be added

to (2.33) giving

$$y_{\omega_t} = \omega_t + b_{\omega_t} + e_{\omega_t}, \tag{2.34}$$

and b_{ω_t} could also be included in the state vector with slowly varying dynamics. It is here assumed that the local, flat, navigation frame is inertial since otherwise (2.33) has to include another two components for the earth rotation rate and the turn rate of the navigation frame w.r.t., to earth. This assumption is good only if navigation is not performed over large distances over earth and that the earth rotation rate, $\omega^{ei} = \frac{2\pi}{24}\text{h}^{-1} \approx 7.27 \cdot 10^{-5}\text{rad/s}$, is small compared to the sensor noise. Furthermore, sensor alignment, orthogonality of sensor axes, sensor gain and other parameters are assumed known and accounted for by a factory calibration.

2.3.2 Accelerometers

Accelerometers using MEMS technology are based upon measuring deflections of a cantilever beam to which a so-called proof mass is attached. The mass deflects the beam when external acceleration along the sensitivity axis is applied to the sensor. Accelerometers do not only measure acceleration when used on earth, since they are subject to the gravitational field. Hence, accelerometers measure the specific force which is a sum of the free acceleration and the gravity field. A measurement model is then

$$y_a = a_t^b - g^b + e_{a_t} = \mathsf{R}_t^{be}\left(a_t^e - g^e\right) + e_{a_t}, \tag{2.35}$$

where a_t^e is the acceleration in the navigation frame g^e is the local gravity vector and e_{a_t} is measurement noise. Since the local navigation frame is considered to be inertial, Coriolis and centripetal acceleration can be neglected. It is clear that the measured specific force depends on the attitude R^{be} of the platform and as a consequence, errors in attitude will introduce so-called gravity leakage into the measurements. In case the body frame does not coincide with the sensor a Coriolis term $\omega \times \omega \times r^{IMUb}$ have to be added to (2.35) as done in Paper C, where r^{IMUb} is the offset vector, and the result must also be rotated by the relative rotation between these two systems. In the papers considering IMU and monocular vision the body center is in the origin of the IMU thus the offset and rotation between the frames are needed in the camera measurement equation.

2.4 Magnetometers

A simple magnetometer measurements model is

$$y_m = \mathsf{R}_t^{be}\, m^e + e_{m_t}, \tag{2.36}$$

where m^e is the local earth magnetic field vector and e_{m_t} is the measurement noise. This model can be used for computing the magnetic north if m^e is known. In Paper A magnetometers are used slightly differently. Here the objective is to track a magnetic object in order to determine locations of the sensor-nodes containing magnetometers in a sensor network. In this case the object is a vessel

with known magnetic signature and it is modelled as a single dipole $\boldsymbol{m} = \mathrm{R}^{eb}\boldsymbol{m}^b$ giving rise to the magnetic flux density

$$y_{\boldsymbol{m}} = \frac{\mu_0}{4\pi|\boldsymbol{r}|^5}(3\boldsymbol{r}(\boldsymbol{m}^T\boldsymbol{r}) - |\boldsymbol{r}|^2\boldsymbol{m}), \tag{2.37}$$

where μ_0 is the permeability of the medium and $\boldsymbol{r}$ is the distance from the sensor to the vessel. The magnetic flux due to the dipole strength decays cubically with the distance. It is here assumed that the effect of the local magnetic field is measured and accounted for before the vessel enters the survey area. For a thorough description on tracking of metallic objects Wahlström (2013) is recommended.

2.5 Vision Sensors

Digital vision sensors capture incident rays of light from the scene onto a sensor array through an optical system. Light reflected by objects in a scene can be mathematically described by *bidirectional reflectance distribution functions* (BRDF). The physical nature of light and image formation is a complex topic which is well beyond the scope of this thesis. There are mainly two manufacturing techniques for digital vision sensors. The first is semiconductor *charge coupled device* (CCD) in which each analog photo sensor (pixel) produces an electrical charge from the incoming light. The second sensor is *complementary metal oxide semiconductor* (CMOS) which directly converts light into voltage. CCD sensors are usually of global shutter type meaning that image is sampled at one time instant. In contrast, CMOS sensors usually sample row by row in a so-called rolling shutter fashion. Rolling shutters are far more complicated to use in geometrical vision problems when either the camera or scene is moving and we will therefore only consider global shutter cameras.

2.5.1 The Pinhole Camera

The by far most commonly used camera model is the pinhole camera. It is a mathematical model describing how points in $\mathbb{R}^3$ relate to points in $\mathbb{R}^2$ on the image plane through a central projection. The model has five intrinsic, or internal, parameters, and for most systems it is an approximation. Additionally, the position and orientation of the camera centre with respect to some other reference frame is also needed if the camera is part of a system, e.g., a stereo rig. Furthermore, most lenses introduce artifacts such as radial and tangential distortion which are important to compensate for.

Figure 2.2 illustrates a frontal pinhole projection. In the pinhole camera model the coordinate frames are:

- c – Camera frame with the *optical centre* as the origin where its z – axis coincides with the *optical axis*, also known as the *principal axis*.
- w – World frame.
- i – Image frame. The *image plane* is orthogonal to the *optical axis* and has

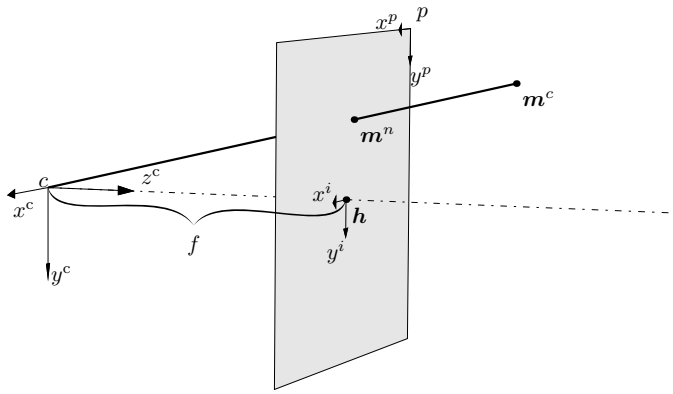


Figure 2.2: *Pinhole projection with image the plane placed in front of the camera centre.*

the *principal point* $\boldsymbol{h}$ as its origin. The distance between the c and the i frame is called the *focal length* f.

- n – Normalised image frame.
- p – Pixel frame with center in the upper left corner as viewed from the optical centre c.

A point $\boldsymbol{m}^c = [x^c, y^c, z^c]^T$ in the c-frame relates to a point $\boldsymbol{m}^n = [x^n, y^n]^T$ in the n-frame as

$$\begin{bmatrix} x^n \\ y^n \end{bmatrix} = \frac{f}{z^c} \begin{bmatrix} x^c \\ y^c \end{bmatrix}. \tag{2.38}$$

It is convenient to write (2.38) as a linear system which can be done by appending unit elements to the vectors giving the homogeneous coordinate representation

$$\begin{bmatrix} x^n \\ y^n \\ 1 \end{bmatrix} \propto z^c \begin{bmatrix} x^n \\ y^n \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\Pi_0} \begin{bmatrix} x^c \\ y^c \\ z^c \\ 1 \end{bmatrix}. \tag{2.39}$$

With focal length $f = 1$ the normalised image frame n and the image frame i coincides. The projection (2.39) can be expressed as

$$\lambda \boldsymbol{m}^n = \Pi_0 \boldsymbol{m}^c, \tag{2.40}$$

where $\lambda \in \mathbb{R}_+$ is an arbitrary scale factor. It is also convenient to define the projection as a function $P([X, Y, Z]^T) = [X/Z, Y/Z]^T$ which is as a map in Euclidean space $P : \mathbb{R}^3 \to \mathbb{R}^2$. A projection model which is suitable for omnidirectional cameras is the spherical perspective projection which has a scale described by $\lambda = \sqrt{X^2 + Y^2 + Z^2}$, whereas in a planar perspective projection $\lambda = Z$. This means that same equations can be used for both cases and only the depth is

described differently. More advanced models for omnidirectional cameras are accounted for in Scaramuzza et al. (2006)

The digitalisalisation of the image plane i is the pixel plane p

$$\begin{bmatrix} x^p \\ y^p \end{bmatrix} = \begin{bmatrix} f_x & s_\alpha \\ 0 & f_y \end{bmatrix} \begin{bmatrix} x^i \\ y^i \end{bmatrix} + \begin{bmatrix} h_x x^i \\ h_y y^i \end{bmatrix}, \tag{2.41}$$

or in homogeneous coordinates

$$\begin{bmatrix} x^p \\ y^p \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} f_x & s_\alpha & h_x \\ 0 & f_y & h_y \\ 0 & 0 & 1 \end{bmatrix}}_{K} \begin{bmatrix} x^i \\ y^i \\ 1 \end{bmatrix}, \tag{2.42}$$

where K is referred to as the *intrinsic parameter matrix* or simply the *calibration matrix*. The parameters of the calibration matrix are the focal lengths in pixel units $f_x = s_x/f$ and $f_y = s_y/f$ where f is the focal length expressed in meters per pixel and s_x, s_y are the pixel sizes in metric units. The center of the image plane $\boldsymbol{h} = [h_x \; h_y]^T$, is the principal point coordinate which is shifted w.r.t., to the pixel frame which has its origin in the upper left corner, see Figure 2.2. The location of the optical center is not needed for pure computer vision problems but it is important when vision sensor are combined with, for instance, inertial sensors. The skew parameter $s_\alpha = f_x \tan \alpha$ can safely be assumed $s_\alpha \approx 0$ (Hartley and Zisserman, 2004) in most cameras.

Cameras, especially in the lower price-range, suffer from non-negligible distortion due to the optical lenses involved and the mounting of the sensor. This can to some extent be compensated for by tangential-, radial- and pincushion distortion models (Ma et al., 2003). Camera matrices and distortion models can be estimated in standard camera calibration software, see e.g., Bouguet (2010); Zhang (2000).

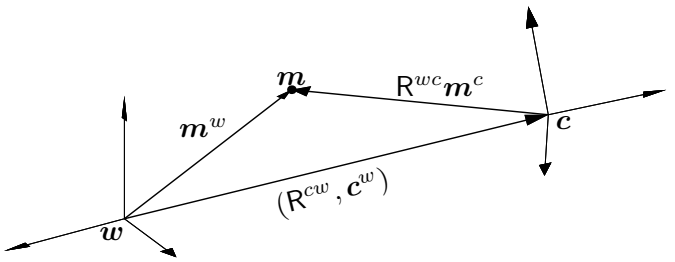


Figure 2.3: *Camera to world transformation.*

Points in another frame, say w, can be expressed in the camera frame c. In the pinhole camera model, such transformation is called extrinsic since it does not depend on the intrinsic camera calibration matrix K. Figure 2.3 describes the relation between a point $\boldsymbol{m}^w$ in world coordinates w expressed in camera coordi-

nates c

$$\boldsymbol{m}^c = \mathsf{R}^{cw}(\boldsymbol{m}^w - \boldsymbol{c}^w) = \mathsf{R}^{cw}\boldsymbol{m}^w - \mathsf{R}^{cw}\boldsymbol{c}^w, \tag{2.43}$$

which can be written in homogeneous coordinates as

$$\begin{bmatrix} x^c \\ y^c \\ z^c \\ 1 \end{bmatrix} = \begin{bmatrix} \mathsf{R}^{cw} & -\mathsf{R}^{cw}\boldsymbol{c}^w \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} x^w \\ y^w \\ z^w \\ 1 \end{bmatrix}. \tag{2.44}$$

Combining the extrinsic end intrinsic parameters, coordinates in the world frame can be expressed in pixel coordinates as

$$\begin{bmatrix} x^p \\ y^p \\ 1 \end{bmatrix} \propto z^c \begin{bmatrix} x^p \\ y^p \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & s_\alpha & h_x \\ 0 & f_y & h_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathsf{R}^{cw} & -\mathsf{R}^{cw}\boldsymbol{c}^w \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} x^w \\ y^w \\ z^w \\ 1 \end{bmatrix}. \tag{2.45}$$

In compact notation (2.45) can be written as

$$\boldsymbol{x}^p = P\boldsymbol{x}^w, \tag{2.46}$$

and the matrix P is often just called the camera. In the problems studied the calibration matrix and distortion models are known which allows to working directly on normalised camera measurements $\boldsymbol{m}^n$ and thus the pinhole camera works as a projective map in Euclidean space.

2.6 Camera Measurement Models

Parametrisation of camera measurements can be done in several ways and the most suitable choice typically depends on the application at hand. For instance, in a filtering context with explicit landmark coordinates included in the state vector measurement models are naturally expressed using the state. Geometrical properties, such as the epipolar constraint, can be used to define implicit measurement models. In this section three different parameterisations that are used in the publications are described.

2.6.1 Direct Parametrisation

From the rigid body transformation in (2.43), the projection of a single measurement of a landmark $\boldsymbol{m}^w$ onto the normalised image plane is on the form

$$y^m = P(\boldsymbol{m}_t^c) + e_t^m = P(\mathsf{R}_t^{cw}(\boldsymbol{m}^w - \boldsymbol{c}_t^w)) + e_t^m \tag{2.47}$$

where $\boldsymbol{c}_t^w$ is the position of the moving camera in the world frame and e_t^m is the so-called re-projection error. This is the preferred error being minimised since it has a sound interpretation and it is straightforward to include intrinsic and extrinsic parameters. We call this the direct parametrisation and it generalises

to n landmark measurements as

$$\begin{bmatrix} y^{m_1} \\ \vdots \\ y^{m_n} \end{bmatrix} = \begin{bmatrix} P(\mathrm{R}_t^{cw}(\boldsymbol{m}_1^w - \boldsymbol{c}_t^w)) \\ \vdots \\ P(\mathrm{R}_t^{cw}(\boldsymbol{m}_n^w - \boldsymbol{c}_t^w)) \end{bmatrix} + e_t^{m_{1:n}}, \tag{2.48}$$

which is the common situation in most practical applications. Depending on which parameters are sought the model can be used in different ways. If, for instance, only landmarks are unknown, then the problem is referred to as mapping or reconstruction and if only the camera pose is unknown the corresponding problem is called localisation or navigation.

2.6.2 Inverse Depth

In (2.47) the measured world point is represented naturally by its Euclidean coordinate. This may however be a poor choice if some points are at a much greater distance from the camera than other points in the scene. This is because the re-projection error cost function becomes very flat in the XYZ-space (Torr and Zisserman, 2000) since the XY coordinates are divided by a large number. This is partly the motivation for using so-called key-frames where only images with sufficient baseline are used for triangulation, see e.g., (Nistér, 2000) for a discussion. The relation to stereo cameras is that range resolution is inversely proportional to the baseline between the cameras and the corresponding change in disparity (image difference for a stereo pair). This phenomena have severe effect in the context of filtering since linearisation errors will be large and may cause the filter to diverge. Bryson et al. (2010) tackles this by tracking landmark coordinates to find the two views with maximal angular separation, parallax, for each landmark and then triangulate the landmark depth based on these two observations alone, known as delayed initialisation. A slightly different approach was proposed in Montiel and Davison (2006) who introduced the Inverse Depth Parametrisation (IDP). It uses six parameters where the first three are the coordinate of the camera from which the landmark was first observed $\boldsymbol{c}_t^w$ and the three parameters describing the vector from the camera to the landmark encoded by two angles φ^w, θ^w and the inverse depth ρ^w

$$\boldsymbol{m}^w = \boldsymbol{c}_t^w + \frac{1}{\rho^w}\boldsymbol{d}(\varphi^w, \theta^w), \tag{2.49a}$$

$$\boldsymbol{d}(\varphi^w, \theta^w) = \begin{bmatrix} \cos\varphi^w \sin\theta^w \\ \sin\varphi^w \sin\theta^w \\ \cos\theta^w \end{bmatrix}. \tag{2.49b}$$

The angles are computed from the normalized pixel coordinates as

$$\boldsymbol{g}^w = \mathrm{R}_t^{wc}\,\boldsymbol{m}^n, \tag{2.50}$$

$$\varphi^w = \arctan2(g_y^w, g_x^w), \tag{2.51}$$

$$\theta^w = \arctan2\left(\|[g_x^w \;\; g_y^w]^T\|_2, g_z^w\right), \tag{2.52}$$

where arctan2 is the four-quadrant arctangent function and the inverse depth is initiated with an educated guess. IDP has a small linearisation error even for large uncertainty in depth and it is easy to represent the range of depth uncertainty including infinity in a confidence region. Also, using IDP delayed landmark initialisation is no longer necessary. Obviously more states are needed, however, as soon as depth uncertainty is small enough the IDP landmarks can be converted into standard Euclidean coordinates. The corresponding measurement equation at time t for a landmark initiated at time j in the camera frame is

$$\boldsymbol{m}_t^c = \mathsf{R}_t^{cw}\left(\rho_t^w\left(\boldsymbol{c}_t^w - \boldsymbol{c}_j^w\right) + \boldsymbol{d}(\varphi_t^w, \theta_t^w)\right), \tag{2.53a}$$

$$y_t^c = P(\boldsymbol{m}_t^c). \tag{2.53b}$$

The IDP model, including sensor frame offset, was used for fusion with IMU measurements in Paper B. The same sensor setup was used in (Pinies et al., 2007) showing that feature initialisation and prediction in difficult cases, such as forward motion, can be handled better using IDP with support of IMU.

2.6.3 Epipolar Geometry

An important concept in computer vision is how to relate the relative pose of two cameras through point observations. This is called the epipolar constraint and is usually credited to the publication of Longuet-Higgins (1981) and is described in most books on computer vision, see e.g., Ma et al. (2003) or Hartley and Zisserman (2004). Given two images acquired from different vantage points and let R and $\boldsymbol{c}$ denote the rotation and translation from camera from the first to the second camera, respectively. Without loss of generality the first camera is located at the origin with no rotation. Thus, a world coordinate expressed in the first camera is simply $\boldsymbol{m}^w = \boldsymbol{m}_1^c$. The same point in the second camera is then related by a rigid body transformation

$$\boldsymbol{m}_2^c = \mathsf{R}\,\boldsymbol{m}_1^c + \boldsymbol{c}. \tag{2.54}$$

Since camera coordinates are related to homogeneous image coordinates through their unknown depths λ as $\boldsymbol{m}^c = \lambda \boldsymbol{m}^n$ we have

$$\lambda_2 \boldsymbol{m}_2^n = \mathsf{R}\,\lambda_1 \boldsymbol{m}_1^n + \boldsymbol{c}. \tag{2.55}$$

The intuition is that the three vectors connecting the two cameras and the point all lie in the same plane and therefore the triple product of the vectors is zero. The triple product is the inner product of one vector with the cross product of the other two. In this case it can be constructed by first eliminating the translation from the right hand side of (2.55) using

$$\boldsymbol{c} \times \boldsymbol{c} = 0 \tag{2.56}$$

from left. Then multiply from left with $\left(\boldsymbol{m}_2^n\right)^T$ to construct the two triple products

$$(\boldsymbol{m}_2^n)^T\, \boldsymbol{c} \times \lambda_2 \boldsymbol{m}_2^n = (\boldsymbol{m}_2^n)^T\, \boldsymbol{c} \times \mathsf{R}\,\lambda_1 \boldsymbol{m}_1^n. \tag{2.57}$$

in which the left hand side is zero since $\boldsymbol{c} \times \lambda_2 \boldsymbol{m}_2^n$ is perpendicular to $\boldsymbol{m}_2^n$. Now the epipolar (or bilinear) constraint, is obtained

$$(\boldsymbol{m}_2^n)^T \, \boldsymbol{c} \times \mathsf{R} \, \lambda_1 \boldsymbol{m}_1^n = 0, \implies (\boldsymbol{m}_2^n)^T \, \boldsymbol{c} \times \mathsf{R} \, \boldsymbol{m}_1^n = 0, \tag{2.58}$$

since λ_1 is non-zero. The product of the translation and the rotation is called the essential matrix

$$E = \boldsymbol{c} \times \mathsf{R} = [\boldsymbol{c}]_\times \mathsf{R} \tag{2.59}$$

where $[\boldsymbol{c}]_\times$ is the matrix form of the cross product

$$[\boldsymbol{c}]_\times = \begin{bmatrix} 0 & -c_z & c_y \\ c_z & 0 & -c_x \\ -c_y & c_x & 0 \end{bmatrix}. \tag{2.60}$$

This means that each correspondence gives an equation

$$\boldsymbol{m}_2^T E \boldsymbol{m}_1 = 0, \tag{2.61}$$

where the n superscript has been dropped for notational convenience. Note that E is homogeneous meaning that it is only defined up to an unknown scale. It has 5 or 6 degrees of freedom, depending on how it is constructed. If the translation is only known up to an unknown positive scale, i.e., it is also a homogeneous quantity, then E will have 5 DOF. This is typically the case when E is estimated from image data since the absolute scale of the scene is unknown without other information. If the scaling is known then E have 6 DOF. In case the calibration of the cameras are unknown the normalised image coordinates will also be unknown and the epipolar constraint then encodes the fundamental matrix $F = K_2^{-T} E K_1^{-1}$ where K_1 and K_2 denote the calibration of the cameras at the two views. The epipolar constraint is a generic property which may hold for any two vantage points if there are correspondences. It can therefore be used to compactly represent information in terms of a rigid body transformation up to a scale of the translation.

Epipolar geometry can also be used for simplifying correspondence search and verification. If the relative pose is known, a point in the first frame corresponds to a line in the other one, and vice versa, according to

$$\boldsymbol{l}_2 \sim E \boldsymbol{m}_1, \tag{2.62a}$$

$$\boldsymbol{l}_1 \sim E^T \boldsymbol{m}_2, \tag{2.62b}$$

where $\sim$ account for the unknown scale. This also highlights that cameras are bearing sensors. Thus, one may search for the correspondence along a line in the other image, see Figure 2.4. With uncertainty in the relative pose a band is the typical search region. Correspondence candidates may also be deemed correct or false by evaluating a cost defined by the point's distance to the line $\|\boldsymbol{l}_2^T \boldsymbol{m}_2\|$ where $\boldsymbol{l}_2$ is normalised. This is also the basis for efficient correspondence search using stereo cameras.

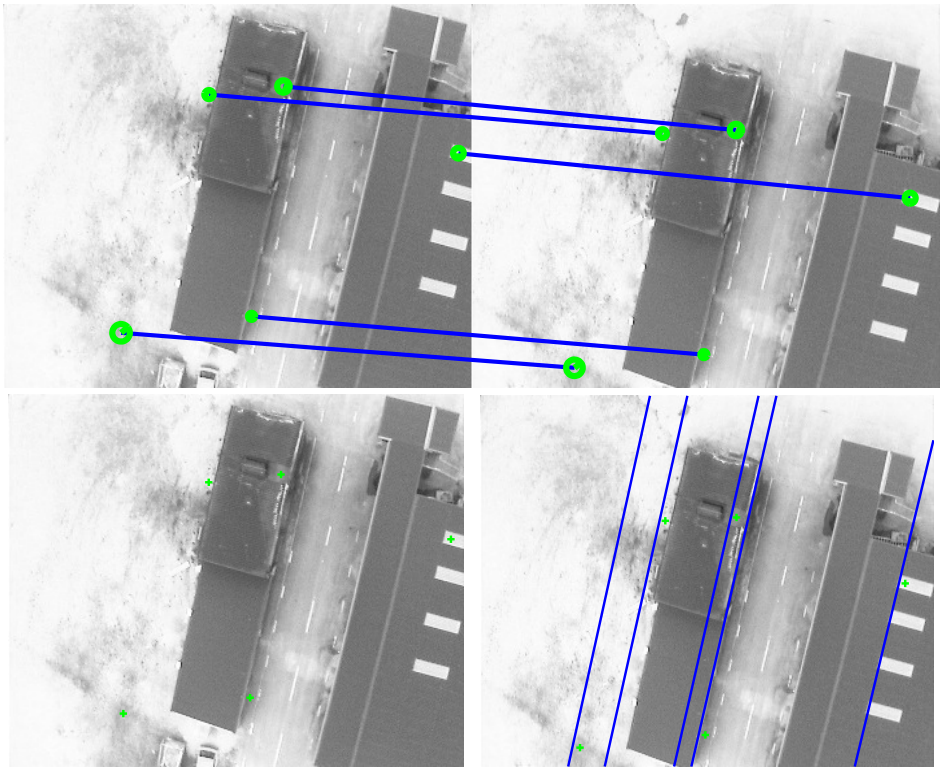


Figure 2.4: *The two top figures show five of the SIFT matches between the two frames which are were used to compute the essential matrix. The bottom left image shows the matchs in the first frame and the bottom right figure epipolar lines with matches in the second frame.*

Essential Matrix Estimation

Essential matrix estimation is used to find the relative pose between cameras from image correspondences such that the epipolar constraint is (2.61) is satisfied for all correspondences. The linear eight-point algorithm, see e.g., (Ma et al., 2003)[p. 121] is a simple essential matrix estimation algorithm. The algorithm computes an estimate of the essential matrix from a minimum of eight correspondences by computing a singular value decomposition (SVD). This starts from noting that the epipolar constraint can be transformed

$$\boldsymbol{m}_2^T E \boldsymbol{m}_1 = 0 \iff (\boldsymbol{m}_1 \otimes \boldsymbol{m}_2)^T E^s = 0 \tag{2.63}$$

where $E^s \in \mathbb{R}^{9\times 1}$ is the stacked matrix of column vectors of E and $\otimes$ denotes the Kronecker product. Using n correspondences results in

$$\underbrace{\begin{bmatrix} \left(\boldsymbol{m}_1^1 \otimes \boldsymbol{m}_2^1\right)^T \\ \vdots \\ \left(\boldsymbol{m}_1^n \otimes \boldsymbol{m}_2^n\right)^T \end{bmatrix}}_{A} E^s = 0 \tag{2.64}$$

which is linear in the unknown E^s. An approximation of the unit length E^s is the solution to (2.64) which is given by minimising $\|AE^s\|$. This is done by computing the SVD of A and define E^s as the singular vector corresponding to the smallest singular value. The required number of correspondences is eight since E has nine entries which are only defined up to scale which means that there are only eight unknowns. Due to noise, the recovered matrix will often not lie in the essential space, however, a solution is to project the matrix onto this space. To further suppress noise the number of correspondences should be more than eight.

There are however situations where the solution to the algorithm is not well defined if e.g., all correspondence points lie in the same plane or the baseline $\boldsymbol{c} \approx 0$. This is partly because the actual degrees of freedom are not taken into account in the algorithm since it considers eight unknowns. Another weakness is the sensitivity to outliers since there is no mechanism for handling these directly. Despite these limitations, the eight-point algorithm was successfully used in Paper D and Paper E for initialisation of rotations without considering potential outliers. However, outliers was later removed in an iterative fashion using IMU data.

The result from eight-point-like methods can be improved by minimising the reprojection error (2.47) using nonlinear optimisation. This requires the landmark coordinates to be known. Instead a suitable error which is only parametrised by the essential matrix is a so-called Sampson error. For n correspondence $\{\boldsymbol{m}_1^i \leftrightarrow$

$\boldsymbol{m}_2^i\}_{i=1}^n$ it looks like

$$V(E) = \sum_{i=1}^{n} \frac{\left(\left(\boldsymbol{m}_2^i\right)^T E \boldsymbol{m}_1^i\right)^2}{(E\boldsymbol{m}_1^i)_1^2 + (E\boldsymbol{m}_1^i)_2^2 + (E^T\boldsymbol{m}_2^i)_1^2 + (E^T\boldsymbol{m}_2^i)_2^2}, \tag{2.65}$$

where $(E\boldsymbol{m})_i$ picks out the i-th entry of the epipolar line $E\boldsymbol{m}$. This Sampson error is a first order approximation of the re-projection error, see e.g., Hartley and Zisserman (2004), and it is extensively used in geometric computer vision problems.

An approach which exploits the 5 DOF imposed by the construction of the essential matrix is the five-point algorithm proposed by (Nistér, 2004). The idea Nister used was that the essential matrix has two non-zero eigenvalues which are equal and this gives rise to a cubical matrix constraint. The algorithm computes the coefficients of a tenth degree polynomial and finds its roots. A perhaps simpler implementation can be found in (Li and Hartley, 2006) which solves the tenth degree polynomial using a hidden variable resultant. An important observation is that the five point method seems less sensitive than the eight-point algorithm to cases when e.g., all points lie on a plane. It also attains higher accuracy because a minimal solver may better exploit the geometric constraints (Li and Hartley, 2006).

RANSAC

Efficient parameterisation which allows fast and/or closed form solutions plays an important role in computer vision algorithms. This is because the set of correspondences between any two images will often contain gross outliers due to association errors which need to be handled. This is usually done by evaluating several candidate models in a RANSAC loop (Fischler and Bolles, 1981). In practice, higher accuracy is obtained by evaluating as many candidates as possible and this is why fast solutions are important since it allows more trials given the computational resources. The acronym RANSAC comes from RANdom Sampling And Consensus Fischler and Bolles (1981) and its paradigm has had a remarkable impact in computer vision. It is an iterative method which aims at finding a model such that the number of inliers is maximised. All that is needed in its basic form is a set of correspondence candidates and some error metric to evaluate the model.

The RANSAC algorithm starts by randomly selecting the minimal set needed to compute a model and then all the other correspondences are evaluated by the error metric. A correspondence pair is labeled as an inlier if it falls below a predefined threshold and is otherwise considered to be an outlier. For each model the number of inliers is saved and this is often referred to as scoring. The Sampson error (2.65) is a popular candidate for scoring the essential matrices within the RANSAC loop. The model with the maximum number of inliers, i.e., the largest consensus set, is considered to be the best. Optionally, a final step is to estimate the model again using all the inliers by e.g., minimising (2.65).

The greatest advantage of RANSAC is the ability to robustly estimate the parameters of a model when the measurements are contaminated with outliers while simultaneously finding an inlier set. The number of samples needed for a robust estimate also depends on the number of correspondences needed to estimate the model. It is therefore desirable to have a minimal parametrisation since it is then more likely that samples only contains inliers. It can also be used only as a sampling based method for outlier rejection. A clear limitation is that there is no bound on how many times the loop has to be run in order meet some stopping criterion such as a specific inlier/outlier ratio. RANSAC typically works poorly if there are less than fifty percent of inliers. Despite (or due to) these limitations there are probably hundreds of RANSAC versions published which have different characteristics. For instance, the Maximum Likelihood version called MLESAC (Torr and Zisserman, 2000) is a good option which modifies the score by penalizing outliers.

An example of essential matrix estimation using RANSAC is shown in Figure 2.5 where the integrated yaw angle of the estimated rotation matrices is compared with the yaw angle obtained by integrating the gyroscope signal. The yaw angle estimate from the camera works well for the first 15 seconds and then deteriorates rapidly from the integrated yaw angle of the gyroscope. The gyroscope can be considered a reliable ground truth for this short experiment. This highlight some of the different characteristics of the two sensors. Gyroscopes can handle fast dynamics but may deteriorate over time, due to noise integration, while cameras may have problems with fast dynamics they can be used to estimate orientation accurately for moderate dynamics over long time.

2.7 Computer Vision

The intention with this section is only to provide some basic concepts of computer vision. The interested reader is referred to the two survey papers by Fraundorfer and Scaramuzza (2012); Scaramuzza and Fraundorfer (2011) and references therein, for a modern and detailed exposition.

In order to use the camera measurement models some image processing is needed. Image characteristics having strong response in a detector are used to define so-called interest points. For this to work, the scene has to have some structure. The most commonly used detectors are so-called blob-detectors based on *difference of Gaussians* (DoG) such as the *scale invariant feature transform* (SIFT) (Lowe, 1999) or based on distinctive corners such as the Harris detector (Harris and Stephens, 1988).

The process of tracking interest points over several image frames, also known as correspondence generation, relies on selecting well localised interest points and some local interest point description which can be located in the following frames. Although we have not used patch based methods for tracking in the publications a brief description is given here for completeness. For Harris-like features a patch U around the interest point $\boldsymbol{x}$ can be used to search for its cor-

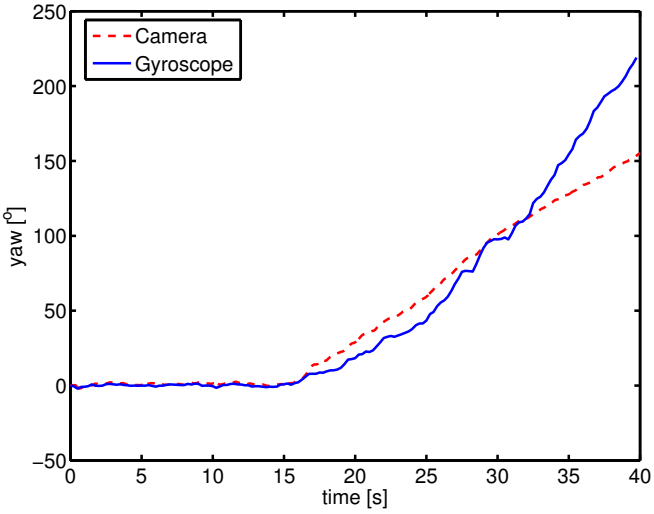


Figure 2.5: *Integrated yaw angle from the 5-point algorithm using RANSAC on the Sampson error compared with the integrated gyroscope measurements on data from a flying platform.*

responding location in the next frame J only assuming a simple displacement model in the image

$$J(\boldsymbol{x} + d) = U(\boldsymbol{x}). \tag{2.66}$$

This is a popular model which usually works well for small perspective changes and it is used in most *Kanade-Lucas-Tomasi* (KLT) trackers as explained in Shi and Tomasi (1994). More advanced transformation models for patch matching allowing for rotation, scaling and deformation as

$$J(\boldsymbol{x} + D\boldsymbol{x} + d) = U(\boldsymbol{x}), \tag{2.67}$$

where $D \in \mathbb{R}^{2\times 2}$ is a deformation matrix, is particularly useful in longer sequences.

The SIFT detector also stores a feature descriptor for each detected interest point, see Figure 2.6. In the standard setting the descriptor vector has 128 values corresponding to an 8-bin histogram of gradient orientations for each of the 16 local sub-regions around the feature point. The SIFT descriptor can be used for generating correspondences without using any transformation model simply by comparing descriptors using any suitable metric, e.g., the Euclidean distance. This way, matches are found using the appearance of feature descriptors and not their locations. Thus, the set of matches which corresponds to the same feature is called a feature track (Thormählen et al., 2008). Such methods are particularly useful if there is a great deal of uncertainty about motion and/or structure or at loop-closing. In Paper D a purely appearance based approach is used for an initial correspondence search. That is, without assuming anything about the mo-

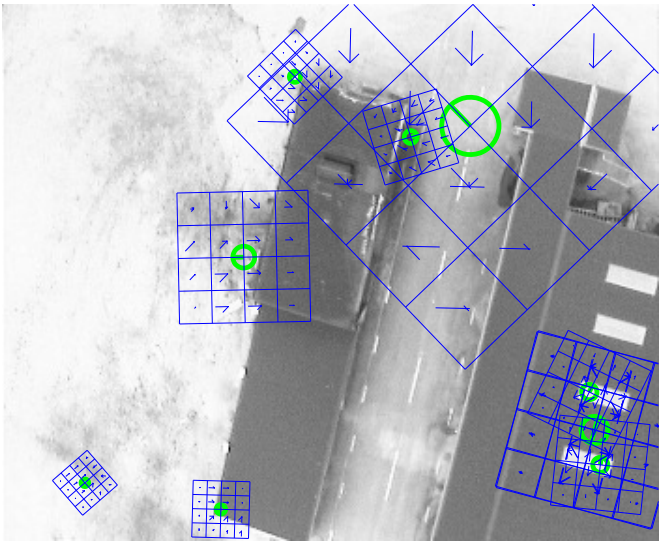


Figure 2.6: *Ten SIFT features are shown with their center location given by the circles. Descriptors computed at different scales with sixteen histograms of gradient orientations indicated by the arrows in each square. Thus, a feature may have multiple orientations depending on their local appearance.*

tion between frames correspondences are found only by matching SIFT feature descriptors and not using their point location information. SIFT descriptors and their predicted locations can also be used directly for tracking which was done in Paper B. A key advantage of SIFT is that feature descriptors are rather insensitive w.r.t., changes in scale, illumination, rotation and even viewpoint changes up to 60° (Fraundorfer and Scaramuzza, 2012). These properties make SIFT suitable for matching over wide baselines which also implies that good tracking results and essential matrix estimates can be obtained using a lower sampling frequency compared to KLT-like trackers.

3
Estimation

Estimation is the problem of taking measured data, y_t, to reveal properties, x_t and θ, of systems which are of interest. Basic ingredients are model structures representing the assumptions made on the interaction and time evolution of x_t, θ and y_t. As tools for solving estimation problems, state space models

$$x_t = f(x_{t-1}, u_t, \theta) + w_t, \tag{3.1a}$$
$$y_t = h(x_t, \theta) + e_t, \tag{3.1b}$$

are commonly used model structures. State-space formulations can also be described in terms of their *probability density functions* (PDFs)

$$x_t \sim p(x_t|x_{t-1}, \theta), \tag{3.2a}$$
$$y_t \sim p(y_t|x_t, \theta), \tag{3.2b}$$

where (3.2a) is known as the state transition density, (3.2b) is known as the measurement likelihood and the symbol $\sim$ corresponds to a distribution relation.

3.1 Sensor Fusion

Sensor fusion is the process of combining multi-sensory data in a clever way to obtain a filtered state estimate, $\hat{x}_t$, or a state sequence estimate $\{\hat{x}_i\}_{i=0}^{t} = \hat{x}_{0:t}$. A state-space model is the key component of many sensor fusion algorithms and model parameters, θ, are usually not of interest. Sensor fusion often use *maximum likelihood* (ML) (Fisher, 1912) estimators which have the form

$$\hat{\theta}^{ML} = \arg\max_{\theta} \; p(Y|\theta). \tag{3.3}$$

Another important estimator is the *maximum a posteriori* (MAP)

$$\hat{\theta}^{MAP} = \arg\max_{\theta} \; p(\theta|Y) = \arg\max_{\theta} \; \frac{p(Y|\theta)p(\theta)}{p(Y)} = \arg\max_{\theta} \; p(Y|\theta)p(\theta), \tag{3.4}$$

where the second equality is known as Bayes' therorem and the last equality is using the fact that the maximising argument is independent of the normalising constant $p(Y)$.

3.1.1 Smoothing and Filtering

The smoothed state $\hat{x}_{0:t}$ defines an estimate of the whole state trajectory and it can be obtained as the MAP estimate of the state sequence given the measurements

$$\hat{x}_{0:t} = \arg\max_{x_{0:t}} \; p(y_{0:t}|x_{0:t})p(x_{0:t}). \tag{3.5}$$

With Gaussian noise and initial state $x_0 \sim \mathcal{N}(\bar{x}_0, P)$ the negative log of (3.5) times two becomes a Gaussian MAP estimation problem

$$\begin{aligned} \hat{x}_{0:t} = \arg\min_{x_{0:t}} \quad & \|\bar{x}_0 - x_0\|^2_{P^{-1}} + \sum_{i=1}^{t} \|x_i - f(x_{i-1})\|^2_{Q^{-1}} + \sum_{i=1}^{t} \|y_i - h(x_i)\|^2_{R^{-1}} = \\ \arg\min_{x_{0:t}} \quad & V(x_{0:t}), \end{aligned} \tag{3.6}$$

where the terms not directly depending on $x_{0:t}$ have been left out, see for instance Rao (2000) for details. This is also a nonlinear least squares formulation, a topic which will be treated in Section 4.3.3. Note that if only a part of the whole batch is considered at each time step a so-called moving horizon estimation problem is obtained. Alternatively, the process and measurements can be viewed as equality constraints and the constrained formulation of (3.6) is then

$$\hat{x}_{0:t} = \arg\min_{x_{0:t}} \quad \|\bar{x}_0 - x_0\|^2_{P^{-1}} + \sum_{i=1}^{t} \|w_i\|^2_{Q^{-1}} + \sum_{i=1}^{t} \|e_i\|^2_{R^{-1}}, \tag{3.7a}$$

$$\text{subject to} \quad x_i = f(x_{i-1}) + w_i, \tag{3.7b}$$

$$y_i = h(x_i) + e_i. \tag{3.7c}$$

MAP estimation can be extended to include other parameters in f and h, besides the state, resulting in a joint smoothing and parameter estimation problem. The particular benefit with this optimisation viewpoint is that it is straightforward to add constraints which is not easily done in a filtering context.

In the case of linear dynamics and linear measurement equations (3.6) becomes a convex optimisation problem that can be efficiently implemented as a *Rauch-Tung-Striebel* (RTS) smoother Rauch et al. (1965). A straightforward, yet approximate, extension to nonlinear systems is given by the *extended-RTS* (E-RTS) smoother where the forward filter is realised using an EKF. This method is used in Paper E. Note that in this nonlinear setting the E-RTS may be improved using iterations, step control, and other techniques, since in the end we are just solving a nonlinear optimisation problem.

Algorithm 1 Extended Kalman Filter

Available measurements are $Y = \{y_1, \ldots, y_N\}$. Require an initial state, $\hat{x}_{0|0}$, and an initial state covariance, $P_{0|0}$, and use the models (3.1).

1. Time Update

$$\hat{x}_{t|t-1} = f(\hat{x}_{t-1|t-1}, u_t), \tag{3.8a}$$

$$P_{t|t-1} = F_t P_{t-1|t-1} F_t^T + Q_t, \tag{3.8b}$$

2. Measurement Update

$$S_t = H_t P_{t|t-1} H_t^T + R_t, \tag{3.9a}$$

$$K_t = P_{t|t-1} H_t^T S_t^{-1}, \tag{3.9b}$$

$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + K_t \left(y_t - h(\hat{x}_{t|t-1}) \right), \tag{3.9c}$$

$$P_{t|t} = P_{t|t-1} - K_t H_t P_{t|t-1}. \tag{3.9d}$$

Where

$$F_t \triangleq \left. \frac{\partial f(x_t, u_t)}{\partial x_t} \right|_{(x_t, u_t) = (\hat{x}_{t-1|t-1}, u_t)}, \qquad H_t \triangleq \left. \frac{\partial h(x_t)}{\partial x_t} \right|_{(x_t) = (\hat{x}_{t|t-1})}, \tag{3.10}$$

while, Q_t and R_t are the covariance matrices of w_t and e_t, respectively.

Extended Kalman Filter

A popular estimator is the extended Kalman filter which is described in many books, see e.g., Kailath et al. (2000). The EKF works in a two step procedure summarised in Algorithm 1. As with the E-RTS there are no convergence guarantees since the involved functions are nonlinear.

3.2 Optimisation

Many batch and filtering problems can be formulated in terms of optimisation programs to which there are many software packages readily available. A quite general optimisation program is

$$\underset{\theta}{\text{minimise}} \quad V(\theta) \tag{3.11a}$$

$$\text{subject to} \quad c_E(\theta) = 0 \tag{3.11b}$$

$$c_I(\theta) \leq 0, \tag{3.11c}$$

where $V : \mathbb{R}^n \to \mathbb{R}$ is the objective function, c_E are equality constraints, c_I are inequality constraints and θ are the variables.

The Lagrange function, often just called the Lagrangian, is obtained by taking the optimisation problem (3.11) and augmenting the objective function with a weighted sum of the constraints as

$$L(\theta, \lambda, \nu) = V(\theta) + \lambda^T c_E(\theta) + \nu^T c_I(\theta), \tag{3.12}$$

with associated dual variable vectors, λ and ν. The first order necessary optimal-

ity conditions are

$$\nabla V(\theta) + \lambda^T \nabla c_E(\theta) + \nu^T \nabla c_I(\theta) = 0 \tag{3.13a}$$
$$c_E(\theta) = 0 \tag{3.13b}$$
$$c_I(\theta) \leq 0 \tag{3.13c}$$
$$\nu \geq 0 \tag{3.13d}$$
$$\operatorname{diag}(\nu) c_I(\theta) = 0 \tag{3.13e}$$

which have to be satisfied at the optimum $(\theta^*, \lambda^*, \nu^*)$ and (3.13a) is the gradient of the Lagrangian (3.12). These equations are often referred to as the *Karush-Kuhn-Tucker* (KKT) conditions, see e.g., (Boyd and Vandenberghe, 2004; Nocedal and Wright, 2006) for detailed explanations and some historical notes. For convex problems the KKT conditions are also sufficient whereas for non-convex problems a KKT point is merely a candidate solution.

Specific classes of problems can be identified depending on e.g., the properties of V, the choice of variables, among others. If, for example, the objective function is $V(\theta) = \frac{1}{2}\theta^T G\theta + \theta^T c$, G is a symmetric matrix, and the constraints are linear in θ, then a *quadratic program* (QP) is obtained. Many problems can be transformed into an equivalent QP and an instructive example on linear Kalman filtering is given below.

Example 3.1: KF measurement update as a Quadratic Program

The Kalman filter iterates two equations for the state

$$\hat{x}_{t|t-1} = A\hat{x}_{t-1|t-1} \tag{3.14a}$$
$$\hat{x}_{t|t} = \hat{x}_{t|t-1} + P_{t|t-1}C^T(CP_{t|t-1}C_t^T + R)^{-1}(y_t - C\hat{x}_{t|t-1}), \tag{3.14b}$$

where (3.14a) is the time update and (3.14b) is the measurement update. Equivalently, the left hand side of the measurement update can be specified as the following QP

$$\{\hat{x}_{t|t}, \hat{e}_t\} = \underset{x,e_t}{\arg\min} \quad e_t^T R^{-1} e_t + (x - \hat{x}_{t|t-1})^T P_{t|t-1}^{-1}(x - \hat{x}_{t|t-1}) \tag{3.15a}$$
$$\text{subject to} \quad y_t = Cx + e_t, \tag{3.15b}$$

or alternatively, the unconstrained version is

$$\hat{x}_{t|t} = \underset{x}{\arg\min}\, (y_t - Cx)^T R^{-1}(y_t - Cx) + (x - \hat{x}_{t|t-1})^T P_{t|t-1}^{-1}(x - \hat{x}_{t|t-1}). \tag{3.16}$$

The actual gain from these formulations, compared to the standard KF equations (3.14), is that constraints can be added easily and modification of the objective function becomes straightforward. Such an example is modelling of so called heavy-tailed noise which is a direct approach to handle non-Gaussian residuals. This can be treated by adding another variable z_t to (3.15), as in (Mattingley and

Boyd, 2010), resulting in

$$\{\hat{x}_{t|t}, \hat{e}_t, \hat{z}_t\} = \underset{x,e_t,z_t}{\arg\min} \quad e_t^T R^{-1} e_t + (x - \hat{x}_{t|t-1})^T P_{t|t-1}^{-1}(x - \hat{x}_{t|t-1}) + \lambda \|z_t\|_1 \tag{3.17a}$$

$$\text{subject to} \quad y_t = Cx + e_t + z_t, \tag{3.17b}$$

where $\lambda \geq 0$ is a design parameter controlling how much the $\|z_t\|_1$-norm (ℓ_1) should be favoured. This is not a QP anymore, however it can be transformed into another QP which has the form

$$\{\hat{x}_{t|t}, \hat{e}_t, \hat{z}_t\} = \underset{x,e_t,z_t}{\arg\min} \quad e_t^T R^{-1} e_t + (x - \hat{x}_{t|t-1})^T P_{t|t-1}^{-1}(x - \hat{x}_{t|t-1}) + \lambda \mathbf{1}^T u \tag{3.18a}$$

$$\text{subject to} \quad y_t = Cx + e_t + z_t, \tag{3.18b}$$

$$-u \leq z_t \leq u. \tag{3.18c}$$

For small λ some elements in z_t will be exactly zero, whereas for large λ all elements will be zero and the original problem is obtained. This means that the filter is more robust with respect to non-Gaussian errors. The ℓ_1-norm can be interpreted in a statistical sense as z_t being Laplace distributed $p(z_t) = \mathcal{L}(0, 2\sigma^2/\lambda)$, see e.g., Hastie et al. (2009). ℓ_1-regularisation is also used to find sparse parameter estimates and these metods are known as *least absolute shrinking and selection operator* (LASSO) Tibshirani (1996).

Constrained programs can be transformed into unconstrained counterparts using the Lagrangian and additional penalisation terms to account for dropped constraints. Many software packages can solve constrained problems by parsing them as unconstrained ones, or vice versa, if it is more efficient to solve them this way. Popular unconstrained methods are steepest descent, (quasi-) Newton and trust-region, see e.g., Nocedal and Wright (2006); Dennis and Schnabel (1983).

3.2.1 Iterated Extended Kalman Filter

An interesting parallel between filtering and optimisation is that the measurement update in the EKF (3.9) is the solution to the following NLS problem

$$\hat{x}_{t|t} = \underset{x}{\arg\min} \ \frac{1}{2}\|y_t - h(x)\|^2_{R_t^{-1}} + \frac{1}{2}\|x - \hat{x}_{t|t-1}\|^2_{P^{-1}_{t|t-1}} = \underset{x}{\arg\min} \ V(x) \tag{3.19}$$

which is the nonlinear counterpart of (3.16). The EKF solution is given by a full, single step, in a Gauss-Newton procedure (Bertsekas, 1994). Note that in the filtering case it does not matter if the state dynamics are nonlinear or not. The pure EKF may be a poor choice if e.g., the predicted state is far from the true one. Albeit Gauss-Newton does not promise global (or even local) convergence, iterations may improve the estimate. This approach is used in the *iterated-EKF* (IEKF), see e.g., Jazwinski (1970); Bell and Cathey (1993); Bar-Shalom and Li (1993), in

which the measurement update for the state is iterated a few times

$$H_i = \left.\frac{\partial h(x)}{\partial x}\right|_{x=x_i}, \tag{3.20a}$$

$$K_i = P_{t|t-1}H_i^T(H_iP_{t|t-1}H_i^T + R_t)^{-1}, \tag{3.20b}$$

$$x_{i+1} = \hat{x}_{t|t-1} + K_i\left(y_t - h(x_i) - H_i(\hat{x}_{t|t-1} - x_i)\right), \tag{3.20c}$$

where the first iteration is exactly the same as in the EKF measurement update. The measurement updated state and the covariance approximation is

$$\hat{x}_{t|t} = x_{i+1}, \tag{3.21a}$$

$$P_{t|t} = P_{t|t-1} - K_iH_iP_{t|t-1}, \tag{3.21b}$$

where the updated covariance is only updated using the last Kalman gain and measurement Jacobian, K_i and H_i, respectively. The IEKF assumes that each step reduces the cost (3.19)

$$V(x_{i+1}) \leq V(x_i), \tag{3.22}$$

without evaluating if that was the case. Modifying the update (3.20c) as

$$\begin{aligned} x_{i+1} &= \hat{x}_{t|t-1} + K_i\left(y_t - h(x_i) - H_i(\hat{x}_{t|t-1} - x_i)\right), \\ &= x_i + \left(\hat{x}_{t|t-1} - x_i + K_i\left(y_t - h(x_i) - H_i(\hat{x}_{t|t-1} - x_i)\right)\right), \end{aligned} \tag{3.23}$$

a step control parameter α can be introduced to ensure cost reduction. The modified measurement update of the IEKF is then on the form

$$x_{i+1} = x_i + \alpha_i\left(\hat{x}_{t|t-1} - x_i + K_i\left(y_t - h(x_i) - H_i(\hat{x}_{t|t-1} - x_i)\right)\right), \tag{3.24}$$

$$= x_i + \alpha_i p_i, \tag{3.25}$$

where the step length $0 < \alpha \leq 1$ and the search direction p is chosen such that (3.22) is satisfied in each step. We call this approach IEKF-L. More advanced line search strategies that employs conditions on the curvature for sufficient decrease could of course be used. The IEKF with and without line search is illustrated in Example 3.2.

Example 3.2: Bearings Only Tracking and IEKF

An applied example of IEKF bearings-only tracking is studied, which also can be found in Gustafsson (2012) as Example 8.1. For simplicity, the target is stationary, i.e., $w_t = 0$ and $Q = 0$, at the true position $x^* = [1.5, 1.5]^T$. The bearing measurement function from the j-th sensor S^j at time t is

$$y_t^j = h_j(\boldsymbol{x}_t) + e_t = \arctan2(y_t - S_y^j, x_t - S_x^j) + e_t, \tag{3.26}$$

where arctan2() is the two argument arctangent function, S_y and S_x denotes the y and the x coordinates of the sensors, respectively. The noise is $e_t \sim \mathcal{N}(0, R_t)$.

The Jacobians of the dynamics and measurements are

$$F = I, \qquad H^j = \frac{1}{(x - S_x^j)^2 + (y - S_y^j)^2} \begin{bmatrix} -(y - S_y^j) \\ x - S_x^j \end{bmatrix}. \tag{3.27}$$

With the two sensors having positions $S^1 = [0, 0]^T$ and $S^2 = [1.5, 0]^T$. The filter is initialised with $\hat{x}_{0|0} = [1, 1]^T$ and $P_{0|0} = I_2$. In Figure 3.1 the IEKF is compared with the EKF for a single realisation with $e_t \sim \mathcal{N}\left(0, \pi 10^{-3} I_2\right)$ for the left plot and ten noise-free measurements for the right plot with the same covariance as in the left plot. If this artificial measurement covariance is decreased by a few orders of magnitude, then the EKF covariance does not capture the true uncertainty but the IEKF does. For this simple example 10 iterations are used since then the IEKF

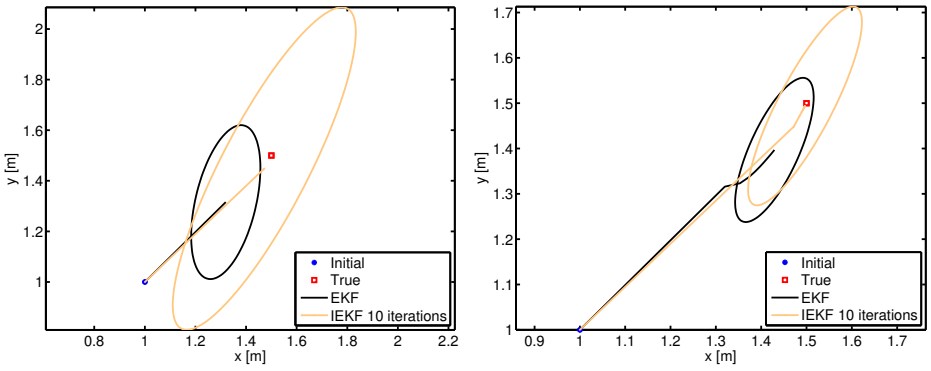


Figure 3.1: *Left: The IEKF is iterated 10 times and comes closer to the true state with a covariance that captures the uncertainty. Right: 10 noise-free measurements are given and the IEKF, again iterated 10 times, converges to the true position much faster than the EKF.*

have converged and the result clearly improves the estimate. However, it may be sufficient to perform fewer iterations to speed up execution.

The cost function for this example is nearly convex close to the true target. This means that if we start close enough it should be safe to take apply the IEKF without any modifications. It was however also verified that the actual cost decreased in each step. In the left plot in Figure 3.2 the target is in the same location as before but the initial guess is $\hat{x}_{0|0} = [0.2, 1.5]^T$. Given one measurement from each sensor the EKF performs one update and the IEKF performs 3 iterations. It is obvious that both the target location and the covariance estimate is much better when using the IEKF.

Starting with the initial guess $\hat{x}_{0|0} = [0.3, 0.1]^T$ with $P_{0|0} = 2I_2$ where the curvature of the cost function is bit more difficult. The EKF and IEKF-L are given perfect measurement but with assumed covariances $Q = R = \pi 10^{-3} I_2$. The position estimate is shown in the right plot in Figure 3.2. Note that the IEKF-L covariance is consistent as opposed to the EKF. The rapid convergence of the IEKF-L is not

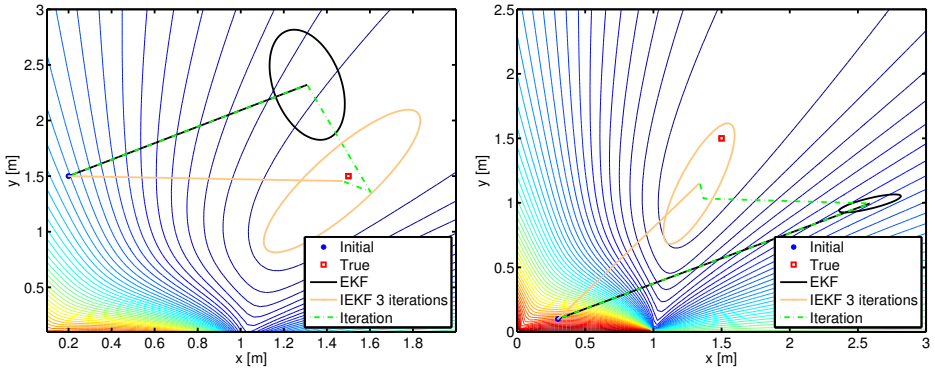


Figure 3.2: *Left: The target estimate and the covariance is much better using the IEKF. The dashed lines corresponds to the iterated estimate in the IEKF. Note that the first step in the IEKF is the same as the EKF. The level curves illustrates the NLS cost with a minimum at the true target position. Right: The EKF has poor performance and under-estimates the covariance while IEKF-L gives good results by step size reduction and just 3 iterations.*

surprising since the variable step is based on cost decrease at each iteration. More iterations does not give any significant improvement and the steps gets short.

3.2.2 Nonlinear Least Squares

Nonlinear least squares problems are obtained when the objective function is a sum of squared errors as in the MAP smoothing problem (3.6) or the constrained form (3.7a). Popular solvers are Gauss-Newton and Levenberg-Marquardt which approximates nonlinear functions by appropriate linear functions. The goal in NLS is to estimate parameters, θ, by minimising a parametrised residual vector, $\widetilde{\varepsilon}(\theta) \sim \mathcal{N}(0, \Sigma_\theta)$. For notational convenience the residuals are normalised according to their assumed covariance

$$\varepsilon(\theta) = \Sigma_\theta^{-T/2} \widetilde{\varepsilon}(\theta), \tag{3.28}$$

where $\Sigma^{-T/2}$ denotes the transposed matrix square-root and then the Mahalanobis notation can be dropped since $\|\widetilde{\varepsilon}(\theta)\|^2_{\Sigma_\theta^{-1}} = (\Sigma_\theta^{-T/2}\widetilde{\varepsilon}(\theta))^T(\widetilde{\varepsilon}(\theta)\Sigma_\theta^{-T/2}) = \|\varepsilon(\theta)\|_2^2$. The residuals are said to be minimised in a least-squares sense by the cost

$$V(\theta) = \frac{1}{2}\sum_{t=1}^{N} \|\varepsilon_t(\theta)\|_2^2 = \frac{1}{2}\sum_{t=1}^{N} \varepsilon_t^T(\theta)\varepsilon_t(\theta). \tag{3.29}$$

Algorithm 2 Gauss-Newton

1. Require initial an estimate θ and Jacobian $J(\theta)$
2. Compute a search direction p by solving
$$J(\theta)J(\theta)^T p = -J(\theta)\varepsilon(\theta). \tag{3.33}$$
3. Compute a step length, α, such that the cost (3.29) is decreased. This can be done using line search, see e.g., Nocedal and Wright (2006, page 297) or Boyd and Vandenberghe (2004, page 464).
4. Update the parameters
$$\theta := \theta + \alpha p. \tag{3.34}$$
5. Terminate if a stopping criteria is met. Such criteria can be; the change in cost is small, the number of iterations has exceeded some threshold, among others.
6. Otherwise, return to step 2.

Define $\varepsilon(\theta) = [\varepsilon_1(\theta)\ \varepsilon_2(\theta)\ \dots\ \varepsilon_N(\theta)]^T$ and the Jacobian $J(\theta) = \frac{d\varepsilon^T(\theta)}{d\theta}$, then the gradient and the Hessian of (3.29) with respect to the parameters are given by

$$\nabla V(\theta) = \frac{dV(\theta)}{d\theta} = \frac{1}{2}\sum_{t=1}^{N} \varepsilon_t(\theta)\frac{d\varepsilon_t(\theta)}{d\theta} = J(\theta)\varepsilon(\theta), \tag{3.30}$$

and

$$\frac{d^2V(\theta)}{d\theta^2} = J(\theta)J(\theta)^T + \frac{1}{2}\sum_{t=1}^{N} \varepsilon_t(\theta)\frac{d^2\varepsilon_t^T(\theta)}{d\theta^2}, \tag{3.31}$$

respectively. The extension to multivariable residuals is easily obtained by stacking the vectorisation of the individual residuals which again gives a scalar cost function.

The Gauss-Newton method can be seen as a modified Newton method which applies only to NLS problems. It is computationally cheap since the Hessian of the objective function is approximated as

$$\frac{d^2V(\theta)}{d\theta^2} \approx J(\theta)J(\theta)^T, \tag{3.32}$$

thus there is no need for second order derivatives. The approximation is good when the initial θ is close to the optimum but it may be bad if some residuals are large. An option is then to include the second order terms in (3.31) or approximate them with some secant method. The Gauss-Newton method as an algorithm is summarised in Algorithm 2. The Gauss-Newton method may encounter problem if the Jacobian is singular or ill-conditioned. A straightforward remedy is given by the Levenberg-Marquardt algorithm which modifies the normal equations as

$$(JJ^T + \mu I)p = -J\varepsilon. \tag{3.35}$$

where μ is a positive number and the θ dependence have been omitted. The μ parameter acts as an interpolation of the Gauss-Newton method and gradient

descent. This can be seen by noticing that Gauss-Newton corresponds to $\mu = 0$ and for large μ the step will be approximately in the gradient descent direction which typically is the preferred option if the initial solution is far from the optimum. The parameter is adaptively updated by some method and should typically decrease as the minimum is approached. Another modification suggested by Marquardt (1963) is to take larger steps in the direction in which the gradient is small by

$$(JJ^T + \mu \operatorname{diag}\{JJ^T\})p = -J\varepsilon, \tag{3.36}$$

and thus speed up convergence.

Note that in practice parameters are often subject to constraints, for instance; unit quaternions should have unit length; physical landmarks need to be in front of the camera (Hartley, 1998); motion is constrained by motion models. When parameters are updated by a simple increment such constraints may be violated. It is therefore important to have a stable local parametrisation for the update followed by some procedure such that the parameters do not violate the constraints, for instance normalisation of an updated quaternion. Such an approach is used in Paper D since the gyroscope rates are natural parameters for the local update and the unit quaternion for global rotation parametrisation.

3.3 Problem Structure

For problems with many parameters the key to efficiency is to utilise the specific structure of each problem in the corresponding Jacobian and the Hessian approximation. Although the normal equations can be computed efficiently by numerical matrix factorisations, such as QR, LDL and SVD it may not be a good option if the factorisation needs to be updated. For large problems even the computation of the normal equations may be infeasible due to e.g., memory or time constraints. Explicit, rather than numerical, factorisations are then a good option. A well-known trick for efficient equation system solving is to use the Schur complement and an example of this is given below.

Example 3.3: Schur Complement

In batch problems such as NLS-SLAM and *bundle adjustment* (BA) there is a natural sparsity in the Jacobian and thus in the Hessian approximation. Partition the normal equations (3.33) as

$$\begin{bmatrix} J_c J_c^T & J_c J_m^T \\ J_m J_c^T & J_m J_m^T \end{bmatrix} \begin{bmatrix} p_c \\ p_m \end{bmatrix} = -\begin{bmatrix} J_c \varepsilon_c \\ J_m \varepsilon_m \end{bmatrix} \iff \begin{bmatrix} H_{cc} & H_{cm} \\ H_{mc} & H_{mm} \end{bmatrix} \begin{bmatrix} p_c \\ p_m \end{bmatrix} = -\begin{bmatrix} J_c \varepsilon_c \\ J_m \varepsilon_m \end{bmatrix} \tag{3.37}$$

where J_c and J_m are the Jacobian w.r.t., the camera poses and landmarks respectively. The primary sparsity structure in BA comes from the fact that each landmark observation only depends on one camera pose which means that H_{cc} and H_{mm} are block diagonal. Solving (3.37) for p_c and p_m can be done efficiently using

the Schur complement of e.g., H_{mm} as

$$\left(H_{cc} - H_{cm}H_{mm}^{-1}H_{mc}\right)p_c = -J_c\varepsilon_c + H_{cm}H_{mm}^{-1}J_m\varepsilon_m \tag{3.38a}$$

$$p_m = H_{mm}^{-1}\left(-J_m\varepsilon_m - H_{mc}p_c\right) \tag{3.38b}$$

where the much smaller reduced system (3.38a) is solved first and p_c is back-substituted into (3.38b). And since H_{mm} is block diagonal, its inverse is cheap to compute. Furthermore, it is often the case that all landmarks are not observed the whole time and this gives rise to a secondary sparsity structure in the H_{cm} matrix which can be exploited.

For very large problems the normal equations may be solved by quasi-Newton methods such as L-BFGS (Nocedal, 1980) which efficiently approximate the inverse Hessian. Other approaches are conjugate gradients or in a distributed way using the *alternating direction method of multipliers* (ADMM) (Boyd et al., 2011). These methods converge slowly, yet they may be the only feasible option for large systems where direct methods are not suitable. In general there are few, if any, guarantees that the global optimum will be found by any nonlinear program. However, careful selection of the initial starting point greatly improves the chance of reaching the optimum.

4

SLAM

In this chapter a brief overview of some SLAM estimation methods is given. More thorough descriptions are given in the three appended SLAM publications.

4.1 Introduction

SLAM problems are combinations of localisation and mapping type problems solved simultaneously. There are mainly two strategies and these are either based on filtering or on batch optimisation. The early research almost exclusively focused on filtering methods which recursively incorporate the measurements estimating the posterior filtering density $p(x_t, m|y_{1:t})$ and the list of filtering based acronyms is long. In summary, many of them are either using particle filters Montemerlo et al. (2002) or extended Kalman filters, Smith et al. (1990). Figure 4.1 illustrates a SLAM setup with a moving platform and observing landmark features from different locations.

4.1.1 Probabilistic Models

The target in SLAM is to either maximise the posterior density of the complete trajectory and map

$$p\left(x_{0:t}, m|y_{0:t}\right), \tag{4.1}$$

or the filtering density

$$p\left(x_t, m|y_{0:t}\right), \tag{4.2}$$

which is obtained by marginalising old states. Here $x_{0:t}$ is the whole state trajectory, x_t is the current state, m is a static map and are the $y_{0:t}$ measurements relating to the state and the map. For notational convenience, correspondence

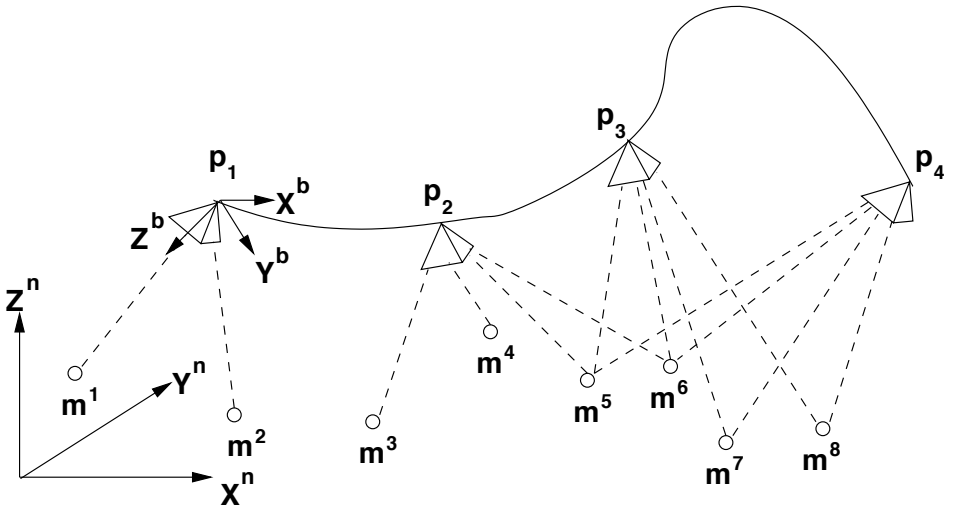


Figure 4.1: *A moving platform with body coordinate system* (b) *is observing an environment represented by point landmarks,* $m^1, \ldots, m^8$*. Also, a global, fixed navigation coordinate system,* (n)*, is drawn.*

variables and input signals are not made explicit. The SLAM posterior (4.1) can for instance be factorised as

$$p\left(x_{0:t}, m \,|y_{0:t}\right) = p(x_{0:t}|y_{0:t})p(y_{0:t}|x_{0:t}, m) = p(x_{0:t})p(y^x_{0:t}|x_{0:t})p(y^{x,m}_{0:t}|x_{0:t}, m), \quad (4.3)$$

where the first factor is the process model, the second factor is the measurement likelihood independent of the map, e.g., GPS or IMU measurements, and the third is the measurement likelihood of measurements that depend on the map and the process. From (4.3) the filtering and smoothing forms are straightforwardly obtained. As was shown in Section 3.1.1 the smoothing density becomes an NLS problem if the noise sources are Gaussian as is the common assumption in most algorithms. The batch formulation is the target in GraphSLAM (Thrun and Montemerlo, 2006; Thrun et al., 2005). Similar to FastSLAM (Montemerlo et al., 2002, 2003), the posterior is factorised as

$$p(x_{0:t}, m \,|y_{0:t}) = p(x_{0:t}|y_{0:t})p(m \,|x_{0:t}, y_{0:t}), \quad (4.4)$$

where $p\left(x_{0:t}|y_{0:t}\right)$ is the posterior of trajectories. This density is obtained by marginalising the landmark parameters which introduces links, relative pose constraints, between any two poses measuring the same landmark and the result is a pose graph. This is in close relation to the reduced system in (3.38a) obtained using the Schur complement. GraphSLAM maintains a pose graph as a Gaussian with mean and covariance but only computes the conditional mean of the map using the factorisation

$$p\left(m \,|x_{0:t}, y_{0:t}\right) = \prod_{i=1}^{N_m} p\left(m^i \,|x_{0:t}, y_{0:t}\right), \quad (4.5)$$

which is the same approach as in FastSLAM. This means that each landmark can be treated independently given the true trajectory, thus avoiding to keep track of the full correlation structure of the map which is the main draw-back with EKF-SLAM.

4.2 EKF-SLAM

EKF-SLAM is probably the most common SLAM method and it is often straightforward to implement as described in e.g., Durrant-Whyte and Bailey (2006); Smith et al. (1990). For a thorough treatment, the book by Thrun et al. (2005) serves as a standard reference. In feature based SLAM, coordinates in the global frame are explicitly represented as landmarks, m, which are part of the state vector. The standard assumption is that the landmarks are stationary but dynamic objects can naturally be included in the state vector (Bibby and Reid, 2007). Assume that measurements arrive in the same rate as the dynamic model. Then the landmark and the measurement models are given by

$$x_t = f(x_{t-1}) + w_t, \tag{4.6a}$$
$$m_t = m_{t-1}, \tag{4.6b}$$
$$y_t = h(x_t, m_t) + e_t. \tag{4.6c}$$

The EKF given in Algorithm 1 applies to (4.6) with just a few modifications. The prediction step in EKF-SLAM is given by

$$\hat{x}_{t|t-1} = f(\hat{x}_{t-1|t-1}), \tag{4.7a}$$
$$\hat{m}_{t|t-1} = \hat{m}_{t-1|t-1}, \tag{4.7b}$$
$$P^{xx}_{t|t-1} = F_t P^{xx}_{t-1|t-1} F_t^T + Q, \tag{4.7c}$$
$$P^{xm}_{t|t-1} = F_t P^{xm}_{t-1|t-1}, \tag{4.7d}$$
$$P^{mx}_{t|t-1} = P^{mx}_{t-1|t-1} F_t^T, \tag{4.7e}$$

where

$$F_t \triangleq \left.\frac{\partial f(x_{t-1})}{\partial x_{t-1}}\right|_{(x_{t-1})=(\hat{x}_{t-1|t-1})}. \tag{4.7f}$$

Note that only the vehicle state covariance and the cross terms are updated while that the map mean and covariance remains unchanged. The full covariance matrix is

$$P = \begin{bmatrix} P^{xx} & P^{x\,m} \\ P^{m\,x} & P^{m\,m} \end{bmatrix}. \tag{4.8}$$

When new landmarks are initialised they are appended to the state vector but the vehicle state dimension stays the same. If the map estimation is only used locally for vehicle state estimation i.e., odometry, then old landmarks can be removed

from the filter. The measurement update for EKF-SLAM is given by

$$K_t = P_{t|t-1} H_t^T (H_t P_{t|t-1} H_t^T + R_t)^{-1}, \tag{4.9a}$$

$$\begin{bmatrix} \hat{x}_{t|t} \\ \hat{m}_{t|t} \end{bmatrix} = \begin{bmatrix} \hat{x}_{t|t-1} \\ \hat{m}_{t|t-1} \end{bmatrix} + K_t \left(y_t - h(\hat{x}_{t|t-1}, \hat{m}_{t|t-1}) \right), \tag{4.9b}$$

$$P_{t|t} = P_{t|t-1} - K_t H_t P_{t|t-1}, \tag{4.9c}$$

where

$$H_t \triangleq \left[\frac{\partial}{\partial x_t} h(x_t, m_t) \quad \frac{\partial}{\partial m_t} h(x_t, m_t) \right] \Big|_{(x_t, m_t) = (\hat{x}_{t|t-1}, \hat{m}_{t|t-1})}. \tag{4.10}$$

The measurement Jacobian (4.10) is often rather sparse since the sensor will typically only observe a part of the landmark state at each time instant and an efficient implementation exploits this structure. Since the measurements are assumed independent the measurement update can be processed iteratively avoiding the need for inverting a large matrix in the Kalman gain computation (4.9a).

In Paper B EKF-SLAM is used for intialisation of the trajectory and map which makes it limited to small problems. In Figure 4.2 the horizontal speed estimate from EKF-SLAM and NLS-SLAM is shown.

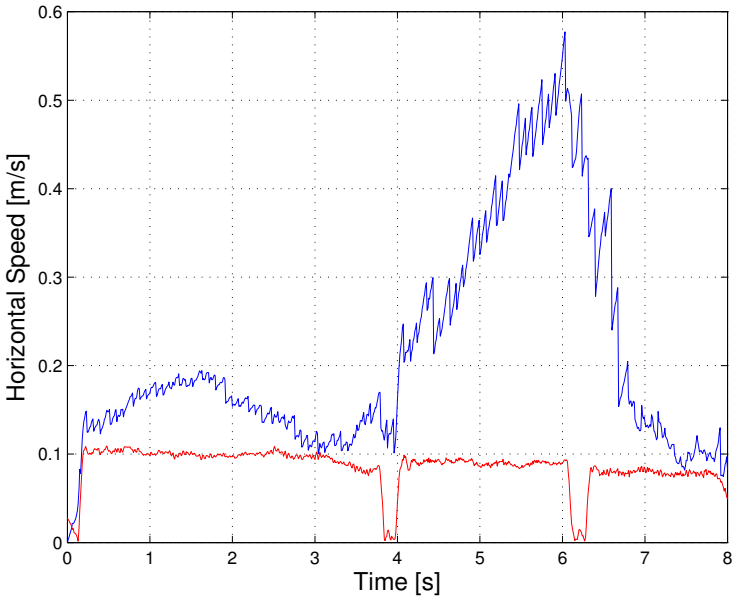


Figure 4.2: *The smoothed horizontal speed of the camera in red and EKF in blue. The true speed is* 0.1 *m/s except for when the robot stops and changes direction, this happens at about 4 seconds and 6 seconds.*

The EKF-SLAM framework also applies to sensor network calibration, as in Paper A. Obviously there is then no need for correspondence search since the sensor identities (landmarks) are known and the map size is also fixed. As was pointed out in Section 2.4 measurements are here obtained in the sensor node magnetometers assuming a known magnetic dipole model of the survey vessel. This is the reverse measurement relation to the ordinary SLAM concept.

4.3 Batch SLAM

Batch methods, also known as Full SLAM, have recently come to dominate SLAM research both offline and as a sub-system in online applications. Some beneficial properties of batch methods are:

- Loop closing and thus drift compensation is easier.
- The effect of linearisation errors can be reduced through iterative refinement.
- Efficient optimisation routines can be utilised.
- Data association decisions can easily be undone.
- The inherent primary sparsity is utilised.
- The complete map correlation structure does not have to (but can be) computed.

In both filtering and batch SLAM applications the system unavoidably grows with the exploited space since more memory is needed to store the map and possibly historical motion estimates. However, in most realistic scenarios only a few parameters are affected by the measurements and therefore only small parts of the system, or its factorisation, need to be updated. A concrete example is the square-root *smoothing and mapping* (SAM) Dellaert and Kaess (2006) which solves the Full SLAM problem incrementally in real-time.

Efficient solutions to batch formulations are utilising the inherent sparsity in the Jacobian J or equivalently the associated information (inverse covariance) matrix $I = JJ^T$. In contrast, the filter covariance, and likewise the filter information matrix, is full since past vehicle states are marginalised as shown in Paskin (2003). The batch sparsity ideas are exploited in a SLAM context by (M. Kaess and A. Ranganathan and F. Dellaert, 2008; Grisetti et al., 2011; Thrun and Montemerlo, 2006; U. Frese, 2005; Paskin, 2003) and many others. Furthermore, Dellaert and Kaess make no special distinction between SAM and BA since they are both often treated as NLS problems in batch form.

4.3.1 Graphs

Recently, optimisation based batch methods on graphs, and especially pose graphs (Lu and Milios, 1997), have attracted much interest and will therefore be given some attention here as well. The GraphSLAM algorithm, proposed by (Thrun

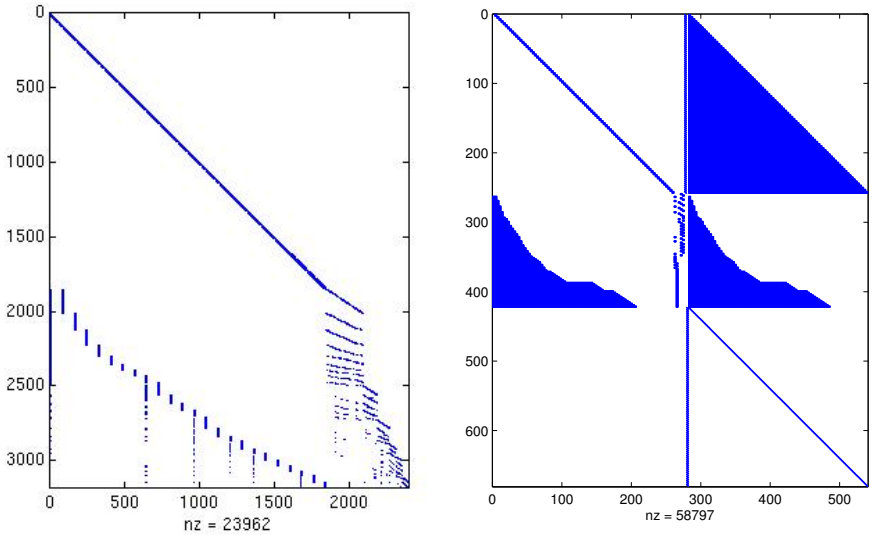


Figure 4.3: *Left: Structure of a Jacobian from the NLS-SLAM problem in Paper B. The upper left diagonal block corresponds to the process derivatives, the left lower block corresponds to the derivatives of the camera measurements w.r.t., the IMU/camera states and the lower right block are derivatives of the camera measurements w.r.t., the landmark parameters. Right: Jacobian matrix from NLS-SLAM problem in Paper D. The structure is more complicated since there are three blocks of measurements (rows) corresponding to the accelerometer measurements, camera measurements and gyroscope measurements, respectively. The parameters (columns) are velocity, acceleration, landmarks, acceleration bias, gyroscope bias, and angular velocity, respectively.*

and Montemerlo, 2006; Thrun et al., 2005), is such a method. It is assumed that the data association is known and there is no general mechanism in GraphSLAM for treating false associations once the graph is constructed. The graph-interpretation is another way of describing dependencies among parameters as in the Jacobian and the Hessian. Examples of two SLAM graphs are shown in Figure 4.3. For instance, the information matrix associated with the Full SLAM posterior is the graph of a *Markov fandom field* (Thrun et al., 2004). The various graph representations are primarily used to gain insight into the underlying inference problem and to enable efficient solution strategies. For most scenarios the corresponding graph contains loops since some places may be visited several times. Inference on the graph can therefor only be done approximately e.g., with the *sum-product algorithm* or *"loopy" belief propagation*. Junction trees eliminates these cycles by clustering them into single nodes (Paskin, 2003) leaving a graph which is a tree and thus exact belief propagation is possible.

The full-SLAM and BA graphs are matrices with known structure when correspondences are fixed. In the tutorial on graph-based SLAM (Grisetti et al., 2011) a clear distinction is made between the sensor specific data association used to construct the graph, which they call the *front-end*, and the strategies used to optimise the graphs which are referred to as the *back-end*. We will here refer to the front-end as the *initialisation*.

4.3.2 Initialisation

Depending on the application the initialisation can be anything from cheap to very computationally expensive. Pose graphs are intuitive to use when the sensor model is invertible which is the case with ground robots having laser range scanners in 2D or 3D. By pair-wise matching of the laser scans, e.g., using *iterative closest point* (ICP), see for instance Besl and McKay (1992), local relative poses are obtained. The pose graph nodes are then the trajectory of robot poses and the edges contains observations and odometry. There are also no natural 3D-3D point correspondences when using laser measurements only since measurements do not provide information around the point. Initialisation in visual applications typically consists of feature tracking and essential matrix estimation within a RANSAC loop.

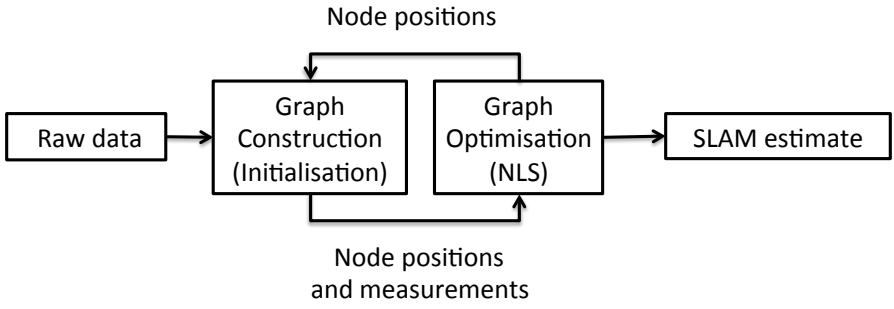


Figure 4.4: *Iterative correspondence search and graph optimisation.*

There are few (if any) visual methods that use pose graphs only. For instance, Kim and Eustice (2009); Eade et al. (2010) use view-based matching for whole images combined with odometry and the relative poses obtained are used to construct a pose graph. Both of them are using view-based recognition based on matching entire frames to each other, similar to ICP, thus they are not doing any local tracking. Strasdat et al. (2011) combines a sub-mapping BA approach much like, PTAM (Klein and Murray, 2007), and connects them via a pose graph in a separate optimisation thread. It is good to incorporate iterative an correspondence search in the optimisation thread, as illustrated in Figure 4.4, since associations may become too unlikely by some metric once the optimisation has started. One such error metric for visual tracking is the bi-directional error (Kalal et al., 2010) which is the difference between the image coordinate of the forward-track and the end image coordinate of the backward-track (reversed point tracking) which should be small for well-localised features (Hedborg et al., 2011, 2012). This idea

is similar to what is used in the Paper D where iterates between state estimation and correspondence search is done. Along the same line Dellaert et al. (2003) employ simultaneous SFM and correspondence estimation using EM to learn 3D models. While a multimodal pose graph model is devised Pfingsthorn and Birk (2012) optimising correspondences using Gaussian mixtures which admits corrections for poor initialisation.

4.3.3 NLS-SLAM

In state-space formulations of SLAM the map is included in the state vector without process noise

$$x_t = f(x_{t-1}) + w_t, \tag{4.11a}$$
$$m_t = m_{t-1}, \tag{4.11b}$$
$$y_t = h(x_t, m_t) + e_t. \tag{4.11c}$$

This formulation is however equivalent to

$$x_t = f(x_{t-1}) + w_t, \tag{4.12a}$$
$$y_t = h(x_t, m) + e_t, \tag{4.12b}$$

meaning that the map can be excluded from state vector and simply viewed as a parameter. The idea with NLS-SLAM to use an initial estimate of the map and the whole state-space sequence $x_{0:t}$ and then minimise all the measurement errors and the state trajectory errors as the NLS problem

$$\{\hat{x}_{0:t},\ \hat{m}\} = \underset{x_{0:t},\ m}{\arg\min} \quad \|\bar{x}_0 - x_0\|^2_{P^{-1}} + \sum_{i=1}^{t} \|x_i - f(x_{i-1})\|^2_{Q^{-1}} + \sum_{i=1}^{t} \|y_i - h(x_i, m)\|^2_{R^{-1}}. \tag{4.13}$$

In Paper D, the dynamic model is considered being exact, i.e., it is just used to define a static map for the parameter-space and is left out of the optimisation. Then the following NLS problem is obtained

$$\{\hat{x}_{0:t},\ \hat{m}\} = \underset{x_{0:t},\ m}{\arg\min} \quad \sum_{i=0}^{t} \|y_i^m - h_m(x_i, m)\|^2_{R_m^{-1}} + \|y_i^{\boldsymbol{a}} - h_{\boldsymbol{a}}(x_i)\|^2_{R_a^{-1}} + \|y_i^{\boldsymbol{\omega}} - h_{\boldsymbol{\omega}}(x_i)\|^2_{R_\omega^{-1}}, \tag{4.14}$$

where h_m denotes the direct parametrisation (2.47) for camera, $h_{\boldsymbol{a}}$ denotes accelerometer measurement function and $h_{\boldsymbol{\omega}}$ denotes gyroscope measurements. Besides the IMU measurements, this is a standard BA formulation. Since both the camera and accelerometer measurement functions are nonlinear w.r.t., the state and the map, a good initial value is needed. It is the sole purpose of Paper D to show how such an initial point can be obtained through a sequence of almost linear steps with only SIFT features and IMU data as input.

4.3.4 EM-SLAM

In Paper E the map is also considered being a static parameter, $\theta := m$, according to

$$x_t = f(x_{t-1}, u_t, w_t), \tag{4.15a}$$
$$y_t = h_t(x_t, \theta) + e_t, \tag{4.15b}$$

where the IMU is input to the motion model. In an ML setting, the joint likelihood of measurement and states parametrising the map is

$$p_\theta(y_{1:t}, x_{1:t}) = \prod_{i=1}^{t} p_\theta(y_i|x_i)p(x_i|x_{i-1}), \tag{4.16}$$

where the state trajectory is considered being a latent variable. This density can be maximised using expectation-maximisation (Dempster et al., 1977) by solving two coupled, but hopefully easier, problems iteratively. This is done by computing the expected value of the log of (4.16)

$$\mathcal{Q}(\theta, \theta_k) = \mathbf{E}_{\theta_k}\left\{\log\left[\prod_{i=1}^{t} p_\theta(y_i|x_i)p(x_i|x_{i-1})\right]\middle| y_{1:t}\right\} = \tag{4.17}$$
$$= -\sum_{t=1}^{N} \mathbf{E}_{\theta_k}\left\{\frac{1}{2}\|y_i - h_t(x_i, \theta)\|^2_{R_m^{-1}}\middle| y_{1:t}\right\} + \text{const.}, \tag{4.18}$$

where the measurement errors are assumed Gaussian. The motion model and other terms independent of the map are lumped into a constant parameter. The expected value cannot be obtained in closed form, instead we approximate (4.17) as

$$\mathcal{Q}(\theta, \theta_k) \approx -\frac{1}{2}\sum_{i=1}^{N}\Big(\|y_i - h_i(\hat{x}_{i|t}, \theta)\|^2_{R_m^{-1}} + \\ \operatorname{Tr}(R^{-1}\nabla_x h_i(\hat{x}_{i|t}, \theta)P^s_{i|t}(\nabla_x h_t(\hat{x}_{i|t}, \theta))^T)\Big), \tag{4.19}$$

where, ∇_x denotes the Jacobian w.r.t., to x, $\hat{x}_{i|t}$ is the smoothed estimate of the latent variable and $P^s_{i|t}$ is its covariance. The smoothed estimate is obtained with an E-RTS smoother. The trace term compensates for the use of the estimated latent variables instead of the true ones. The $\mathcal{Q}(\theta, \theta_k)$ function is then maximised w.r.t., the map θ using the quasi-Newton method BFGS, see e.g., Nocedal and Wright (2006) as further explained in Section 3.2 in Paper E. To start the EM-SLAM iterations an intial estimate of the map and the state is needed and it is here obtained using the results in Paper D. It should be noted that the splitting of batch SLAM into state estimation and mapping is also the key in GraphSLAM and FastSLAM as discussed in Section 4.1.1.

4.4 IMU and Camera

Fusion of IMU and monocular vision information is conceptually straightforward but can be a challenging task in practice. A nice introduction to this sensor setup is given in (Corke et al., 2007) and the thorough exposition in Hol (2011) is also recommended. The complementary characteristics of these sensors are attractive since the unbounded error growth in position and orientation from IMU integration can be corrected using the camera.

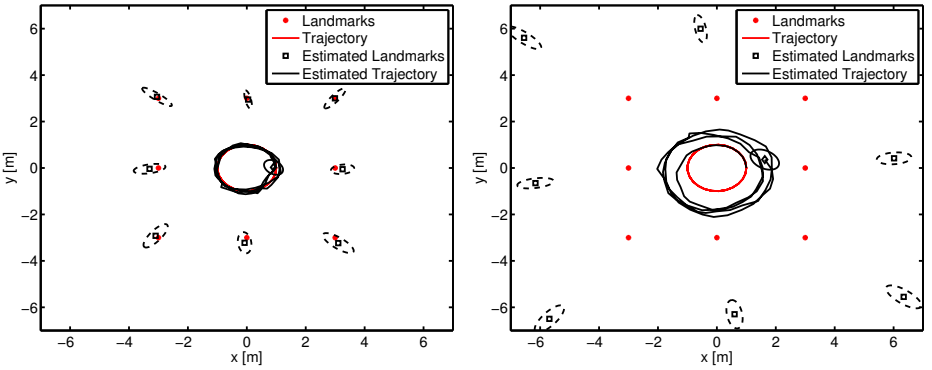


Figure 4.5: *Bearings-only SLAM example illustrating the depth ambiguity. A similar example can also be found in Bailey (2003) but here only the initial velocity* 0.25m/s *of the platform is known. Left: Landmarks are initialised at approximately the true range and the resulting estimate is consistent. Right: The landmarks are initialised at two times the true range and the estimated trajectory scales accordingly giving a biased estimate. This corresponds to the velocity estimate of the platform being* 0.5m/s.

A calibrated stereo camera gives the ability to directly measure 3D coordinates of landmarks in metrical space. However, the baseline between cameras in a stereo rig are physically limited, thus the depth resolution is often limited to close range scenarios since the range resolution itself decreases with range. Motion-stereo does not have this limitation since the baseline is created arbitrary by motion. Obvious downsides of motion-stereo is the need for recovery of the pose, e.g., using epipolar geometry, and the inability to estimate the scale which is due to the depth ambiguity which is illustrated in Figure 4.5. A comparison of stereo and monocular vision based SLAM approaches are presented in Lemaire et al. (2007) and the introductory chapters in Chli (2009) gives a nice overview of vision based SLAM in general.

Apart from SLAM applications the combination of an IMU and monocular camera can be used for:

- IMU supported *structure from motion* (SFM);
- vision supported inertial navigation, i.e., ego-motion estimation;

- loosly or tightly coupled SFM;
- camera IMU fusion for augmented realitys and;
- image stabilisation.

In the context of SLAM, observability is a non-negligible issue and some examples are the case of constant velocity Bryson and Sukkarieh (2008), motion along the optical axis or if landmarks are far away. Short term unobservability is not problematic in filtering problems but may lead to complete failure in batched solutions due to rank deficiency. Simple means for avoiding some of these problems are to use measurement counters. It is however non-trivial to characterise the unobservable subspace in SLAM completely which the substantial amount of publications on this issue is a proof of, see e.g., (Lupton and Sukkarieh, 2012; Hesch et al., 2013; Perera et al., 2009; Kim and Sukkarieh, 2004; Andrade-Cetto and Sanfeliu, 2005; Lee et al., 2006; Wang and Dissanayake, 2008) and the many references therein. Observability is also highlighted in the vision aided IMU systems of Mourikis and Roumeliotis (2007); Mourikis et al. (2009); Li and Mourikis (2013, 2012a); Dong-Si and Mourikis (2012); Li and Mourikis (2012b) which operates without explicit landmark parametrisation and instead keeps old poses in a sliding window expressed as constraints through shared observations.

4.4.1 Linear Triangulation

A monocular camera in combination with an IMU can be used to estimate the pose and to recover the metric scale, a topic which have studied by Martinelli (2012); Nützi et al. (2011); Kneip et al. (2011) and others. It was showed in Martinelli (2012) that given measurements of a landmark from five distinct vantage points with known rotation it is possible to recover the landmark position, the camera position and velocity, and accelerometer bias in closed form. His results are based on well-known linear formulations of the direct parametrisation (2.48) but without considering noise. This formulation was used in Paper D including noise terms and will now be explained. Camera coordinates and the pinhole projection are given by

$$\boldsymbol{m}^c = \mathrm{R}^{cw}(\boldsymbol{m}^w - \boldsymbol{c}_t^w), \tag{4.20a}$$

$$P(\boldsymbol{m}^c) = \frac{1}{z^c}\begin{bmatrix} x^c \\ y^c \end{bmatrix}, \tag{4.20b}$$

which also can be written as a linear form. With normalised measurements $[u, v]^T$ we have

$$\begin{bmatrix} u \\ v \end{bmatrix} z^c = \begin{bmatrix} x^c \\ y^c \end{bmatrix}, \tag{4.21}$$

which is linear in the unknown $\boldsymbol{m}^c$ and also in $\boldsymbol{m}^w$ and $\boldsymbol{c}_t^w$ if the rotation is known. We can solve (4.21) w.r.t., camera and landmark coordinates with enough measurements available. However, for each new 2D camera measurement there is also another unknown 3D camera position. A solution is to use approximate positions calculated from the accelerometer. This results in a linear system in the

unknown $\boldsymbol{m}^w$ and we need at least two camera measurement from distinct vantage points to find a solution. In this linear formulation, the camera measurement noise does not reflect depth uncertainty in a proper manner since we have

$$\begin{bmatrix} u \\ v \end{bmatrix} z^c = \begin{bmatrix} x^c \\ y^c \end{bmatrix} + z^c \begin{bmatrix} e_u \\ e_v \end{bmatrix} \Longleftrightarrow \begin{bmatrix} u - e_u \\ v - e_v \end{bmatrix} z^c = \begin{bmatrix} x^c \\ y^c \end{bmatrix}, \tag{4.22}$$

meaning that the noise scales with the depth parameter. This can be treated using *iteratively reweighted least squares* (IRWLS), see e.g., Björck (1996) by updating the camera measurement covariance weighting matrix at each iterate using $\hat{z}^c$ from the previous iterate. The key is that we bypass solving the nonlinear re-projection error minimisation problem by solving a simple linear problem. This method is algebraic in the sense that it does not consider minimisation of the measurement errors since the projective model is not directly used to form the error. However, the linear solution also minimise the re-projection error if the correct weighting is used (Zhang and Kanade, 1998; Hartley and Zisserman, 2004) which here corresponds to that $\hat{z}^c$ is sufficiently close to its true value. This section finishes with an example of IRWLS triangulation, inspired by the paper (Hartley and Sturm, 1997).

Example 4.1: IRWLS Triangulation

Consider two 3×4 camera matrices, P and P' with known rotation and translation both measuring a 3D point $\boldsymbol{m}$. The two cameras and the point are then related by

$$\lambda[u, v, 1]^T = P[\boldsymbol{m}^T 1]^T = Px, \tag{4.23a}$$

$$\lambda'[u', v', 1]^T = P'[\boldsymbol{m}^T 1]^T = P'x. \tag{4.23b}$$

The λ's are unknown scalars accounting for the projective ambiguity. They can be eliminated from the last rows of the two equation systems. After some algebra the unknown 3D point can be expressed as a linear system $Ax = 0$ where $A = [p_1 - up_3,\ p_2 - vp_3,\ p_1' - u'p_3',\ p_2 - v'p_3']^T$ is a 4×4 matrix and the vectors, $p_i,\ p_i',\ i = 1, 2, 3$, are the rows of the camera matrices. This equation system is solved by computing the SVD of A from which the point is recovered from the singular vector $\sigma[\hat{\boldsymbol{m}}^T, 1]$ which corresponds to the smallest singular value σ. More measurements can be added as rows in A to further suppress noise.

The minimum of $\|Ax\|$ does not have a geometric meaning. At the minimum there will be an error $\epsilon = up_3^T - p_1^T x$. The re-projection error of the measured image coordinate u and x is given by $\epsilon' = u - p_1^T x / x p_3^T = \epsilon / p_3^T$. Thus, if the first row of A was multiplied with $w = 1/p_3^T x$ and similarly for the other rows, then the minimum of the linear method would correspond to the one of the nonlinear re-projection error. This is not possible to do since it requires x to be known. Instead, we can iterate the linear procedure starting with unit weights and then use the weights from the previous iteration and hopefully converge to a solution with an error close to the one of the re-projection error. The algorithm can for instance be terminated when the change in the weights is small.

The setup is the following: The point is located in $\boldsymbol{m}^* = [1,\ 1,\ 2]^T$, the first camera

P in the origin, the second camera P' is translated to $= [0,\ 1,\ 30]^T$ and both of them without rotation. Each camera receives 10 measurements subject to noise $e \sim \mathcal{N}(0, 0.01 I_2)$. Results for $M = 100$ simulations and 3 iterations are shown in Table 4.1 where it is clear that the iterations gives an improvement.

Iteration	0	1	2	3
Average RMSE $[\frac{1}{M}\sqrt{\sum_i^M \|\hat{\boldsymbol{m}}_i - \boldsymbol{m}^*\|}]$	0.0274	0.0203	0.0202	0.0202

Table 4.1: *Iteratively reweighted least squares triangulation.*

The strength of this method is that it is simple to compute and no special initialisation is needed, contrary to nonlinear optimisation methods. A disadvantage is that the method may fail to converge (Hartley and Sturm, 1997) if, for example, the baseline is very short which results in a poorly conditioned system.

5

Concluding remarks

This chapter ends the first part of the thesis which consisted of background material. The work in the whole thesis is summarised here with conclusions of the publications in Part II and suggested directions for future work. For detailed conclusions and future work, the reader is referred to each of the appended publication.

5.1 Conclusions

The common denominator for the problems studied in this thesis is that they deal with various aspects of navigation and mapping in unknown environments.

Calibration of sensor networks is important for the network's detection and tracking capabilities. Underwater sensor positions can be difficult to obtain in fast deployment scenarios and sensors can also move due to currents or a non-rigid seabed. An automatic and inexpensive EKF-SLAM method for underwater sensor network positioning without the need for GNSS was presented. Using only magnetic sensors and a vessel with known magnetic signature the sensor positions and the vessel's route was determined. The expected performance of the method and the network was studied using sensitivity- and CRLB analysis on simulated data. This analysis could also be used for sensor network design.

ROV's cannot utilise GNSS for localisation because these signals are greatly attenuated in water. Due to their limited payload capacity, and in order to have a competitive price, the onboard navigation sensors are relatively cheap. We showed that fusion of a complex hydrodynamic model of the ROV with onboard sensor data can improve the navigation performance and the robustness to sensor failure. Experimental results from sea trials showed that in particular the vehicle

speed can be accurately estimated.

The combination of inertial sensor and optical camera can be used both for navigational and mapping purposes. Experiments and simulations indicated that metrically correct estimates can be obtained. The structure of the batch SLAM problem was exploited and solved with NLS and EM utilising efficient optimisation routines. The batch methods requires a good initial estimate for avoiding convergence to a local minimum. A multistage initialisation procedure for batch SLAM was proposed where a series of almost uncoupled and simple problems were solved.

5.2 Future Work

Underwater sensor networks, and sensor networks in general, are important for monitoring and surveillance. The simulated setup in Paper A can certainly benefit from experimental validation which, however, could be rather expensive. A natural extension is to try NLS-SLAM minimisation of the whole data batch to find out if the results can be improved.

Increased navigation performance and robustness w.r.t., disturbances is always desirable. The ROV model in Paper C is used in conjunction with the onboard sensor for this task. The model for rotational dynamics can be improved, perhaps by using speed dependent damping and stronger coupling in the inertia matrix. These, and other, ideas should first be explored using simulations and then preferably in controlled tank tests.

SLAM in general is a mature and diversified field of research. In the near future perhaps the gap between the robotics community and the computer vision community can be shortened. For this to happen, more work on the similarities between the two areas are needed. For instance, it is not that common to see model-based filters in computer vision applications.

System analysis concepts such as observability, controllability and robustness are often difficult to apply to SLAM systems directly. It would therefore be useful to have easy-to-use tools for SLAM system analysis as to guide design and evaluation on a general basis. For the specific application of SLAM with IMU and vision more work on performance bounds, along the line of Nyqvist and Gustafsson (2013), is desirable.

The coupling between filtering and optimisation is an interesting area for more research. Perhaps the convergence rate of Rao-Blackwellized particle filters could be improved by treating some of the nonlinearities using IEKF-L. Iterated smoothers, such as moving horizon estimation further gives the possibility to handle constraints in a systematic manner.

Bibliography

J. Andrade-Cetto and A. Sanfeliu. The effects of partial observability when building fully correlated maps. *IEEE Transactions on Robotics*, 21(4):771–777, Aug. 2005. ISSN 1552-3098. doi: 10.1109/TRO.2004.842342.

M. Axholt, M. Skoglund, S. D. Peterson, M. D. Cooper, T. B. Schön, F. Gustafsson, A. Ynnerman, and S. R. Ellis. Optical see-through head mounted display direct linear transformation calibration robustness in the presence of user alignment noise. In *Proceedings of the 54th Annual Meeting of the Human Factors and Ergonomics Society*, volume 54, pages 2427–2431, San Francisco, CA, USA, 27-1 Sept./Oct. 2010. Human Factors and Ergonomics Society.

M. Axholt, M. A. Skoglund, S. D. O'Connell, M. D. Cooper, S. R. Ellis, and A. Ynnerman. Parameter estimation variance of the single point active alignment method in optical see-through head mounted display calibration. In *Proceedings of the IEEE Virtual Reality Conference*, pages 27–34, Singapore, Republic of Singapore, Mar. 2011.

T. Bailey. Constrained initialisation for bearing-only SLAM. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1966–1971, Taipei, Taiwan, 14-19 Sept. 2003. doi: 10.1109/ROBOT.2003.1241882.

Y. Bar-Shalom and X.-R. Li. *Estimation and tracking : principles, techniques and software*. Artech House, Boston, 1993. ISBN 0-89006-643-4.

B. Bell and F. Cathey. The iterated Kalman filter update as a Gauss-Newton method. *IEEE Transactions on Automatic Control*, 38(2):294–297, Feb. 1993. ISSN 0018-9286. doi: 10.1109/9.250476.

D. Bertsekas. Incremental least squares methods and the extended Kalman filter. In *Proceedings of the 33rd IEEE Conference on Decision and Control*, volume 2, pages 1211–1214, Lake Buena Vista, Florida, USA, 14-16 Dec. 1994. doi: 10.1109/CDC.1994.411166.

P. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Transac-*

tions on Pattern Analysis and Machine Intelligence, 14(2):239–256, Feb. 1992. ISSN 0162-8828. doi: 10.1109/34.121791.

C. Bibby and I. Reid. Simultaneous localisation and mapping in dynamic environments (SLAMIDE) with reversible data association. In *Proceedings of Robotics: Science and Systems*, Atlanta, GA, USA, 27-30 June 2007.

Å. Björck. *Numerical Methods for Least Squares Problems*. SIAM, 1996. ISBN 0-89871-360-9.

M. Bosse and R. Zlot. Map matching and data association for large-scale two-dimensional laser scan-based SLAM. *International Journal of Robotic Research*, 27(6):667–691, 2008.

J.-Y. Bouguet. Camera Calibration Toolbox for Matlab. `www.vision.caltech.edu/bouguetj/calib_doc/`, 2010.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. URL `http://www.stanford.edu/~boyd/bv_cvxbook.pdf`.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, Jan. 2011. ISSN 1935-8237. doi: 10.1561/2200000016. URL `http://dx.doi.org/10.1561/2200000016`.

K. Britting. *Inertial Navigation Systems Analysis*. John Wiley & Sons Inc., New York, USA, 1971.

M. Bryson and S. Sukkarieh. Observability analysis and active control for airborne SLAM. *Aerospace and Electronic Systems, IEEE Transactions on*, 44(1): 261–280, 2008. ISSN 0018-9251. doi: 10.1109/TAES.2008.4517003.

M. Bryson and S. Sukkarieh. Architectures for Cooperative Airborne Simultaneous Localisation and Mapping. *Journal of Intelligent and Robotic Systems*, 55(4-5):267–297, 2009. doi: 10.1007/s10846-008-9303-9. URL `http://dx.doi.org/10.1007/s10846-008-9303-9`.

M. Bryson, A. Reid, F. Ramos, and S. Sukkarieh. Airborne vision-based mapping and classification of large farmland environments. *Journal of Field Robotics*, 27(5):632–655, Sept. 2010. ISSN 1556-4959. doi: 10.1002/rob.v27:5. URL `http://dx.doi.org/10.1002/rob.v27:5`.

F. Caballero, L. Merino, J. Ferruz, and A. Ollero. Vision-based odometry and SLAM for medium and high altitude flying UAVs. *Journal of Intelligent and Robotics Syststems*, 54(1-3):137–161, 2009. ISSN 0921-0296. doi: http://dx.doi.org/10.1007/s10846-008-9257-y.

J. Callmer, K. Granström, J. Nieto, and F. Ramos. Tree of words for visual loop closure detection in urban SLAM. In J. Kim and R. Mahony, editors, *Proceedings of the 2008 Australasian Conference on Robotics & Automation*, Canberra, Australia, 3-5 Dec. 2008.

J. Callmer, M. Skoglund, and F. Gustafsson. Silent localization of underwater sensors using magnetometers. *EURASIP Journal on Advances in Signal Processing*, 2010, 2010. doi: 10.1155/2010/709318. URL http://dx.doi.org/10.1155/2010/709318. Article ID 709318.

J. Callmer, D. Törnqvist, F. Gustafsson, H. Svensson, and P. Carlbom. RADAR SLAM using visual features. *EURASIP Journal on Advances in Signal Processing*, 2011(71), 2011.

M. Chli. *Applying Information Theory to Efficient SLAM*. PhD thesis, Imperial College London, 2009.

J. Choi, S. Ahn, and W. K. Chung. Robust sonar feature detection for the SLAM of mobile robot. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems*, pages 3415–3420, Edmonton, Alberta, Canada, 2-6 Aug. 2005. doi: 10.1109/IROS.2005.1545284.

P. Corke, J. Lobo, and J. Dias. An introduction to inertial and visual sensing. *The International Journal of Robotics*, 26:519–535, 2007.

M. Cummins and P. Newman. Appearance-only SLAM at large scale with FAB-MAP 2.0. *The International Journal of Robotics Research*, 2010. doi: 10.1177/0278364910385483. URL http://ijr.sagepub.com/content/early/2010/11/11/0278364910385483.abstract.

A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proceedings of the 9th IEEE International Conference on computer vision*, pages 1403–1410, Nice, France, 13-16 Oct. 2003.

A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007.

F. Dellaert and M. Kaess. Square Root SAM: Simultaneous location and mapping via square root information smoothing. *International Journal of Robotics Research (IJRR)*, 25(12):1181–1203, 2006. URL http://www.cc.gatech.edu/~dellaert/pubs/Dellaert06ijrr.pdf. Special issue on RSS 2006.

F. Dellaert, S. Seitz, C. Thorpe, and S. Thrun. EM, MCMC, and chain flipping for structure from motion with unknown correspondence. *Machine Learning*, 50 (1-2):45–71, 2003.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977. ISSN 00359246. doi: 10.2307/2984875. URL http://web.mit.edu/6.435/www/Dempster77.pdf.

J. Dennis and R. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Prentice Hall, 1983.

T. Dong-Si and A. I. Mourikis. Initialization in vision-aided inertial navigation with unknown camera-imu calibration. In *Proceedings of the IEEE/RSJ Inter-*

national Conference on Robotics and Intelligent Systems (IROS), pages 1064–1071, Vilamoura, Portugal, October 2012.

H. Durrant-Whyte and T. Bailey. Simultaneous Localization and Mapping: Part I. *IEEE Robotics and Automation Magazine*, 13(12):99–110, June 2006.

E. Eade. *Monocular Simultaneous Localisation and Mapping*. PhD thesis, Cambridge University, 2008.

E. Eade, P. Fong, and M. Munich. Monocular graph SLAM with complexity reduction. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, Taipei, Taiwan, 18-22 Oct. 2010.

R. Eustice, H. Singh, J. Leonard, and M. Walter. Visually mapping the RMS Titanic: Conservative covariance estimates for SLAM information filters. *International Journal of Robotics Research*, 25(12):1223–1242, December 2006.

M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981. ISSN 0001-0782. doi: 10.1145/358669.358692. URL `http://doi.acm.org/10.1145/358669.358692`.

R. A. Fisher. On an absolute criterion for fitting frequency curves. *Messenger of Mathematics*, 41:155–160, 1912.

J. Fossel, D. Hennes, D. Claes, S. Alers, and K. Tuyls. OctoSLAM: A 3D mapping approach to situational awareness of unmanned aerial vehicles. In *Unmanned Aircraft Systems (ICUAS), 2013 International Conference on*, pages 179–188, May 2013. doi: 10.1109/ICUAS.2013.6564688.

T. I. Fossen. *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons Ltd., 2011. ISBN 978-1-1199-9149-6.

F. Fraundorfer and D. Scaramuzza. Visual odometry: Part II: Matching, robustness, optimization, and applications. *IEEE Robotics and Automation Magazine*, 19(2):78–90, 2012. ISSN 1070-9932. doi: 10.1109/MRA.2012.2182810.

F. Gerossier, P. Checchin, C. Blanc, R. Chapuis, and L. Trassoudaine. Trajectory-oriented EKF-SLAM using the Fourier-Mellin transform applied to microwave radar images. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 4925–4930, St. Louis, MO, USA, 11-15 Oct. 2009. doi: 10.1109/IROS.2009.5354548.

G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard. A tutorial on graph-based SLAM. 2(4):31–43, 2011. ISSN 1939-1390. doi: 10.1109/MITS.2010.939925.

F. Gustafsson. *Statistical Sensor Fusion*. Utbildningshuset/Studentlitteratur, 2012. ISBN 9789144077321.

S. W. Hamilton. On quaternions; or on a new system of imaginaries in algebra. *Philosophical Magazine*, xxv:10–13, July 1844.

J. Han and J. Kim. Navigation of an unmanned surface vessel under bridges. In *Proceedings of the International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, pages 206–210, Jeju, Korea, 3-2 Oct./Nov. 2013. doi: 10.1109/URAI.2013.6677343.

C. Harris and M. Stephens. A combined corner and edge detector. In *proceedings of the 4th Alvey Conference*, pages 147–151, Manchester, UK, 1988.

R. I. Hartley. Chirality. *International Journal of Computer Vision*, 26(1):41–61, 1998.

R. I. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146 – 157, 1997. ISSN 1077-3142. doi: 10.1006/cviu.1997.0547. URL http://www.sciencedirect.com/science/article/pii/S1077314297905476.

R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004. ISBN 0-521-54051-8.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer-Verlag, 2 edition, 2009.

J. Hedborg, E. Ringaby, P.-E. Forssen, and M. Felsberg. Structure and motion estimation from rolling shutter video. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 17–23, Barcelona, Spain, Nov. 2011. doi: 10.1109/ICCVW.2011.6130217.

J. Hedborg, P.-E. Forssén, M. Felsberg, and E. Ringaby. Rolling shutter bundle adjustment. In *IEEE Conference on Computer Vision and Pattern Recognition*, Providence, Rhode Island, USA, June 2012. ISBN 978-1-4673-1227-1. http://dx.doi.org/10.1109/CVPR.2012.6247831.

J. A. Hesch, D. G. Kottas, S. L. Bowman, and S. I. Roumeliotis. Camera-IMU-based localization: Observability analysis and consistency improvement. *The International Journal of Robotics Research*, 2013. doi: 10.1177/0278364913509675. URL http://ijr.sagepub.com/content/early/2013/11/13/0278364913509675.abstract.

J. Hol. *Sensor Fusion and Calibration of Inertial Sensors, Vision, Ultra-Wideband and GPS*. Linköping studies in science and technology. Dissertations. no. 1368, Linköping University, The Institute of Technology, June 2011.

H. Hopf. Systeme symmetrischer Bilinearformen und euklidische Modelle der projektiven Räume. *Vierteljahrsshrift der Naturforschenden Gesellenschaft in Zürich*, 85(Beiblatt (Festschrift Rudolf Fueter)):165–177, 1940.

A. H. Jazwinski. *Stochastic Processes and Filtering Theory*, volume 64 of *Mathematics in Science and Engineering*. Academic Press, Inc, 1970.

K. Jönsson. Position Estimation of Remotely Operated Underwater Vehicle. Master's thesis, Linköping University, 2010.

I.-K. Jung and S. Lacroix. Simultaneous localization and mapping with stereovision. In *Proceedings of The Eleventh International Symposium Robotics Research*, volume 15 of *Springer Tracts in Advanced Robotics*, pages 315–324, Siena, Italy, 19-22 Oct. 2003. Springer. ISBN 978-3-540-23214-8.

S.-H. Jung and C. Taylor. Camera trajectory estimation using inertial sensor measurements and structure from motion results. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages II–732–II–737 vol.2, 2001. doi: 10.1109/CVPR.2001.991037.

T. Kailath, A. H. Sayed, and B. Hassibi. *Linear Estimation*. Prentice-Hall, Upper Saddle River, New Jersey, 2000.

Z. Kalal, K. Mikolajczyk, and J. Matas. Forward-backward error: Automatic detection of tracking failures. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 2756–2759, Istanbul, Turkey, 23-26 Aug. 2010. doi: 10.1109/ICPR.2010.675.

A. Karlsson and J. Bjärkefur. Simultaneous Localisation and Mapping of Indoor Environments Using a Stereo Camera and a Laser Camera. Master's thesis, Department of Electrical Engineering, Linköping University, Linköping, Sweden, 2010.

R. Karlsson, T. Schön, D. Törnqvist, G. Conte, and F. Gustafsson. Utilizing model structure for efficient simultaneous localization and mapping for a UAV application. In *Proceedings of the IEEE Aerospace Conference*, Big Sky, MT, USA, Mar. 2008.

A. Kim and R. Eustice. Pose-graph visual SLAM with geometric model selection for autonomous underwater ship hull inspection. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1559–1565, St. Louis, MO, USA, 11-15 Oct. 2009. doi: 10.1109/IROS.2009.5354132.

J. Kim and S. Sukkarieh. Improving the real-time efficiency of inertial SLAM and understanding its observability. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 21–26, Sendai, Japan, 28-2 Sept./Oct. 2004. doi: 10.1109/IROS.2004.1389323.

R. Kim, Ayoungand Eustice. Real-time visual SLAM for autonomous underwater hull inspection using visual saliency. *IEEE Transactions on Robotics*, 29(3): 719–733, June 2013. ISSN 1552-3098. doi: 10.1109/TRO.2012.2235699.

G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 225–234, Nara, Japan, 2007.

L. Kneip, A. Martinelli, S. Weiss, D. Scaramuzza, and R. Siegwart. Closed-form solution for absolute scale velocity determination combining inertial measurements and a single feature correspondence. In *Proceedings of the International*

Conference on Robotics and Automation, Shanghai, China, May 2011. URL http://hal.inria.fr/hal-00641772.

K. Konolige and M. Agrawal. FrameSLAM: From bundle adjustment to real-time visual mapping. *IEEE Transactions on Robotics*, 24(5):1066–1077, 2008.

J. Kuipers. *Quaternions and Rotations Sequences*. Princeton University Press, 2002.

K. W. Lee, W. Wijesoma, and I. Javier. On the observability and observability analysis of SLAM. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3569–3574, Beijing, China, 9-15 Oct. 2006. doi: 10.1109/IROS.2006.281646.

T. Lemaire, C. Berger, I.-K. Jung, and S. Lacroix. Vision-based SLAM: Stereo and monocular approaches. *International Journal of Computer Vision*, 74(3):343–364, Sept. 2007. ISSN 0920-5691. doi: 10.1007/s11263-007-0042-3. URL http://dx.doi.org/10.1007/s11263-007-0042-3.

H. Li and R. Hartley. Five-point motion estimation made easy. In *Proceedings of the International Conference on Pattern Recognition*, volume 1, pages 630–633, Hong Kong, 20-24 Aug. 2006. doi: 10.1109/ICPR.2006.579.

M. Li and A. I. Mourikis. Improving the accuracy of EKF-based visual-inertial odometry. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 828–835, Minneapolis, MN, May 2012a.

M. Li and A. I. Mourikis. Optimization-based estimator design for vision-aided inertial navigation. In *Proceedings of Robotics: Science and Systems*, Sydney, Australia, July 2012b.

M. Li and A. I. Mourikis. High-precision, consistent EKF-based visual-inertial odometry. *International Journal of Robotics Research*, 32(6):690–711, May 2013. ISSN 0278-3649. doi: 10.1177/0278364913481251. URL http://dx.doi.org/10.1177/0278364913481251.

H. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135, Sept. 1981.

D. Lowe. Object Recognition from Local Scale-Invariant Features. In *Proceedings of the Seventh International Conference on Computer Vision (ICCV'99)*, pages 1150–1157, Corfu, Greece, 1999.

F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997.

C. Lundquist, M. A. Skoglund, K. Granström, and T. Glad. Insights from implementing a system for peer-review. *IEEE Transactions on Education*, 56(3): 261–267, 2013.

T. Lupton and S. Sukkarieh. Removing scale biases and ambiguity from 6DoF monocular SLAM using inertial. In *Proceedings of the International Confer-*

ence on Robotics and Automation (ICRA), pages 3698–3703, Pasadena, California, USA, 2008. IEEE. doi: 10.1109/ROBOT.2008.4543778. URL http://dx.doi.org/10.1109/ROBOT.2008.4543778.

T. Lupton and S. Sukkarieh. Efficient integration of inertial observations into visual SLAM without initialization. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1547–1552, St. Louis, MO, USA, 11-15 Oct. 2009. IEEE. doi: 10.1109/IROS.2009.5354267. URL http://dx.doi.org/10.1109/IROS.2009.5354267.

T. Lupton and S. Sukkarieh. Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions. *IEEE Transactions on Robotics*, 28(1):61 –76, Feb. 2012. ISSN 1552-3098. doi: 10.1109/TRO.2011.2170332.

M. Kaess and A. Ranganathan and F. Dellaert. iSAM: Incremental smoothing and mapping. *IEEE Transactions on Robotics*, 24(6):1365–1378, Dec. 2008.

Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer Verlag, 2003. ISBN 0387008934.

I. Mahon, S. B. Williams, O. Pizarro, and M. Johnson-Roberson. Efficient view-based SLAM using visual loop closures. *IEEE Transactions on Robotics*, 24(5):1002–1014, 2008.

J. Marck, A. Mohamoud, E. v.d.Houwen, and R. van Heijster. Indoor radar SLAM a radar application for vision and gps denied environments. In *Proceedings of the European Radar Conference (EuRAD)*.

D. W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM Journal on Applied Mathematics*, 11(2):431–441, 1963. doi: 10.1137/0111030. URL http://dx.doi.org/10.1137/0111030.

A. Martinelli. Vision and imu data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination. *IEEE Transactions on Robotics*, 28(1):44 –60, Feb. 2012. ISSN 1552-3098. doi: 10.1109/TRO.2011.2160468.

J. Mattingley and S. Boyd. Real-time convex optimization in signal processing. *IEEE Signal Processing Magazine*, 27(3):50–61, 2010. ISSN 1053-5888. doi: 10.1109/MSP.2010.936020.

M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, page 593–598, Edmonton, Alberta, Canada, 28-1 July/Aug. 2002.

M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, Acapulco, Mexico, 9-15 Aug. 2003.

J. Montiel and A. Davison. A visual compass based on SLAM. In *Proceedings the IEEE International Conference on Robotics and Automation (ICRA)*, pages 1917–1922, Orlando, Florida, USA, May 2006. doi: 10.1109/ROBOT.2006.1641986.

A. Mourikis and S. Roumeliotis. A multi-state constraint Kalman filter for vision-aided inertial navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, Roma, Italy, 10-14 Apr. 2007.

A. I. Mourikis, N. Trawny, S. I. Roumeliotis, A. E. Johnson, A. Ansar, and L. Matthies. Vision-aided inertial navigation for spacecraft entry, descent, and landing. *IEEE Transactions on Robotics*, 25(2):264 – 280, 2009. ISSN 15523098. URL `https://lt.ltag.bibl.liu.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=buh&AN=38716223&site=ehost-live`.

J. Mullane, S. Keller, A. Rao, M. Adams, A. Yeo, F. Hover, and N. Patrikalakis. X-band radar based SLAM in Singapore's off-shore environment. In *Proceedings of the International Conference on Control Automation Robotics Vision (ICARCV)*, Singapore, 7-10 Dec. 2010.

P. M. Newman, J. J. Leonard, and R. J. Rikoski. Towards constant-time SLAM on an autonomous underwater vehicle using synthetic aperture sonar. In *Proceedings of the Eleventh International Symposium on Robotics Research*, pages 409–420, Siena, Italy, 19-22 Oct. 2003. Springer Verlag.

E. Nilsson. An optimization based approach to visual odometry using infrared images. Master's thesis, Department of Electrical Engineering, Linköping University, Linköping, Sweden, 2010.

E. Nilsson, C. Lundquist, T. B. Schön, D. Forslund, and J. Roll. Vehicle motion estimation using an infrared camera. In *Proceedings of the IFAC World Congress*, Milan, Italy, Aug./Sept. 2011.

D. Nistér. Reconstruction from uncalibrated sequences with a hierarchy of trifocal tensors. In *Proceedings of the 6th European Conference on Computer Vision-Part I*, ECCV '00, pages 649–663, London, UK, UK, 2000. Springer-Verlag. ISBN 3-540-67685-6. URL `http://dl.acm.org/citation.cfm?id=645313.649415`.

D. Nistér. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–770, 2004. ISSN 0162-8828. doi: 10.1109/TPAMI.2004.17.

J. Nocedal. Updating quasi-Newton matrices with limited storage. *Mathematics of Computation*, 35(151):773–782, 1980. URL `http://www.jstor.org/stable/2006193`.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 2nd edition, 2006. ISBN 978-0-387-40065-5.

G. Nützi, S. Weiss, D. Scaramuzza, and R. Siegwart. Fusion of IMU and vision for absolute scale estimation in monocular SLAM. *Journal of Intelligent Robotics Systems*, 61:287–299, Jan. 2011. ISSN 0921-0296. doi: http://dx.doi.org/10.1007/s10846-010-9490-z. URL http://dx.doi.org/10.1007/s10846-010-9490-z.

H. Nyqvist and F. Gustafsson. A high-performance tracking system based on camera and imu. In *Proceedings of the 16th International Conference on Information Fusion*, Istanbul, Turkey, 9-12 July 2013.

M. Paskin. Thin junction tree filters for simultaneous localization and mapping. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03)*, pages 1157–1164, San Francisco, CA, 2003.

L. Perera, A. Melkumyan, and E. Nettleton. On the linear and nonlinear observability analysis of the SLAM problem. pages 1–6, Málaga, Spain, 14-17 Apr. 2009. doi: 10.1109/ICMECH.2009.4957168.

M. Pfingsthorn and A. Birk. Simultaneous localization and mapping (slam) with multimodal probability distributions. *The International Journal of Robotics Research*, 2012. doi: 10.1177/0278364912461540. URL http://ijr.sagepub.com/content/early/2012/10/05/0278364912461540.abstract.

P. Pinies, T. Lupton, S. Sukkarieh, and J. Tardos. Inertial aiding of inverse depth SLAM using a monocular camera. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2797–2802, Roma, Italy, 10-14 Apr. 2007.

C. Rao. *Moving Horizon Strategies for the Constrained Monitoring and Control of Nonlinear Discrete-Time Systems*. PhD thesis, University of Wisconsin Madison, 2000.

H. E. Rauch, S. C. T., and T. F. Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, 3(8):1445–1450, 1965.

D. Ribas, P. Ridao, J. Neira, and J. Tardos. SLAM using an imaging sonar for partially structured underwater environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, 9-15 Oct. 2006.

D. Scaramuzza and F. Fraundorfer. Visual odometry : Part I: The first 30 years and fundamentals. *IEEE Robotics and Automation Magazine*, 18(4):80–92, 2011. ISSN 1070-9932. doi: 10.1109/MRA.2011.943233.

D. Scaramuzza, A. Martinelli, and R. Siegwart. A toolbox for easily calibrating omnidirectional cameras. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Beijing, China, 9-15 Oct. 2006.

J. Shi and C. Tomasi. Good features to track. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*,

pages 593–600, Seattle, WA, USA, 21-23 June 1994. doi: 10.1109/CVPR.1994.323794.

M. D. Shuster. Survey of attitude representations. *Journal of the Astronautical Sciences*, 41(4):439–517, 1993.

M. D. Shuster. Constraint in attitude estimation part II: Unconstrained estimation. *The Journal of the Astronautical Sciences*, 51(1):75–101, Jan.–Mar. 2003.

Z. Sjanic and F. Gustafsson. Simultaneous navigation and SAR auto-focusing. In *Proceedings of the International Conference on Information Fusion (FUSION)*, pages 1–7, Edinburgh, UK, 26-29 July 2010.

Z. Sjanic, M. A. Skoglund, F. Gustafsson, and T. B. Schön. A nonlinear least squares approach to the SLAM problem. In *Proceedings of the IFAC World Congress*, volume 18, Milan, Italy, 28-2 Aug./Sept. 2011.

Z. Sjanic, M. A. Skoglund, and F. Gustafsson. EM-SLAM with inertial/visual applications. Submitted to IEEE Transactions on Aerospace and Electronic Systems, June 2014.

M. A. Skoglund, K. Jönsson, and F. Gustafsson. Modeling and sensor fusion of a remotely operated underwater vehicle. In *Proceedings of the 15th International Conference on Information Fusion (FUSION), Singapore, 9-12 July 2012*, pages 947–954. IEEE, 2012.

M. A. Skoglund, Z. Sjanic, and F. Gustafsson. Initialisation and estimation methods for batch optimisation of inertial/visual SLAM. Submitted to IEEE Transactions on Aerospace and Electronic Systems, June 2014.

R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In *Autonomous robot vehicles*, pages 167–193. Springer-Verlag New York, Inc., New York, NY, USA, 1990. ISBN 0-387-97240-4.

H. Strasdat, A. Davison, J. M. M. Montiel, and K. Konolige. Double window optimisation for constant time visual SLAM. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Barcelona, Spain, 6-13 Nov. 2011.

J. Stuelpnagel. On the parametrization of the three-dimensional rotation group. *SIAM Review*, 6(4):pp. 422–430, 1964. ISSN 00361445. URL `http://www.jstor.org/stable/2027966`.

T. Thormählen, N. Hasler, M. Wand, and H.-P. Seidel. Merging of Feature Tracks for Camera Motion Estimation from Video. In *Proceedings of the 5th European Conference on Visual Media Production (CVMP 2008)*, pages 1 – 8, Hertfordshire, UK, Jan. 2008. IET.

S. Thrun and M. Montemerlo. The GraphSLAM algorithm with applications to large-scale mapping of urban structures. *International Journal on Robotics Research*, 25(5):403–430, 2006.

S. Thrun, Y. Liu, D. Koller, A. Ng, Z. Ghahramani, and H. Durrant-Whyte. Simultaneous localization and mapping with sparse extended information filters. *International Journal of Robotics Research*, 23(7/8), 2004.

S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005. ISBN 0262201623.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B)*, 58:267–288, 1996.

D. Titterton and J. Weston. *Strapdown Inertial Navigation Technology*. IEE radar, sonar, navigation and avionics series. Peter Peregrinus Ltd., Stevenage, UK, 1997. ISBN 0863413587.

D. Törnqvist. *Estimation and Detection with Applications to Navigation*. Linköping Studies in Science and Technology. Dissertations. No. 1216, Linköping University, Nov. 2008.

P. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78(1):138 – 156, 2000. ISSN 1077-3142. doi: 10.1006/cviu.1999.0832. URL `http://www.sciencedirect.com/science/article/pii/S1077314299908329`.

U. Frese. Treemap: An $O(log\ n)$ algorithm for simultaneous localization and mapping. In C. Freksa, editor, *Spatial Cognition IV*, pages 455–476. Springer Verlag, 2005. URL `http://www.informatik.uni-bremen.de/~ufrese/slamvideos1_e.html`.

N. Wahlström. *Localization using Magnetometers and Light Sensors*. Linköping Studies in Science and Technology. Thesis No. 1581, Linköping University, Sweden, 2013.

Z. Wang and G. Dissanayake. Observability analysis of SLAM using Fisher information matrix. In *Proceedings of the International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 1242–1247, Hanoi, Vietnam, 17-20 Dec. 2008. doi: 10.1109/ICARCV.2008.4795699.

Z. Wang and G. Dissanayake. Exploiting vehicle motion information in monocular SLAM. In *Proceedings of the International Conference on Control Automation Robotics Vision (ICARCV)*, pages 1030–1035, Guangzhou, China, 5-7 Dec. 2012. doi: 10.1109/ICARCV.2012.6485299.

W. Wijesoma, L. Perera, and M. Adams. Toward multidimensional assignment data association in robot localization and mapping. *IEEE Transactions on Robotics*, 22(2):350 – 365, Apr. 2006. ISSN 1552-3098. doi: 10.1109/TRO.2006.870634.

K. Wrobel. SLAM-based approach to dynamic ship positioning. *TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation*, 8(1):21–25, 2014.

Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, Nov. 2000. ISSN 0162-8828. doi: 10.1109/34.888718.

Z. Zhang and T. Kanade. Determining the epipolar geometry and its uncertainty: A review. *International Journal of Computer Vision*, 27:161–195, 1998.

Part II

Publications

Paper A

Silent Localization of Underwater Sensors using Magnetometers

Authors: Jonas Callmer, Martin Skoglund and Fredrik Gustafsson

Edited version of the paper:

J. Callmer, M. Skoglund, and F. Gustafsson. Silent localization of underwater sensors using magnetometers. *EURASIP Journal on Advances in Signal Processing*, 2010, 2010. doi: 10.1155/2010/709318. URL http://dx.doi.org/10.1155/2010/709318. Article ID 709318.

Silent Localization of Underwater Sensors using Magnetometers

Jonas Callmer, Martin Skoglund and Fredrik Gustafsson

Dept. of Electrical Engineering,
Linköping University,
SE–581 83 Linköping, Sweden
{callmer, ms, fredrik}@isy.liu.se

Abstract

Sensor localization is a central problem for sensor networks. If the sensor positions are uncertain, the target tracking ability of the sensor network is reduced. Sensor localization in underwater environments is traditionally addressed using acoustic range measurements involving known anchor or surface nodes. We explore the usage of triaxial magnetometers and a friendly vessel with known magnetic dipole to silently localize the sensors. The ferromagnetic field created by the dipole is measured by the magnetometers and is used to localize the sensors. The trajectory of the vessel and the sensor positions are estimated simultaneously using an Extended Kalman Filter (EKF). Simulations show that the sensors can be accurately positioned using magnetometers.

1 Introduction

Today, surveillance of ports and other maritime environments is getting increasingly important for naval and customs services. Surface vessels are rather easy to detect and track, unlike submarines and other underwater vessels which pose new threats such as terrorism and smuggling. To detect these vessels, an advanced underwater sensor network is necessary. Such sensors can measure fluctuations in for example magnetic and electric fields, pressure changes and acoustics.

Deploying an underwater sensor in its predetermined position can be difficult due to currents, surge and the lack of a Global Navigation Satellite System (GNSS) functioning underwater. Sometimes the sensors must be deployed fast, resulting in very uncertain sensor positions. These positions must then be estimated in order to enable the network to accurately track an alien vessel.

Lately, many solutions to the underwater sensor localization problem have been suggested. They can be broadly divided into two major categories; range-based

and range-free. In general, range-based schemes provide more accurate positioning than range-free schemes.

Range-based schemes use information about the range or angle between sensors. The problem is thereafter formulated as a multilateral problem. Common methods to measure range or angle include Time of Arrival (ToA), Time Difference of Arrival (TDoA), Angle of Arrival (AoA) or Received Signal Strength Indicator (RSSI). These methods usually require active pinging but silent methods based on TDoA have been suggested Cheng et al. (May. 2008). The 3D positioning problem can be transformed into a 2D problem by the use of depth sensors Cheng et al. (2008). The range positioning scheme is often aided by surface nodes, anchor nodes, mobile beacons or autonomous underwater vehicles Zhou et al. (2007); Erol et al. (2008, 2007). Joint sensor localization and time synchronization was performed in Tian et al. (2007).

Range-free schemes generally provide a coarse estimate of a node's location and their main advantage lie in their simplicity. Examples of range-free schemes are Density-aware Hop-count Localization (DHL) Wong et al. (2005) and Area Localization Scheme (ALS) Chandrasekhar and Seah (2006). A more thorough description of underwater sensor localization solutions, can be found in the surveys Akyildiz et al. (2005); Chandrasekhar et al. (2006).

In this paper we propose a method to silently localize underwater sensors equipped with triaxial magnetometers using a friendly vessel with known static magnetization characteristics. (For methods to estimate the magnetic characteristics, see Nelson and Richards (2007).) Each sensor is further equipped with a pressure sensor and an accelerometer used for depth estimation and sensor orientation estimation, respectively. To enable global positioning of the sensors, the vessel or one sensor must be positioned globally. To the best of the authors knowledge this is the first time magnetic dipole tracking is used for sensor localization.

For target tracking in shallow waters, magnetometers are often a more useful sensor than acoustics, since sound scatters significantly in these environments Birsan (2006). Birsan has explored the use of magnetometers and the magnetic dipole of a vessel for target tracking Birsan (2004, 2005). Two sensors with known positions were used to track a vessel while simultaneously estimating the unknown magnetic dipole of the vessel. Tracking and estimation were performed using an unscented Kalman filter Birsan (2004) and an unscented particle filter Birsan (2005). Dahlberg *et al.* Dalberg et al. (2006) fused electromagnetic and acoustic data to track surface vessels using underwater sensors.

Several studies of the electromagnetic characteristics of the maritime environment have stated that the permeability of the seabed differs considerably from the permeability of air or water. The environment should therefore be modeled as a horizontally stratified model with site specific permeability and layer thickness for each segment Dalberg et al. (2006); Daya et al. (2005); Birsan (2006). This has not been included in our simulation study but should be considered in field experiments.

In the past 10-15 years quite a lot of effort has been put into reducing the static magnetic signature of naval vessels by active signature cancelling. This has increased the importance of other sources of magnetic fields such as Corrosion Related Magnetism (CRM) Daya et al. (2005); Lundin (2003). CRM is generated by currents in the hull, normally induced by corrosion or the propeller. It is therefore very difficult to estimate and subsequently difficult to cancel. This makes CRM important in naval target tracking but not so much in sensor localization. In our study, a friendly vessel used for sensor localization can turn off its active signature cancelling, resulting in a magnetic field from the dipole which is considerably stronger than the field induced by CRM. We have therefore neglected CRM.

An underwater sensor network used for real time surveillance must be silent. Neither sensor localization, surveillance or data transfer can be allowed to expose the sensor network. Silent communication rules out the use of acoustic modems which are the common mean of wireless underwater data transfer Akyildiz et al. (2005). We therefore assume that the sensors are connected by wire. As a consequence, common problems in underwater sensor networks such as time synchronization, limited bandwidth and limited energy resources will be neglected.

The sensor localization problem is basically reversed Simultaneous Localization and Mapping (SLAM). In common SLAM Durrant-Whyte and Bailey (June 2006); Bailey and Durrant-Whyte (Sept. 2006), landmarks in the environment are tracked with on-board sensors. The positions of these landmarks and the vehicle trajectory are estimated simultaneously in a filter. In sensor localization the sensors are observing the vessel from the "landmarks" position. The problem has the same form as SLAM but with a known number of landmarks and known data association.

The paper outline is as follows; Section 2 describes the system, measurement models and state estimation. Simulations of the performance of the positioning scheme, its sensitivity to different errors and the importance of the appearance of the trajectory are studied in Section 3. The paper ends with conclusions.

2 Methodology

This section describes the nonlinear state estimation problem here solved with EKF-SLAM, how the vessel dynamics and sensors are modeled and how different performance measures are computed. All vectors are expressed in a world fixed coordinate system unless otherwise stated.

2.1 System Description

The sensor positioning system is assumed to have the following process and measurement model

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k) + \boldsymbol{w}_k \tag{1}$$

$$\boldsymbol{y}_k = h(\boldsymbol{x}_k, \boldsymbol{u}_k, \boldsymbol{e}_k^u) + \boldsymbol{e}_k \tag{2}$$

where $f(\cdot)$ is a nonlinear state transition function, $h(\cdot)$ is a nonlinear measurement function, $\boldsymbol{x}_k$ the state vector, $\boldsymbol{u}_k$ the inputs, $\boldsymbol{w}_k$ the process noise, $\boldsymbol{e}_k^u$ input noise and $\boldsymbol{e}_k$ measurement noise. In SLAM the state vector consists of both the vessel position $\boldsymbol{p}_v = [x, y]^T$ and landmark (sensor) states $\boldsymbol{s}$ stacked, i.e. $\boldsymbol{x} = [\boldsymbol{p}_v^T, \boldsymbol{s}^T]^T$.

Process Model

The process model describes the vehicle and the sensors dynamics. There are complex vessel models available which include 3D orientation, angular rates, engine speed, rudder angle, waves, hull, etc., see e.g. Fossen and Perez (2004). Since we do not consider any particular vessel or weather condition, a very simple vessel model is used. It is assumed that no substantial movement in the z-coordinate, pitch and roll angles of the vessel are made, hence a nonlinear 5 states coordinated turn model is sufficient. The parametrization used is a linearized discretisation according to Gustafsson (2001)

$$x_{k+T} = x_k + \frac{2v_k}{\omega_k} \sin(\omega_k T) \cos(h_k + \frac{\omega_k T}{2}) \tag{3a}$$

$$y_{k+T} = y_k + \frac{2v_k}{\omega_k} \sin(\omega_k T) \sin(h_k + \frac{\omega_k T}{2}) \tag{3b}$$

$$v_{k+T} = v_k \tag{3c}$$

$$h_{k+T} = h_k + \omega_k T \tag{3d}$$

$$\omega_{k+T} = \omega_k \tag{3e}$$

where T is the sampling interval and (x, y), v, h, ω denote position, speed, heading and angular rate, respectively. Furthermore, it is assumed that the sensors are static and do not move after some time of deployment, hence a process model for the sensors is

$$s_{x_j,k+T} = s_{x_j,k} \tag{4a}$$

$$s_{y_j,k+T} = s_{y_j,k} \qquad j = 1, \ldots, M \tag{4b}$$

where M is the number of sensors, s_{x_j} and s_{y_j} are sensor j's x and y position, respectively.

Measurement Model

Each sensor contains a pressure sensor which is used as an input, $d_{j,k}$, of the z-component. The sensor also contains accelerometers which are used to determine the direction of the earth gravitational field. The magnetometers in the sensor can be used to compute the direction of the earth magnetic inclination if the environment is free from magnetic disturbances such as ships. In most cases the

magnetic inclination vector will not be parallel to the gravitational vector (except for the magnetic north and south pole) and the sensor orientation may be readily measured. The sensor orientation is modeled as a static input $\boldsymbol{C}_j$.

In this paper we only consider the ferromagnetic signature due to the iron in vessel construction. The ferromagnetic signature stems from the large pieces of metal used to construct a vessel. Each piece has its own magnetic dipole and the sum of these dipoles can roughly be simplified into a single dipole. The magnetic flux density for a dipole diminishes cubically with the distance to the dipole. With vector magnetometers dipole orientation can be estimated. Triaxial measurements of the magnetic flux density from a dipole can be modeled as

$$h(\boldsymbol{x}_k, \boldsymbol{u}_k) = \tag{5}$$

$$\frac{\mu_0}{4\pi|\boldsymbol{r}_{j,k}|^5}(3\langle \boldsymbol{r}_{j,k}, \boldsymbol{m}(h_k)\rangle \boldsymbol{r}_{j,k} - |\boldsymbol{r}_{j,k}|^2 \boldsymbol{m}(h_k)) \tag{6}$$

where $\boldsymbol{m}(h_k) = [m_x \cos(h_k), m_y \sin(h_k), m_z]^T$ is the magnetic dipole of the vessel and μ_0 is the permeability of the medium. $\boldsymbol{r}_{j,k} = \boldsymbol{C}_j[x_k - s_{x_{j,k}}, y_k - s_{y_{j,k}}, 0 - d_{j,k}]^T$ is the vector from each sensor to the vessel where $\boldsymbol{C}_j$ is the static orientation of sensor j in the global coordinate frame and $d_{j,k}$ is the measured depth of the sensor. Note that $d_{j,k}$ and $\boldsymbol{C}_{j,k}$ should be seen as inputs, $\boldsymbol{u}_k = \{d_{j,k}, \boldsymbol{C}_j\}_{j=1}^M$, since these are measured variables but not part of the state vector. The dipole model without coordinate transformations can be found in e.g. Cheng (1989). In the proximity of the vessel, a possibly better model would be a multiple dipole model Lindin (2007) where the measurement is the sum of several dipoles, but this is out of the scope of this paper. A single dipole is a reasonable approximation if the measurements are made at a large distance compared to vessel size Nelson and Richards (2007).

The magnetic dipole used throughout the simulations was $\boldsymbol{m} = [50, -5, 125]^T$ kAm^2 (same as in Birsan (2005)). Fig. 1 shows the measured magnetic flux density at sensor 3 in Fig. 2 from a vessel where the dipole has been slightly rotated around the z-axis between each simulation. The upper left figure in Fig. 1 was acquired using the magnetic dipole discussed earlier. Clearly the direction of the dipole affects the measured magnetic field. This indicates the importance of using a accurate dipole estimate.

2.2 State Estimation

Our approach to the state estimation problem is to use an Extended Kalman Filter (EKF) in the formulation of EKF-SLAM, for details see e.g. Durrant-Whyte and Bailey (June 2006). There are some characteristics in this system which do not usually exist in the common SLAM problem;

- The landmarks (sensor) are naturally associated to the measurements, i.e. data association is solved.
- The sensors global orientations are known which in turn makes it possible to estimate the orientation of the trajectory.

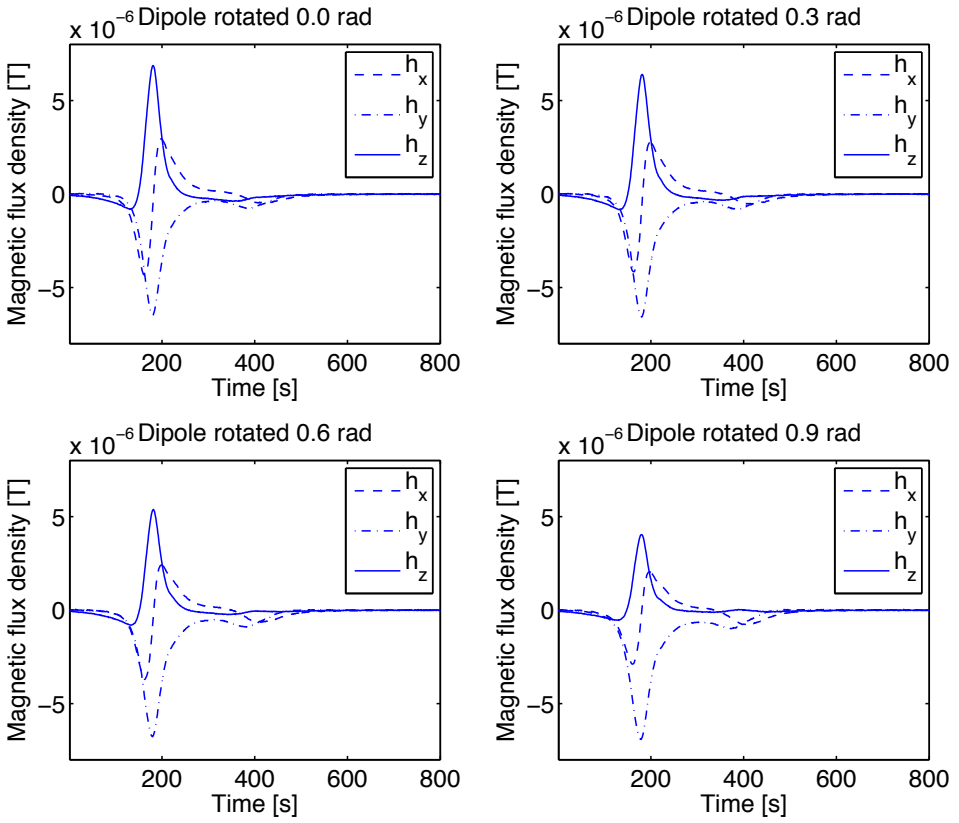


Figure 1: *Measured magnetic flux density at sensor 3 in Fig. 2 for vessels with slightly rotated dipoles.*

- The planar motion assumption and the pressure sensor makes it possible to transform sensor positioning into 2D.

A well known problem with SLAM is the ever expanding state space that comes with addition of new landmarks which will eventually make it intractable to compute a solution. In a sensor network the number of sensors (landmarks) will normally be known.

Due to the duality of the estimation problem implied in SLAM, i.e. *estimate a map and simultaneously localize the vehicle in the map*, the question of state observability needs to be answered. Previous observability analyzes on the SLAM problem Kim and Sukkarieh (2004); Bryson and Sukkarieh (2008); Andrade-Cetto and Sanfeliu (2005); Lee et al. (2006); Wang and Dissanayake (2008); Perera et al. (2009) has focused on vehicle fixed range and/or bearing sensors, such as laser and camera. Perera et al. (2009); Andrade-Cetto and Sanfeliu (2005) conclude that only one known landmark needs to be observed in 2D SLAM for the global map to be locally weakly observable. In our proposed system the sensors are in the actual landmarks position and their measurements are informative in both range and bearing to the dipole, hence the global map is observable if one sensor position is known. Theoretically this means that the sensor positions and the trajectory can be estimated in a global coordinate frame with a global map position error depending only on the error of the known sensor. If no global position of either sensor or vessel is available the sensors can be positioned locally.

Even if the system is observable there are no guarantees that a EKF will converge since it depends on the linearization error and the initial linearization point. More recent approaches to the SLAM problem Kaess et al. (2007); Dellaert (2005) consider smoothing instead of filtering. These methods can handle linearization errors better since the whole trajectory and map can be re-linearized. Yet, a good initial linearization point is necessary.

2.3 Cramér-Rao Lower Bound

Given the trajectory of a vessel, it is interesting to study a lower bound on the covariance of the estimated sensor positions. We have chosen to study the Cramér-Rao lower bound (CRLB) due to its simplistic advantages. CRLB is the inverse of the Fisher Information Matrix (FIM), $I(\boldsymbol{x})$, which in case of Gaussian measurement errors can be calculated as

$$I(\boldsymbol{x}) = H(\boldsymbol{x})^T R(\boldsymbol{x})^{-1} H(\boldsymbol{x}), \tag{7a}$$

$$H(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} h(\boldsymbol{x}) \tag{7b}$$

where $R(\boldsymbol{x})$ is Gaussian measurement noise and $H(\boldsymbol{x})$ denote the gradient of $h(\boldsymbol{x})$ w.r.t. $\boldsymbol{x}$. The CRLB of a sensor position is given by

$$Cov(\boldsymbol{s}) = E\left\{(\boldsymbol{s^0} - \boldsymbol{s})(\boldsymbol{s^0} - \boldsymbol{s})^T\right\} \tag{8a}$$

$$\geq I(\boldsymbol{s})^{-1} \tag{8b}$$

where $\boldsymbol{s}^0$ is true sensor position and $\boldsymbol{s}$ is the corresponding estimate. Since information is additive the FIM of a sensor location for a certain trajectory can be calculated as the sum of the FIMs from all vessel positions along the trajectory. The lower bound of the covariance of the sensor position estimate is then the inverse of the sum of the FIMs. A more extensive study of the fundamentals of CRLB can be found in Kay (1993).

3 Simulation Results

The sensor positioning problem can, depending on which sensors are available, be solved in different ways. If no accurate global position of the vessel or a sensor is available during the experiment (GPS is for example easily jammed), the sensors can only be positioned locally. In Section 3.1, magnetometers are used to localize the sensors. If global vessel position is available throughout the experiment, from GNSS or using a radar sensor and a sea chart, it can be used as a measurement of the position of the vessel. This will not only position the sensors globally but also enable a more accurate trajectory estimation. This experimental configuration is simulated in Section 3.2. The parameters used in the simulations are listed in Table 1.

Param.	Covariance SLAM/GNSS	**Param.**	Value
x_0	10/10 m	$\boldsymbol{m}$	[50, -5, 125] kAm^2
y_0	10/10 m	μ_0	4π 10^{-7} Tm/A
v_0	0/0 m/s	$d_{j,0}$	{-5, -15} m
h_0	1/1 rad	T	0.1 s
ω_0	0/0 rad/s		
s_{x_j}	400/400 m		
e_{GNSS}	1 m		
e_h	10^{-16} T		

Table 1: *The parameters used in the simulations.*

3.1 Magnetometers Only

If there is no reliable global position measurement of the vessel, the trajectory of the vessel must be estimated using the same magnetic fluctuatuions as are being used to localize the sensors. Simulations show that the sensor network needs to be more dense when no GNSS is available. If there is little or no overlap in which two or more sensors observe the vessel simultaneously, the trajectory estimate, and in the end the sensor position estimates, depend more on the vessel model than observations. Yet, the sensor positions are still coupled through the covariance matrix.

A sensor localization simulation using 7 sensors and a generated trajectory is shown in Fig. 2 and Fig. 3. Fig. 4 shows the Root Mean Square Error (RMSE) of

each sensor as it developes over time. Since the initial guesses of sensor positions were generated independently, different sensors have different initial errors. All sensor though have the same initial uncertainty covariance (400 m, see Table 1). The initial guesses are meant to represent the prior information of the true sensor positions, acquired during sensor deployment. The limited range of the magnetic fluctuations causes the sensor position estimate to change only when the vessel is sufficiently close. This can be studied in Fig. 4. Sensor 4 in Fig. 2 is too far away from the vessel for accurate positioning, resulting in a large uncertainty ellipse. From Fig. 2, it is clear that error in trajectory estimates result in errors in estimated sensor positions.

200 Monte Carlo simulations using different trajectories and sensor locations show that this configuration results in a positioning error of 26.3% in average. A sensor failing to retain the true sensor position within two standard deviations was considered incorrectly positioned. In Fig. 2, sensor 7 is incorrectly positioned.

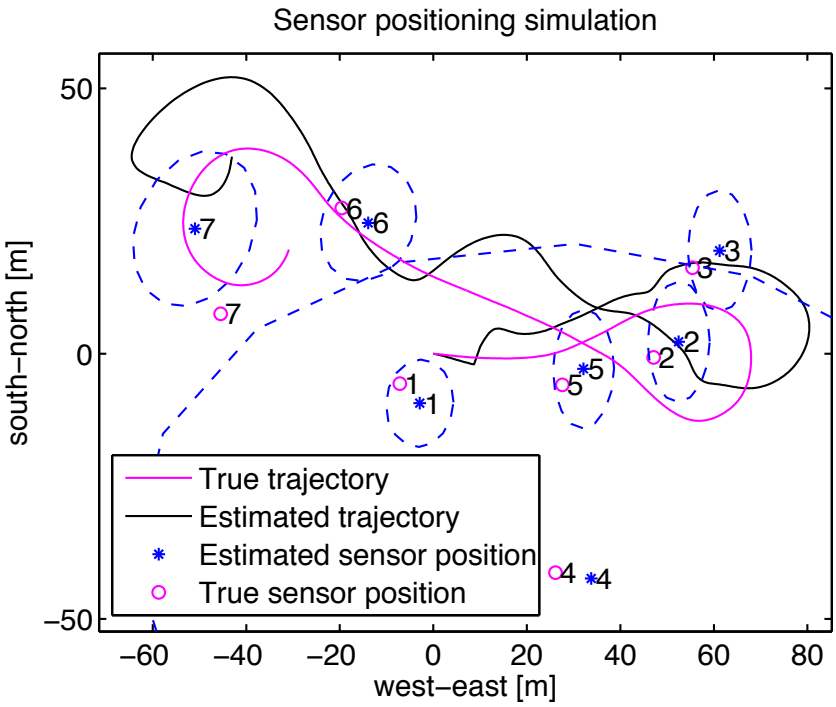


Figure 2: *Estimated sensor positions with 2σ uncertainty and vessel trajectory, for simulations using magnetometers.*

3.2 Magnetometers and GNSS

If global position measurements of the vessel are available throughout the trajectory, these measurements are used to improve the trajectory estimate. Each

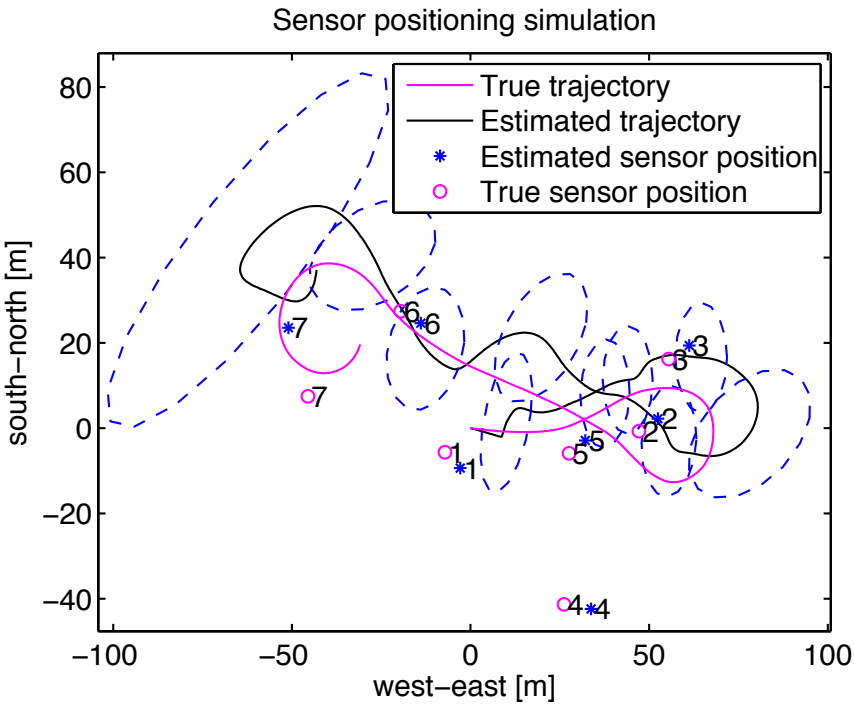


Figure 3: *Estimated vessel trajectory with 2σ uncertainty and sensor positions, for simulations using magnetometers.*

sensor is positioned relative to the trajectory of the vessel and is therefore less dependent of other sensor positions than in Section 3.1. This is quite natural since the cross correlations will not have such great impact on the sensor position estimates when the trajectory is known. Simulation results using the same sensor positions and trajectory used in Section 3.1, are shown in Fig. 5. Fig. 6 shows the RMSE of each sensor throughout the simulation. The global trajectory measurements result in more accurate sensor position estimates and lower uncertainties than using only magnetometers. Sensor 4 is far away from the trajectory resulting in a very uncertain position estimate.

200 Monte Carlo simulations using different trajectories and sensor locations show that using magnetometers and GNSS results in a sensor positioning error of 12.9% in average.

3.3 Trajectory Evaluation using CRLB

CRLB for sensor positions surrounding a couple of trajectories were calculated for the case of GNSS and magnetometers. A high CRLB indicates that after a simulation, a sensor in that position would still have a high uncertainty. Fig. 7 shows the trajectory used in Sections 3.1 and 3.2. Fig. 8 and Fig. 9 show two other

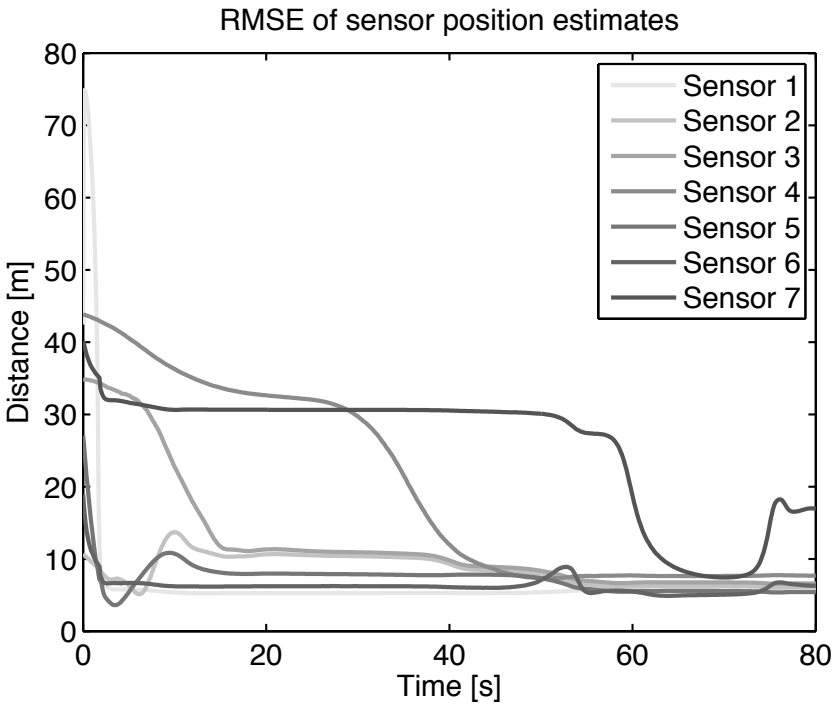


Figure 4: *Root Mean Square Error of estimated sensor positions throughout the simulation using only magnetometers.*

trajectories. It is clear that the CRLB becomes low in an area where the vessel can be observed from many directions. In Fig. 8 sensor positions quite close to the end of the trajectory have a high CRLB since they only observe the vessel from one direction. In Fig. 9 sensor positions between the start and end point of the trajectory are relatively difficult to estimate since it only observe the vessel from two opposite directions. The simulations suggest that in field experiments the vessel should be manouvered in a trajectory that allows each sensor to observe the vessel from as many directions as possible.

3.4 Sensitivity Analysis, Magnetic Dipole

The magnetic dipole of the vessel will probably not be accurately measured in a real world experiment. How will the positioning perform if the estimated magnitude of the dipole is for example 102% or 110% of the true magnitude?

The trajectory previously used has been simulated using an assumed dipole that differs from the true one. A dipole with a magnitude of 98% of the true one is generated and the error is divided over the three components of the dipole. Each dipole error is simulated multiple times using the same trajectory and each time the error is distributed amongst the dipole components differently. Again, a sen-

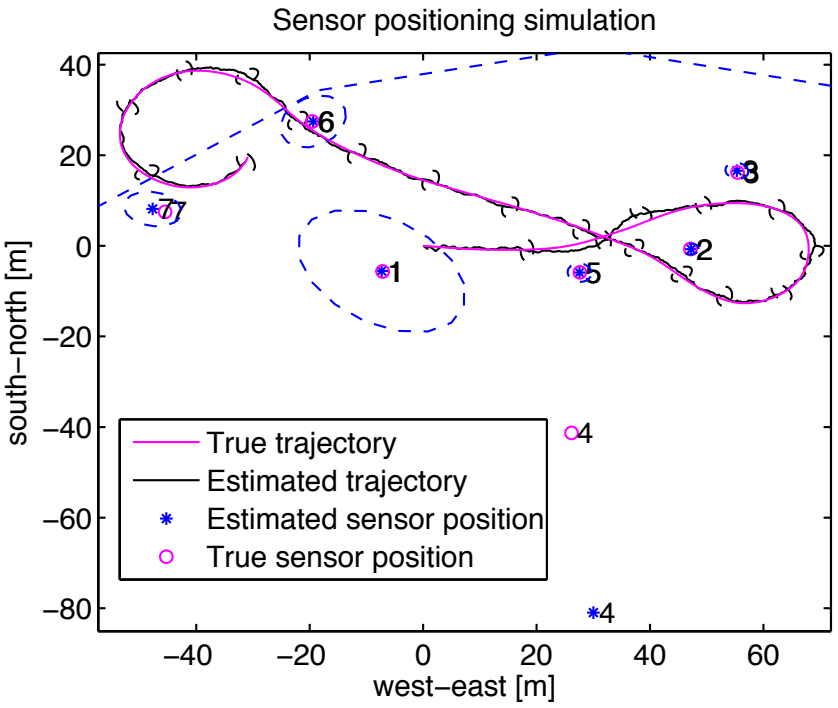


Figure 5: *Estimated vessel trajectory and sensor positions with 2σ uncertainty. GNSS and magnetometers are used as sensors.*

sor failing to retain the true sensor position within two standard deviations is considered incorrectly positioned. Table 2 shows the percentage of incorrectly positioned sensors for different errors of magnitude and different simulation settings.

Dipole	80%	90%	95%	98%	99%	100%
SLAM	44.6%	25.7%	23.4%	23.4%	18.9%	14.3%
GNSS	38.3%	9.7%	3.4%	2.9%	0.0%	0.0%

Dipole	101%	102%	105%	110%	120%
SLAM	14.3%	14.3%	16.6%	34.3%	53.1%
GNSS	4.0%	4.6%	8.6%	12.0%	38.3%

Table 2: *Sensitivity analysis of error in dipole estimate.*

3.5 Sensitivity Analysis, Sensor Orientation

The sensor orientation is assumed measured in the previous experiments since it can be estimated prior to the experiment. We will now study how sensitive the system is to errors in the orientation estimate. The positioning performance

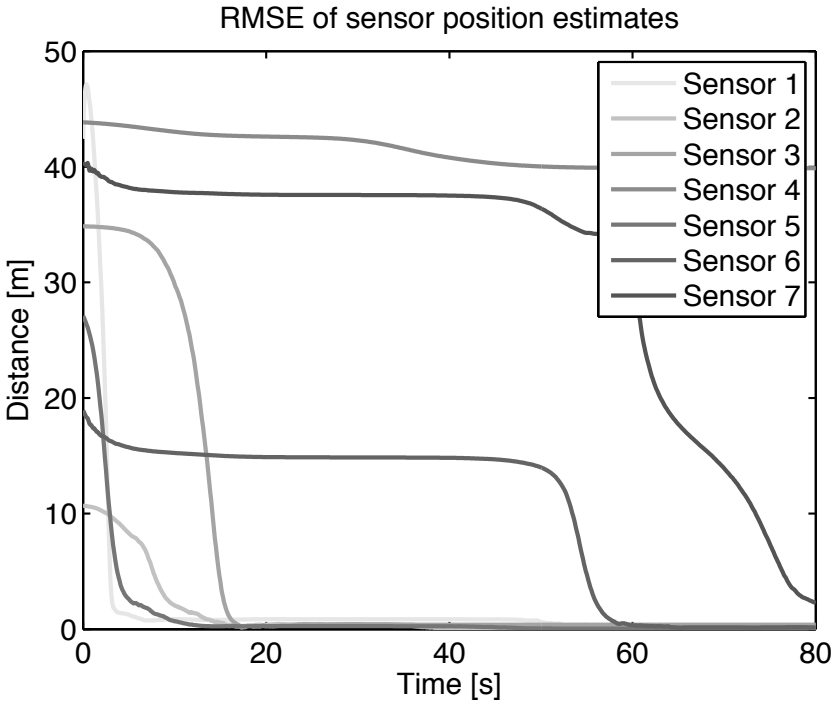


Figure 6: *RMSE of estimated sensor positions throughout the simulation. GNSS and magnetometers are used as sensors.*

when sensor orientation errors are present, are evaluated using 25 Monte Carlo simulations for each orientation error using different trajectories. For each simulation, random orientation errors with the stated covariance are generated. (A covariance of 0.16 rad results in orientation errors up to 0.8 rad or 45°.) Table 3 shows the percentage of incorrectly positioned sensors for different sensor orientation error covariances.

Note that the sensor positioning error of a system using GNSS and magnetometers was merely unaffected by the introduction of an orientation covariance of up to 0.04 rad. If the sensor observes the vessel from many different directions, the positioning still succeeds. When only magnetometers are used, the trajectory measurements cannot compensate for the errors in orientation, rendering larger positioning errors.

Ori Cov	0.0 rad	0.01 rad	0.04 rad	0.16 rad	0.36 rad
SLAM	26.3%	29.8%	36.9%	54.8%	52.4%
GNSS	12.9%	12.5%	11.9%	18.5%	26.8%

Table 3: *Sensitivity analysis of error in estimated sensor orientation.*

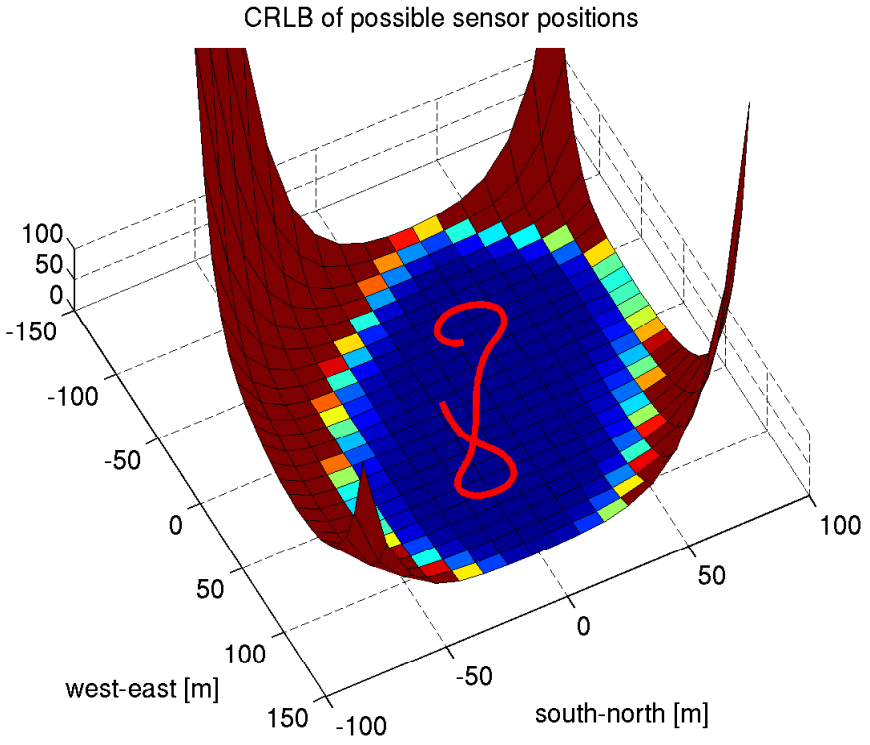


Figure 7: *CRLB for all sensor positions surrounding the trajectory (in red). Trajectory 1.*

4 Conclusions

We have presented a silent underwater sensor localization scheme using triaxial magnetometers and a friendly vessel with known magnetic characteristics. More accurate sensor positions will enhance the detection, tracking and classification ability of the underwater sensor network. Monte Carlo simulations indicate that a sensor positioning accuracy of 26.1% is achievable when only magnetometers are used and of 12.9% when GNSS and magnetometers are used. Knowing the magnetic dipole of the vessel is important and a dipole magnitude error of 10% results in a positioning error increase of about 10%. Simulations also show that our positioning scheme is quite unsensitive to minor errors in sensor orientation, when GNSS is used throughout the trajectory.

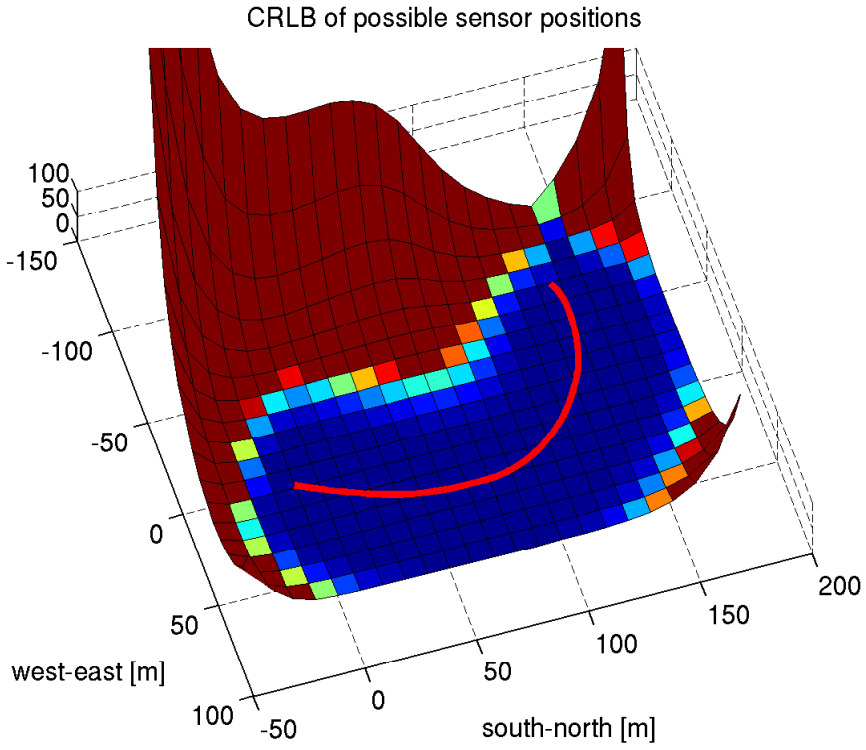


Figure 8: *CRLB for all sensor positions surrounding the trajectory (in red). Trajectory 2.*

Acknowledgment

This work was supported by the Strategic Research Center MOVIII, funded by the Swedish Foundation for Strategic Research, SSF, CADICS, a Linnaeus center funded by the Swedish Research Council, and LINK-SIC, an Industry Excellence Center founded by Vinnova.

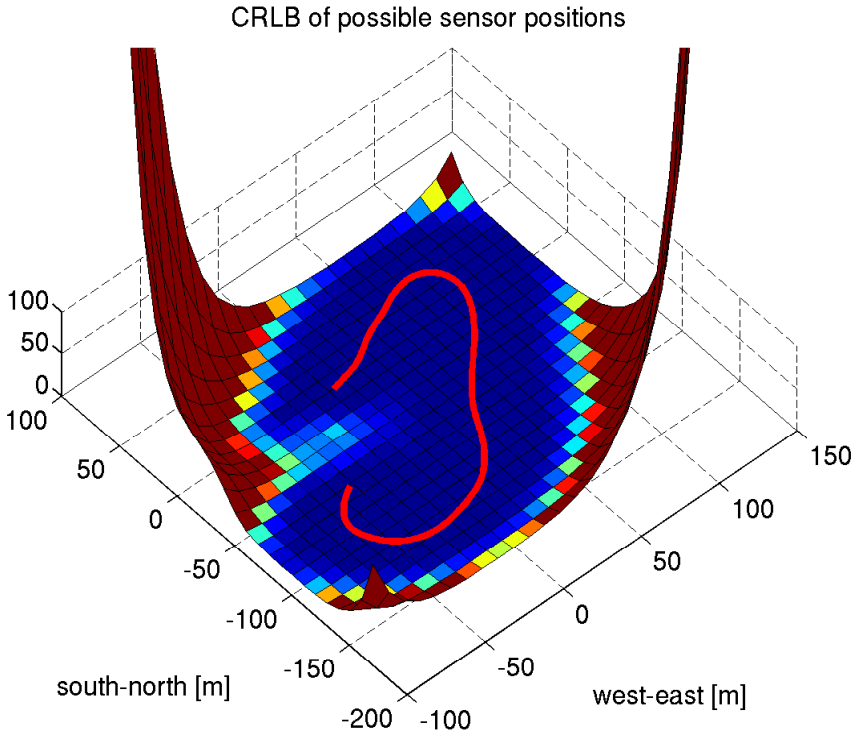


Figure 9: *CRLB for all sensor positions surrounding the trajectory (in red). Trajectory 3.*

Bibliography

I. F. Akyildiz, D. Pompili, and T. Melodia. Underwater acoustic sensor networks: research challenges. *Elsevier Journal of Ad Hoc Networks*, 3(3), 2005.

J. Andrade-Cetto and A. Sanfeliu. The effects of partial observability when building fully correlated maps. 21(4):771–777, Aug. 2005. ISSN 1552-3098. doi: 10.1109/TRO.2004.842342.

T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (SLAM): part II. *Robotics & Automation Magazine, IEEE*, 13(3):108 – 117, Sept. 2006.

M. Birsan. Non-linear kalman filters for tracking a magnetic dipole. In *Proc. of Intl. Conf. on Maritime Electromagnetics, MARELEC*, 2004.

M. Birsan. Unscented particle filter for tracking a magnetic dipole target. In *Proc. of MTS/IEE OCEANS*, 2005.

M. Birsan. Electromagnetic source localization in shallow waters using Bayesian matched-field inversion. *Inverse Problems*, 22(1):43–53, 2006.

M. Bryson and S. Sukkarieh. Observability analysis and active control for airborne SLAM. *Aerospace and Electronic Systems, IEEE Transactions on*, 44(1): 261–280, 2008. ISSN 0018-9251. doi: 10.1109/TAES.2008.4517003.

J. Callmer, M. Skoglund, and F. Gustafsson. Silent localization of underwater sensors using magnetometers. *EURASIP Journal on Advances in Signal Processing*, 2010, 2010. doi: 10.1155/2010/709318. URL `http://dx.doi.org/10.1155/2010/709318`. Article ID 709318.

V. Chandrasekhar and W. Seah. Area Localization Scheme for Underwater Sensor Networks. In *Proc. of the IEEE OCEANS Asia Pacific Conf.*, 2006.

V. Chandrasekhar, W. Seah, Y. Sang Choo, and H. Ee. Localization in underwater sensor networks: survey and challenges. In *ACM Intl. Workshop on UnderWater Networks WUWNet*, 2006.

D. Cheng. *Field and Wave Electromagnetics*. Addison-Wesley, Reading, Massachusetts, 2nd edition, 1989.

W. Cheng, A. Teymorian, L. Ma, X. Cheng, X. Lu, and Z. Lu. Underwater Localization in Sparse 3D Acoustic Sensor Networks. In *Proc. of conf. on Computer Communications, INFOCOM,IEEE*, 2008.

X. Cheng, H. Shu, Q. Liang, and D. Du. Silent Positioning in Underwater Acoustic Sensor Networks. *Transactions on Vehicular Technology, IEEE*, 57(3), May. 2008.

E. Dalberg, A. Lauberts, R. Lennartsson, M. Levonen, and L. Persson. Underwater target tracking by means of acoustic and electromagnetic data fusion. In *Proc. of Intl. Conf. on Information Fusion*, 2006.

Z. Daya, D. Hutt, and T. Richards. Maritime electromagnetism and DRDC Signature Management research. Technical report, Defence R&D Canada, 2005.

F. Dellaert. Square root sam. In *Robotics: Science and Systems*, pages 177–184, 2005.

H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping (SLAM): part I. *Robotics & Automation Magazine, IEEE*, 13(2):99 – 110, June 2006.

M. Erol, L. Vieira, and M. Gerla. AUV-Aided Localization for Underwater Sensor Networks. In *Proc. of Intl. Conf. on Wireless Algorithms, Systems and Applications, WASA*, 2007.

M. Erol, L. Vieira, A. Caruso, F. Paparella, M. Gerla, and S. Oktug. Multi Stage Underwater Sensor Localization using Mobile Beacons. In *Proc. of Intl. Conf. on Sensor Technologies and Applications, SENSORCOMM*, 2008.

T. I. Fossen and T. Perez. Marine Systems Simulator (MSS), 2004. www.marinecontrol.org.

F. Gustafsson. *Adaptive Filtering and Change Detection*. John Wiley & Sons, Hoboken, New Jersey, 2nd edition, 2001.

M. Kaess, A. Ranganathan, and F. Dellaert. iSAM: Fast Incremental Smoothing and Mapping with Efficient Data Association. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 1670–1677, April 2007. doi: 10.1109/ROBOT.2007.363563.

S. M. Kay. *Fundamentals of Statistical Signal Processing - Estimation Theory*. Prentice Hall, 1993.

J. Kim and S. Sukkarieh. Improving the real-time efficiency of inertial SLAM and understanding its observability. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 21–26, Sendai, Japan, 28-2 Sept./Oct. 2004. doi: 10.1109/IROS.2004.1389323.

K. W. Lee, W. Wijesoma, and I. Javier. On the observability and observability analysis of SLAM. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3569–3574, Beijing, China, 9-15 Oct. 2006. doi: 10.1109/IROS.2006.281646.

A. Lindin. Analysis and modelling of magnetic mine sweep for naval purposes. Master's thesis, Linköping University, The Department of Physics, Chemistry and Biology, 2007. [In swedish].

A. Lundin. Underwater electric signatures. Are they important for a future navy? Master's thesis, Swedish National Defence College, 2003. [In swedish].

J. Nelson and T. Richards. Magnetic source parameters of MR OFFSHORE measured during trial MONGOOSE 07. Technical report, Defence R&D - Atlantic, Dartmouth NS (CAN), 2007.

L. Perera, A. Melkumyan, and E. Nettleton. On the linear and nonlinear observability analysis of the SLAM problem. pages 1–6, Málaga, Spain, 14-17 Apr. 2009. doi: 10.1109/ICMECH.2009.4957168.

C. Tian, W. Liu, J. Jin, Y. Wang, and Y. Mo. Localization and synchronization for 3D underwater acoustic sensor network. In *Intl. Conf. on Ubiquitous Intelligence and Computing UIC*, 2007.

Z. Wang and G. Dissanayake. Observability analysis of SLAM using Fisher information matrix. In *Proceedings of the International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 1242–1247, Hanoi, Vietnam, 17-20 Dec. 2008. doi: 10.1109/ICARCV.2008.4795699.

S. Wong, J. Lim, S. Rao, and W. Seah. Multihop Localization with Density and Path Length Awareness in Non-Uniform Wireless Sensor Networks. In *Proc. of IEEE Vehicular Technology Conf.*, 2005.

Z. Zhou, J.-H. Cui, and S. Zhou. Localization for large-scale underwater sensor network. In *Proc. of IFIP Networking*, 2007.

Paper B

A Nonlinear Least-Squares Approach to the SLAM Problem

Authors: Zoran Sjanic , Martin A. Skoglund, Thomas B. Schön and Fredrik Gustafsson

Edited version of the paper:

Z. Sjanic, M. A. Skoglund, F. Gustafsson, and T. B. Schön. A nonlinear least squares approach to the SLAM problem. In *Proceedings of the IFAC World Congress*, volume 18, Milan, Italy, 28-2 Aug./Sept. 2011.

A Nonlinear Least-Squares Approach to the SLAM Problem

Zoran Sjanic* **, Martin A. Skoglund*, Thomas B. Schön* and Fredrik Gustafsson*

*Dept. of Electrical Engineering,
Linköping University,
SE–581 83 Linköping, Sweden
{zoran, ms, schon,
fredrik}@isy.liu.se

**Dept. of Flight Data and Navigation,
Saab Aeronautics
SE-581 88 Linköping, Sweden

Abstract

In this paper we present a solution to the simultaneous localisation and mapping (SLAM) problem using a camera and inertial sensors. Our approach is based on the maximum a posteriori (MAP) estimate of the complete SLAM problem. The resulting problem is posed in a nonlinear least-squares framework which we solve with the Gauss-Newton method. The proposed algorithm is evaluated on experimental data using a sensor platform mounted on an industrial robot. In this way, accurate ground truth is available, and the results are encouraging.

Keywords: Inertial measurement units, Cameras, Smoothing, Dynamic systems, State estimation

1 Introduction

In this paper we present an optimisation based solution to the simultaneous localisation and mapping (SLAM) problem formulated as nonlinear least-squares, and solved with the Gauss-Newton method. The method aims at providing high quality SLAM estimates which could e.g., be used as priors for computing detailed terrain maps.

SLAM is the problem of estimating a map of the surrounding environment from a moving platform, while simultaneously localising the platform. These estimation problems usually involve nonlinear dynamics and nonlinear measurements of a high dimensional state space. In Dellaert, F. and Kaess, M. (2006) a nonlinear least-squares approach to SLAM, called square root Smoothing and Mapping ($\sqrt{\text{SAM}}$) is presented. We extend this approach by considering a full 6 DOF platform, 3 DOF landmarks, inputs using inertial sensors and camera measurements. The resulting algorithm is evaluated on experimental data from a structured in-

door environment and compared with ground truth data.

For more than twenty years SLAM has been a popular field of research and is considered an important enabler for autonomous robotics. An excellent introduction to SLAM is given in the two part tutorial by Durrant-Whyte and Bailey (2006); Bailey and Durrant-Whyte (2006) and for a thorough overview of visual SLAM Chli (2009) is highly recommended. In the seminal work of Smith et al. (1990) the idea of a stochastic map was presented and was first used in P. Moutarlier and R. Chatila (1989), where the estimate is computed with an Extended Kalman Filter (EKF). There are by now quite a few examples of successful EKF SLAM implementations, see e.g., J.E. Guviant and E.M. Nebot (2001); J. J. Leonard and H. Jacob and S. Feder (2000). Another popular approach is the FastSLAM method (Montemerlo et al., 2002, 2003) which uses particle filters. These are known to handle nonlinearities very well. Both EKF SLAM and FastSLAM suffer from inconsistencies due to poor data association, linearisation errors (Bailey et al., 2006a) and particle depletion (Bailey et al., 2006b).

Some impressive work where the SLAM problem is solved solely with cameras can be found in Davison et al. (2007); Davison (2003); Eade (2008); Klein and Murray (2007). The camera only SLAM methods have many similarities with bundle adjustment techniques, (Hartley and Zisserman, 2004; Triggs et al., 2000), and the stochastic map estimation problem can be seen as performing structure from motion estimation (Fitzgibbon and Zisserman, 1998; Taylor et al., 1991). Without any other sensors measuring the platform dynamics, the image frame rate and the visual information contents in the environment are limiting factors for the ego motion estimation, and hence the map quality.

Recent years' increase in computational power has made smoothing an attractive option to filtering. One of the first SLAM related publications, where the trajectory is not filtered out to a single estimate is Eustice et al. (2006), where the whole time history is estimated with a so called *delayed state information filter*. Other, more optimisation like approaches are Dellaert, F. and Kaess, M. (2006); M. Kaess and A. Ranganathan and F. Dellaert (2008); Bibby and Reid (2007); Bryson et al. (2009), which all optimise over the whole trajectory and a feature based map.

2 Problem Formulation

We assume that the dynamic model and the measurements are on the following form

$$x_t = f(x_{t-1}, u_t) + \underbrace{B_w w_t}_{\tilde{w}_t}, \tag{1a}$$

$$l_t = l_{t-1}, \tag{1b}$$

$$y_{t_k} = h(x_{t_k}, l_{t_k}) + e_{t_k}, \tag{1c}$$

where x_t and l_t are vehicle and landmark states, respectively, and the inertial measurements can be modelled as inputs u_t. The meaning of y_{t_k} is a measurement relative to landmark l_{t_k} at time t_k, and this is because the measurements and the dynamic model deliver data in different rates. If we assume that all the measurements and the inputs for $t = \{0 : N\}$ and $k = \{1 : K\}$ ($K \ll N$) are available and the noise is independent and identically distributed (i.i.d.), then the joint probability density of (1) is

$$p(x_{0:N},\ l_N|y_{1:K},\ u_{1:N}) = \\ p(x_0)\prod_{t=1}^{N} p_{\tilde{w}_t}(x_t|x_{t-1}, u_t)\prod_{k=1}^{K} p_{e_{t_k}}(y_{t_k}|x_{t_k}, l_{t_k}). \tag{2}$$

Note that the map, l_N, is static and the estimate is given for the last time step only. Furthermore, the initial platform state x_0 is fixed to the origin without uncertainty. This is a standard SLAM approach and x_0 is therefore treated as a constant. The smoothed maximum a posteriori (MAP) estimate of $x_{0:N}$ and l_N is then

$$[x^*_{0:N},\ l^*_N] = \underset{x_{0:N},\ l_N}{\arg\max}\ p(x_{0:N},\ l_N|y_{t_1:t_K},\ u_{1:N}) = \\ \underset{x_{0:N},\ l_N}{\arg\min}\ -\log p(x_{0:N},\ l_N|y_{t_1:t_K},\ u_{1:N}). \tag{3}$$

If the noise terms $\tilde{w}_t$ and e_{t_k} are assumed to be Gaussian and white, i.e., $e_{t_k} \sim \mathcal{N}(0, R_{t_k})$ and $\tilde{w}_t \sim \mathcal{N}(0, \widetilde{Q}_t)$, (3) then becomes

$$[x^*_{0:N},\ l^*_N] = \underset{x_{0:N},\ l_N}{\arg\min} \sum_{t=1}^{N} \|x_t - f(x_{t-1}, u_t)\|^2_{\widetilde{Q}_t^{-1}} + \\ \sum_{k=1}^{K} \|y_{t_k} - h(x_{t_k}, l_{t_k})\|^2_{R_{t_k}^{-1}}, \tag{4}$$

which is a nonlinear least-squares formulation.

3 Models

Before we introduce the details of the dynamic model some coordinate frame definitions are necessary:

- Body coordinate frame (b), moving with the sensor and with origin fixed in the IMU's inertial centre.
- Camera coordinate frame (c), moving with the sensor and with origin fixed in the camera's optical centre.
- Earth coordinate frame (e), fixed in the world with its origin arbitrary positioned.

When the coordinate frame is omitted from the states it is assumed that they are expressed in the earth frame e.

3.1 Dynamics

The dynamic model used in this application has 10 states consisting of the position and velocity of the b frame expressed in the e frame, $p^e = [p_x \; p_y \; p_z]^T$ and $v^e = [v_x \; v_y \; v_z]^T$, respectively. The orientation is described using a unit quaternion $q^{be} = [q_0 \; q_1 \; q_2 \; q_3]^T$ defining the orientation of the b frame expressed in the e frame. The IMU measurements are treated as inputs, reducing the state dimension needed, and we denote the specific force $u_a^b = [a_x^b \; a_y^b \; a_z^b]^T$ and denote the angular rate $u_\omega^b = [\omega_x^b \; \omega_y^b \; \omega_z^b]^T$. The dynamics of the sensor in (1a) is then

$$
\begin{aligned}
\underbrace{\begin{bmatrix} p_t^e \\ v_t^e \\ q_t^{be} \end{bmatrix}}_{x_t} = \\
\underbrace{\begin{bmatrix} I_3 & TI_3 & 0 \\ 0 & I_3 & 0 \\ 0 & 0 & I_4 \end{bmatrix} \begin{bmatrix} p_{t-1}^e \\ v_{t-1}^e \\ q_{t-1}^{be} \end{bmatrix} + \begin{bmatrix} \frac{T^2}{2} I_3 & 0 \\ TI_3 & 0 \\ 0 & \frac{T}{2} I_4 \end{bmatrix} \begin{bmatrix} \mathcal{R}(q_{t-1}^{be})^T u_{a,t}^b + g^e \\ S(u_{\omega,t}^b) q_{t-1}^{be} \end{bmatrix}}_{f(x_{t-1}, u_t)} \\
+ \underbrace{\begin{bmatrix} \frac{T^2}{2} I_3 & 0 \\ TI_3 & 0 \\ 0 & \frac{T}{2} \widetilde{S}(q_{t-1}^{be}) \end{bmatrix}}_{B_w(x_{t-1})} \underbrace{\begin{bmatrix} w_{a,t}^b \\ w_{\omega,t}^e \end{bmatrix}}_{w_t},
\end{aligned} \tag{5}
$$

where

$$
w_{a,t}^b \sim \mathcal{N}(0, Q_a), \qquad Q_a = \sigma_a I_3, \tag{6a}
$$

$$
w_{w,t}^e \sim \mathcal{N}(0, Q_w), \qquad Q_w = \sigma_w I_3, \tag{6b}
$$

$$
S(u_{\omega,t}^b) = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix}, \; \widetilde{S}(q_t^{be}) = \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix}, \tag{6c}
$$

where $\mathcal{R}(q_t^{be}) \in SO(3)$ is the rotation matrix parametrised using the unit quaternion and $\mathcal{R}(q_t^{be})^T u_{a,t}^b + g^e$ is the specific force input expressed in the e frame, where $g^e = [0 \; 0 \; -9.81]^T$ compensates for the earth gravitational field.

3.2 Landmark State Parametrisation

Landmark states are encoded in the Inverse Depth Parametrisation (IDP) (Civera et al., 2008). The first three states, x^e, y^e and z^e, represent the 3D position of the

camera when the landmark was first observed. The last three states describe a vector to the landmark in spherical coordinates parametrised with azimuthal angle φ^e, elevation angle θ^e and inverse distance ρ^e, giving $l^e = [x\ y\ z\ \theta\ \varphi\ \rho]^T$. The angles φ^e, θ^e and the inverse distance ρ^e are expressed in the right handed earth coordinate frame e with z^e-axis pointing upwards. This means that a landmark l, with earth fixed coordinates $[x_l^e\ y_l^e\ z_l^e]^T$ is parametrised as

$$\begin{bmatrix} x_l^e \\ y_l^e \\ z_l^e \end{bmatrix} = \begin{bmatrix} x^e \\ y^e \\ z^e \end{bmatrix} + \frac{1}{\rho^e} m(\varphi^e, \theta^e), \tag{7a}$$

$$m(\varphi^e, \theta^e) = \begin{bmatrix} \cos\varphi^e \sin\theta^e \\ \sin\varphi^e \sin\theta^e \\ \cos\theta^e \end{bmatrix}. \tag{7b}$$

Since the camera is calibrated, as in Zhang (2000) using the toolbox (Bouguet, 2010), the landmark states can be introduced using normalised pixel coordinates $[u\ v]^T$ according to

$$p^e = \begin{bmatrix} x^e \\ y^e \\ z^e \end{bmatrix}, \tag{8a}$$

$$g^e = \begin{bmatrix} g_x^e \\ g_y^e \\ g_z^e \end{bmatrix} = \mathcal{R}(q_t^{be})^T \mathcal{R}(q^{bc}) \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \tag{8b}$$

$$\varphi^e = \arctan 2(g_y^e, g_x^e), \tag{8c}$$

$$\theta^e = \arctan 2\left(\| [g_x^e\ g_y^e]^T \|_2, g_z^e \right), \tag{8d}$$

$$\rho^e = \frac{1}{d_0^e}. \tag{8e}$$

Here, q^{bc} is the unit quaternion describing the fixed rotation from the camera frame to the body frame. Furthermore, p^e is the camera position when the landmark is observed and d_0 is the initial depth for the landmark. Finally, $\theta = \arctan 2(\cdot)$ is the four-quadrant arc-tangent, $\theta \in [-\pi, \pi]$. The complete landmark vector is of the dimension $6 \times n_{\text{landmarks}}$ and $n_{\text{landmarks}}$ will vary depending on when new landmarks are initiated.

3.3 Camera Measurements

The measurements are sub-pixel coordinates in the images given by the SIFT feature extractor (Lowe, 1999). The dimension of the measurement vector y_{t_k} is $2 \times n_{\text{af}}$, where n_{af} denotes the number of associated features. The measurements are expressed in normalised pixel coordinates. The camera measurement equa-

tion relating states and measurements has the form

$$y_{t_k} = \underbrace{h(x_{t_k}, l_{t_k})}_{y^c_{t_k}} + e_{t_k}, \tag{9}$$

where

$$e_{t_k} \sim \mathcal{N}(0, R_{t_k}),\ R_{t_k} = \sigma_{\text{features}} I_{2\times n_{\text{af}}}. \tag{10}$$

Using the IDP, (7) and (8), for a single landmark j and omitting time dependency, the measurement (9) is calculated as

$$l^c_j = \begin{bmatrix} l^c_{x,j} \\ l^c_{y,j} \\ l^c_{z,j} \end{bmatrix} =$$

$$\mathcal{R}(q^{bc})^T \mathcal{R}(q^{be}) \left(\rho^e_j \left(p^e - p^e_j - \mathcal{R}(q^{be})^T r^{bc} \right) + m(\varphi_j, \theta_j) \right), \tag{11a}$$

$$y^c_j = \frac{1}{l^c_{z,j}} \begin{bmatrix} l^c_{x,j} \\ l^c_{y,j} \end{bmatrix}, \tag{11b}$$

where p^e_j and ρ^e_j are defined in (8a) and (8e), respectively. The translation r^{bc} and orientation $\mathcal{R}(q^{bc})$ defines the constant relative pose between the camera and the IMU. The parameters in r^{bc} and $\mathcal{R}(q^{bc})$ were estimated in the previous work by Hol et al. (2010).

4 Solution

The proposed solution starts with an initialisation of the states using EKF SLAM and the initial states are then smoothed using nonlinear least squares.

4.1 Initialisation

The nonlinear least-squares algorithm needs an initial estimate $x^0_{0:N}$, l^0_N, which is obtained using EKF SLAM. The time update is performed with the model (5) in a standard EKF, for details, see e.g., Kailath et al. (2000). The landmark states (1b) are stationary and will therefore only be corrected in the measurement update.

The measurement update needs some further explanation. Each time an image is available (which in our experiments is 8 times slower than the specific force and the angular rate inputs) a measurement update is made. The measurement update needs an association between the features extracted from the current image and the landmarks present in the state vector. The associations computed during EKF SLAM are found in the following way; first, all landmarks are projected into the image according to (9) and the most probable landmarks are chosen as the nearest neighbours inside a predefined region. Second, the SIFT feature descriptors for the landmarks and features inside the region are matched. In this way a data association sequence is created for each image, relating the measurements

and the landmarks in the state vector. To enhance the feature tracking we discard unstable features (i.e., those that are only measured once or twice) and features are proclaimed usable only if they are found at least three times.

4.2 Nonlinear Least-Squares Smoothing

The nonlinear problem (4) is in our approach solved using the Gauss-Newton method, i.e., at each iteration we solve the linearised version of the problem.

In order to formulate the linearised least-squares smoothing problem for our specific setup we first need some definitions:

$$F_t \triangleq \left.\frac{\partial f(x,u)}{\partial x}\right|_{(x,u)=(x_{t-1}^0,u_t)}, \tag{12}$$

is the Jacobian of the motion model and

$$H_{t_k}^j \triangleq \left.\frac{\partial h(x,l)}{\partial x}\right|_{(x,l)=(x_{t_k}^0,l_j^0)}, \tag{13}$$

is the Jacobian of the measurement k at time t_k with respect to the vehicle states. The IDP gives a special structure to the equations since the measurements of the features are related to the pose where the features where initialised. Therefore, the landmark Jacobian is split into two parts. The first part is

$$J_{x_{t_k}}^j \triangleq \left.\frac{\partial h(x,l)}{\partial x}\right|_{(x,l)=(x_{t_k}^0,l_j^0)}, \tag{14}$$

which is the Jacobian of measurement k at time t_k, with respect to the position where landmark j was initialised. The second part is the Jacobian of measurement k at time t_k of the states ϕ_j, θ_j and ρ_j of landmark j

$$J_{l_k}^j \triangleq \left.\frac{\partial h(x,l)}{\partial l}\right|_{(x,l)=(x_{t_k}^0,l_j^0)}. \tag{15}$$

From the initialisation, Section 4.1, a trajectory $x_{0:N}^0$ and a landmark l_N^0 estimate is given and is therefor treated as a constant. The linearised process model at time t is then

$$x_t^0 + \delta x_t = F_t(x_{t-1}^0 + \delta x_{t-1}) + Bu_t + B_w(x_{t-1}^0)w_t. \tag{16}$$

The linearised measurement equations are given by

$$y_{t_k}^j = h(x_{t_k}^0, l_j^0) + H_{t_k}^j \delta x_{t_k} + J_{x_{t_k}}^j \delta x_{t_k} + J_{t_k}^j \delta l_j + e_{t_k}^j. \tag{17}$$

The linearised least-squares problem for the prediction and measurement errors

is then

$$[\delta x_t^*,\ \delta l_j^*] = \arg\min_{\delta x_t, \delta l_j} \sum_{t=1}^{N} \|F_t \delta x_{t-1} - I\delta x_t - a_t\|^2_{\widetilde{Q}_t^{-1}} + \sum_{k=1}^{K} \|H^j_{t_k}\delta x_{t_k} + J^j_{x_{t_k}}\delta x_{t_k} + J^j_{t_k}\delta l_j - c^j_{t_k}\|^2_{R^{-1}_{t_k}} \tag{18}$$

where $a_t = x_t^0 - F_t x_{t-1}^0 - Bu_t$ and $c^j_{t_k} = y^j_{t_k} - h(x^0_{t_k}, l^0_j)$. Here a_t and $c^j_{t_k}$ are the prediction errors of the linearised dynamics around x_t^0 and the innovations, respectively. The stacked version of the problem (32) can be solved iteratively according to

$$\eta^{i+1} = \arg\min_{\eta} \|A(\eta^i)\eta - b(\eta^i)\|_2^2, \quad \eta^0 = 0, \tag{19}$$

where we define $\eta = [\delta x_t,\ \delta l_j]$, and $A(\eta)$ and $b(\eta)$ is the matrix part and the vector part of (32), respectively.

The structure of the A matrix is perhaps best explained using an example:

$$A(\eta) = \begin{bmatrix} A_{11} & 0 \\ A_{21} & A_{22} \end{bmatrix} =$$

$$\left[\begin{array}{ccccccccc|cc}
-I & & & & & & & & & & \\
F_2 & -I & & & & & & & & & \\
 & F_3 & \ddots & & & & & & & & \\
 & & \ddots & -I & & & & & & & \\
 & & & F_6 & \ddots & & & & & & \\
 & & & & \ddots & -I & & & & & \\
 & & & & & F_{10} & \ddots & & & & \\
 & & & & & & \ddots & -I & & & \\
\hline
J^1_{x_5} & & & H^1_5 & & & & & & J^1_5 & \\
J^2_{x_5} & & & H^2_5 & & & & & & & J^2_5 \\
J^2_{x_9} & & & & & H^2_9 & & & & & J^2_9 \\
J^1_{x_{13}} & & & & & & & H^1_{13} & & J^1_{13} &
\end{array}\right], \tag{20}$$

$t_k = 1$: Two landmarks are seen for the first time giving the landmarks' initialisation positions, i.e., the columns where the Jacobians (34) are placed.

$t_k = 5$: The second camera measurement arrives, landmarks 1 and 2 are observed and the first two block rows of A_{21} and A_{22} are added.

$t_k = 9$: Camera measurement 3 arrives, landmark 2 is observed and the third block row of A_{21} and A_{22} is added.

$t_k = 13$: Camera measurement 4 arrives, landmark 1 is observed and the fourth

Algorithm 1 Nonlinear Least-Squares Smoothing for SLAM

Input: x^0, l^0 (trajectory and map from previous iteration), u (inputs), data association
Output: x^s, l^s (smoothed estimate of the trajectory and the map)
N = # IMU measurements
$A = [\ ],\ a = [\ ], c = [\ ]$
for $i = 1$ to N **do**
  predict states, $x_i = f(x_{i-1}^0, u_i)$
  if image available **then**
    use the data association and calculate $h(x_i^0, l_i^0)$
    calculate $A_{11} = [A_{11}\ A_{11}^i]^T$,
    $A_{21} = [A_{21}\ A_{21}^i]$ and
    $A_{22} = [A_{22}\ A_{22}^i]$ according to (26) - (17)
    calculate $a_i = x_i^0 - x_i$
    calculate $c_i = y_i - h(x_i^0, l_i^0)$
    set $a = [a^T\ a_i^T]^T$
    set $c = [c^T\ c_i^T]^T$
  else
    calculate $A_{11} = [A_{11}\ A_{11}^i]^T$
    calculate $a_i = x_i^0 - x_i$
    set $a = [a^T\ a_i^T]^T$
  end if
end for
Assemble up A according to (17) and $b = [a^T\ c^T]^T$
solve the least squares problem (19)
calculate $[x^{sT},\ l^{sT}]^T = [x^{0T},\ l^{0T}]^T + \eta$

block row of A_{21} and A_{22} is added.

A single iteration of the nonlinear least-squares smoothing algorithm can be summarised in pseudo code as seen in Algorithm 1.

The least-squares problem is weighted, so it is assumed that all of the terms in (32) are multiplied with the corresponding matrix square root of the inverse of the covariance matrices for the process and the measurement noise, respectively. Note that the covariance matrix of the process noise, $\widetilde{Q}_t = B_w(x_t^0)Q_t B_w(x_t^0)^T$, is singular rendering the use of normal inversion impossible. In order to overcome this, we simply regularise the problem by adding a diagonal matrix ΔI to the covariance matrix, with Δ being a small number, rendering the covariance matrix invertible. Furthermore, it is assumed that the associations from the initialisation is good enough and that we do not have to compute new associations after each iterate.

5 Experiments

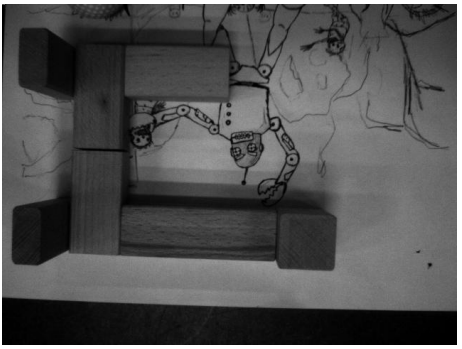
The implementation is done in MATLAB, except for the SIFT binaries, where we use a C code library from Hess.

5.1 Experimental Setup

For the purpose of obtaining high quality ground truth motion data we used an IRB 1400 industrial robot from ABB. In an industrial robot the rotation and translation of the end tool can be logged with high accuracy. This gives an excellent performance evaluation possibility, which is otherwise difficult. The actual robot trajectory was not possible to acquire during the experiment. However, since the industrial robot is very accurate the actual output of the robot will be very close to the programmed trajectory.

We constructed a small synthetic environment with known topography to obtain realistic ground truth map data, see Fig. 1b. We use a combined IMU/camera sensor unit, shown in Fig. 1a. The sensor unit is mounted at the end tool position of the industrial robot. The IMU measurements are sampled at 100 Hz and images of size 640×480 pixels are sampled at 12.5 Hz.

(a) *The combined strap-down IMU and camera system.*



(b) *An image from the camera during the experiment.*

Figure 1: *The IMU/camera sensor unit used in the experiments and an image from the camera over-viewing the synthetic environment.*

5.2 Results

The resulting trajectories and map obtained with the data from an experiment are presented in Fig. 2. The Ground truth trajectory is a reference trajectory for the robot. From these plots it is clearly visible that the smoothed estimate is closer to the true trajectory than the initial estimate. The improvement is also visible if the initial estimate and the final smoothed landmark estimate are compared as in Fig. 3. Note that some landmark positions are already quite accurately estimated

since the change is small after the smoothing. The smoothed estimate also has a more accurate universal scale.

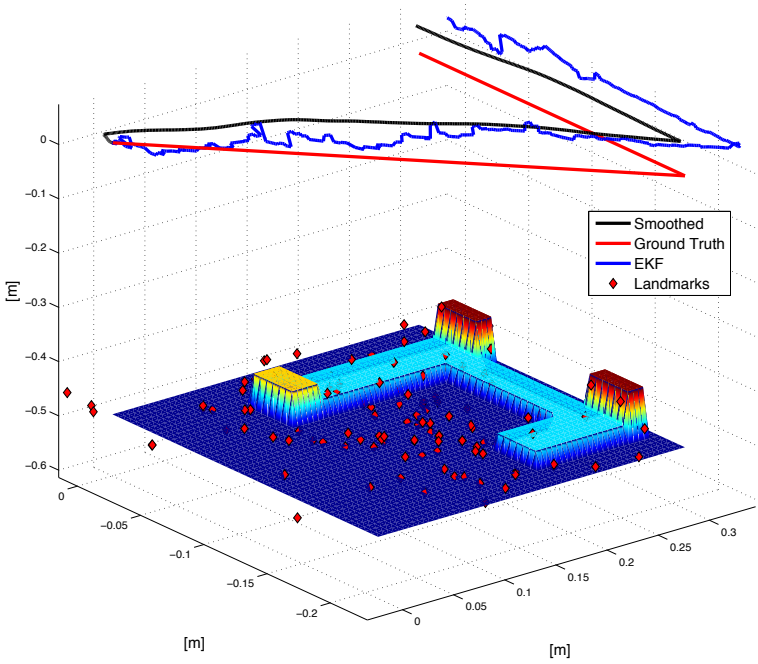


Figure 2: *The smoothed trajectory in red, the initial EKF trajectory in blue and the ground truth trajectory in black. The black crosses are the smoothed landmark estimates.*

Both the smoothed horizontal speed of the platform, defined as $\|[v_t^x \; v_t^y]^T\|_2$, and resulting estimate from the initialisation are plotted in Fig. 4. We see that the smoothed speed is much closer to 0.1 m/s, which is the true speed.

6 Conclusions and Future Work

In this work we have presented the SLAM problem formulated as nonlinear least-squares. For evaluation we have used a combined camera and IMU sensor unit mounted at the manipulator of an industrial robot which gives accurate ground truth.

The experimental results in Section 8 show that the nonlinear least-squares trajectory, Fig. 2, and the speed estimate, Fig. 4, show a significant improvement of the initial estimate. The sparse point cloud in Fig. 3, illustrating the initial landmark

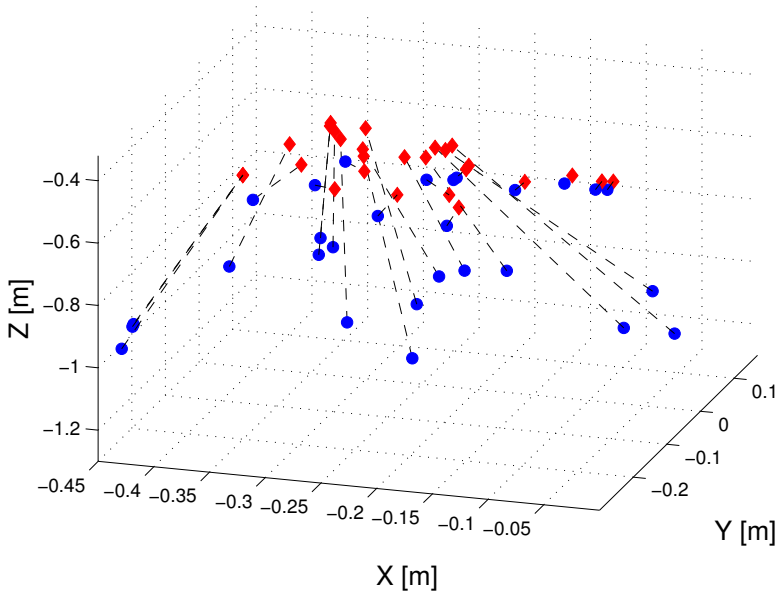


Figure 3: *The initial landmark estimates given by the EKF in blue bullets and the final smoothed estimate in red diamonds, where the black dashed line illustrate the relative displacement.*

estimate and final smoothed estimate, shows also an improvement. The universal scale of the environment is improved since the landmarks have moved towards more probable positions.

For a long-term solution another initialisation procedure is necessary, since EKF SLAM is intractable for large maps. A possible alternative is to use IMU supported visual odometry to get a crude initial estimate. This approach needs a supporting global data association scheme.

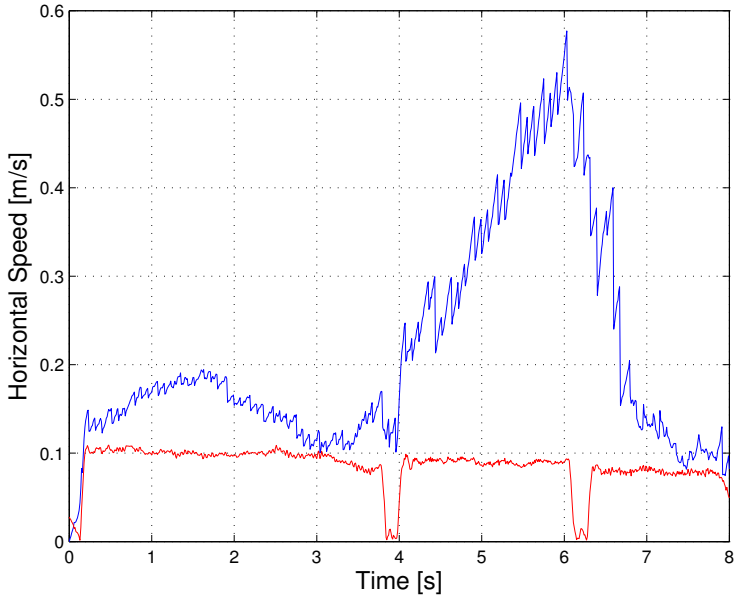


Figure 4: *The smoothed horizontal speed of the camera in red and EKF in blue. The true speed is* 0.1 *m/s except for when the robot stops and changes direction, this happens at about 4 seconds and 6 seconds.*

Bibliography

T. Bailey and H. Durrant-Whyte. Simultaneous Localization and Mapping (SLAM): Part II. *IEEE Robotics & Automation Magazine*, 13(3):108–117, Sept. 2006.

T. Bailey, J. Nieto, J. E. Guivant, M. Stevens, and E. M. Nebot. Consistency of the EKF-SLAM algorithm. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, pages 3562–3568, Beijing, China, 2006a. doi: 10.1109/IROS.2006.281644. URL `http://dx.doi.org/10.1109/IROS.2006.281644`.

T. Bailey, J. Nieto, and E. M. Nebot. Consistency of the FastSLAM algorithm. In *International Conference on Robotics and Automation (ICRA)*, pages 424–429, Orlando, Florida, USA, 2006b.

C. Bibby and I. Reid. Simultaneous Localisation and Mapping in Dynamic Environments (SLAMIDE) with reversible data association. In *Proceedings of Robotics: Science and Systems (RSS)*, Atlanta, GA, USA, June 2007.

J.-Y. Bouguet. Camera Calibration Toolbox for Matlab. `www.vision.caltech.edu/bouguetj/calib_doc/`, 2010.

M. Bryson, M. Johnson-Roberson, and S. Sukkarieh. Airborne smoothing and mapping using vision and inertial sensors. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 3143–3148, Kobe, Japan, 2009. IEEE Press. ISBN 978-1-4244-2788-8.

M. Chli. *Applying Information Theory to Efficient SLAM*. PhD thesis, Imperial College London, 2009.

J. Civera, A. Davison, and J. Montiel. Inverse Depth Parametrization for Monocular SLAM. *IEEE Transactions on Robotics*, 24(5):932–945, October 2008. ISSN 1552-3098. doi: 10.1109/TRO.2008.2003276.

A. J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Proceedings of the 9th IEEE International Conference on computer vision*, pages 1403–1410, Nice, France, 13-16 Oct. 2003.

A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067, 2007.

Dellaert, F. and Kaess, M. Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing. *International Journal of Robotics Research*, 25(12):1181–1203, 2006. ISSN 0278-3649. doi: 10.1177/0278364906072768. URL `http://dx.doi.org/10.1177/0278364906072768`.

H. Durrant-Whyte and T. Bailey. Simultaneous Localization and Mapping: Part I. 13(12):99–110, June 2006.

E. Eade. *Monocular Simultaneous Localisation and Mapping*. PhD thesis, Cambridge University, 2008.

R. M. Eustice, H. Singh, and J. J. Leonard. Exactly sparse delayed-state filters for view-based SLAM. *IEEE Transactions on Robotics*, 22(6):1100–1114, 2006.

A. W. Fitzgibbon and A. Zisserman. Automatic Camera Recovery for Closed or Open Image Sequences. In H. Burkhardt and B. Neumann, editors, *ECCV (1)*, volume 1406 of *Lecture Notes in Computer Science*, pages 311–326. Springer, 1998. ISBN 3-540-64569-1.

R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004. ISBN 0-521-54051-8.

R. Hess. SIFT library. URL `http://robwhess.github.com/opensift/`. http://robwhess.github.com/opensift/.

J. Hol, T. B. Schön, and F. Gustafsson. Modeling and Calibration of Inertial and Vision Sensors. *The International Journal of Robotics Research*, 29(2), Feb. 2010.

J. J. Leonard and H. Jacob and S. Feder. A Computationally Efficient Method for Large-Scale Concurrent Mapping and Localization. In *Proceedings of the Ninth International Symposium on Robotics Research*, pages 169–176, Salt Lake City, Utah, 2000. Springer-Verlag.

J.E. Guviant and E.M. Nebot. Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Transactions on Robotics and Automation*, 17(3):242–257, June 2001. ISSN 1042-296X.

T. Kailath, A. H. Sayed, and B. Hassibi. *Linear Estimation*. Prentice-Hall, Upper Saddle River, New Jersey, 2000.

G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 225–234, Nara, Japan, 2007.

D. Lowe. Object Recognition from Local Scale-Invariant Features. In *Proceedings of the Seventh International Conference on Computer Vision (ICCV'99)*, pages 1150–1157, Corfu, Greece, 1999.

M. Kaess and A. Ranganathan and F. Dellaert. iSAM: Incremental smoothing and mapping. 24(6):1365–1378, Dec. 2008.

M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, page 593–598, Edmonton, Alberta, Canada, 28-1 July/Aug. 2002.

M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proceedings of the Sixteenth International Joint*

Conference on Artificial Intelligence (IJCAI), Acapulco, Mexico, 9-15 Aug. 2003.

P. Moutarlier and R. Chatila. Stochastic multisensory data fusion for mobile robot location and environment modelling. In *5th International Symposium on Robotics Research*, pages 207–216, Tokyo, Japan, 1989.

Z. Sjanic, M. A. Skoglund, F. Gustafsson, and T. B. Schön. A nonlinear least squares approach to the SLAM problem. In *Proceedings of the IFAC World Congress*, volume 18, Milan, Italy, 28-2 Aug./Sept. 2011.

R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In *Autonomous robot vehicles*, pages 167–193. Springer-Verlag New York, Inc., New York, NY, USA, 1990. ISBN 0-387-97240-4.

C. Taylor, D. Kriegman, and P. Anandan. Structure and Motion in Two Dimensions from Multiple Images: A Least Squares Approach. In *Proceedings of the IEEE Workshop on Visual Motion*, pages 242 –248, Princeton, NJ , USA, October 1991.

B. Triggs, P. Mclauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment - a modern synthesis. In B. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, volume 1883 of *Lecture Notes in Computer Science*, pages 298–372. Springer-Verlag, 2000.

Z. Zhang. A flexible new technique for camera calibration. 22(11):1330–1334, Nov. 2000. ISSN 0162-8828. doi: 10.1109/34.888718.

Paper C

Modeling and Sensor Fusion of a Remotely Operated Underwater Vehicle

Authors: Martin A. Skoglund, Kenny Jönsson and Fredrik Gustafsson

Edited version of the paper:

M. A. Skoglund, K. Jönsson, and F. Gustafsson. Modeling and sensor fusion of a remotely operated underwater vehicle. In *Proceedings of the 15th International Conference on Information Fusion (FUSION), Singapore, 9-12 July 2012*, pages 947–954. IEEE, 2012.

Modeling and Sensor Fusion of a Remotely Operated Underwater Vehicle

Martin A. Skoglund*, Kenny Jönsson** and Fredrik Gustafsson*

*Dept. of Electrical Engineering,
Linköping University,
SE–581 83 Linköping, Sweden
{ms, fredrik}@isy.liu.se

**Saab Group
SE-581 88 Linköping, Sweden
kenny.x.jonsson@saabgroup.com

Abstract

We compare dead-reckoning of underwater vehicles based on inertial sensors and kinematic models on one hand, and control inputs and hydrodynamic model on the other hand. Both can be used in an inertial navigation system to provide relative motion and absolute orientation of the vehicle. The combination of them is particularly useful for robust navigation in the case of missing data from the crucial doppler log speedometer. As a concrete result, we demonstrate that the performance critical doppler log can be replaced with longitudinal dynamics in the case of missing data, based on field test data of a remotely operated vehicle.

1 Introduction

Unmanned Underwater Vehicles (UUV) are widely used around the world, both in military applications such as mine hunting and mine disposal, and in civilian applications such as surveillance of divers and tunnel or hull inspections. There are different kinds of UUV which are mainly divided into Remotely Operated Vehicles (ROV) and Autonomous Underwater Vehicles (AUV). The main difference between these two is that the ROV is tethered either to a submarine, a surface vessel or a used in a harbour and is thereby controlled by an operator. AUVs often just move forward and steer the heading and depth with rudders like a torpedo, contrary, ROVs can be steered in many directions since they usually have several thrusters. The ROV used in the experiments is the Saab Seaeye Falcon which is depicted in Fig. 1.

Inertial navigation is the key component in most advanced navigation systems, and the only approach that does not rely on external infrastructure of information. The key idea is that angular speeds measured by gyroscopes can be integrated to provide the orientation of the vehicle, while accelerations from accelerometers are after rotation with the current orientation integrated twice to provide the position relative the starting point. Inertial navigation suffers from

drift, more severe to the consumer grade than tactical grade sensors. A magnetometer and the indirect information from the gravity can be used to stabilize the orientation. Nevertheless, gravity leakage into the acceleration will occur, that causes a drift in position that is cubic in time. Integration of position information from global navigation satellite systems (GNSS) eliminates this drift, but satellite signals are not available in underwater environments. An additional velocity sensor, as the doppler log studied here, reduces this drift considerably to linear in time. Thus, velocity information is crucial for low-cost navigation systems. The doppler log is subject to outliers and missing data during certain operating conditions, so a backup system for speed information is highly desirable.

We here investigate whether a dynamic model of the vehicle can be used for navigation, together with the control signals (engine reference speed and rudder for general AUV's, input voltage to five thrusters for our ROV). The model potentially provides an independent observation of vehicle speed and angular rate. We will demonstrate that in particular the speed can be accurately predicted by the model. That is, fusion of kinematic and hydrodynamic models allows for robust navigation. Further, the dynamic model gives the speed in water, while the doppler log gives the speed over seabed. The difference corresponds to water stream, which is an important parameter for the control system, that can be estimated by fusion of the two models. Our experimental conditions did not have a significant stream and this potential benefit is not further investigated here.

Having two separate sources of speed and angular rate can also be used for fault detection and monitoring. For instance, a change in hydrodynamics caused by a foreign object stuck to the vehicle, or malfunctioning thrusters, can be detected and isolated. This might be a subject for future studies.

This paper is organised as follows. Section 2 discuss related literature. Section 3 gives a brief system overview and introduce the kinematics. The hydrodynamic models are presented in Section 4 while the sensor models are described in Section 5. The sensor fusion in combination with the hydrodynamic model is described in Section 6 and the results are presented in Section 8 on data from actual sea trials conducted in Lake Vättern, Sweden. Finally, Section 8 contains concluding remarks and gives some suggestions for future directions.

2 Related Work

Navigation of underwater vehicles are in general based on the onboard inertial navigation system (INS) but due its inherent drift, these systems are often aided. Support systems can for instance include doppler speedometers Jalving et al. (2004), GNSS fixes at the surface and acoustic baseline positioning using transponders Allen et al. (2006); Mandt et al. (2001). The INS can also be supplemented with map-aided bathymetric navigation Karlsson and Gustafsson (2006); Pappalardi et al. (2001); Ånonsen and Hallingstad (2006) where the bottom profile is measured using echo sounders. Other, interesting, approaches are using stereo

Figure 1: *The Saab Seaeye Falcon vehicle that was used that was used in the experiments.*

camera systems which estimates terrain information and navigation states on the fly Eustice et al. (2006); Mahon et al. (2008). Another way of improving navigation is considering physically motivated dynamic models, which in naval terminology is known as hydrodynamics. Hydrodynamic models can be seen as aiding sensors Fauske et al. (2007) which are independent of the INS, making them particularly useful for robust navigation and model based fault-detection. Estimation of hydrodynamic parameters has a long tradition, see Fossen (2002); Indiveri (1998), usually involving lengthy and costly experiments. Due to the complexity of the often largely over-parametrized models, specialized experiments have to be conducted isolating only a single or a few degrees of freedom. Motions like these will however rarely occur under normal operation conditions. Recently, methods for identification has emerged which do not rely on expensive reference systems as in Taino et al. (2007); Kim et al. (2002). ROVs are often configured differently, depending on the mission, and hence online estimation techniques are of great interest. Identification of underwater vehicles using an observer based method is done in Taino et al. (2007). Caccia et al. Caccia et al. (2000) is a further development of the work in Alessandri et al. (1998) where they identified a ROV from the inexpensive onboard sensor using an EKF with augmented state which inspired the work of Millert et al. Millert and Tayamon (2009).

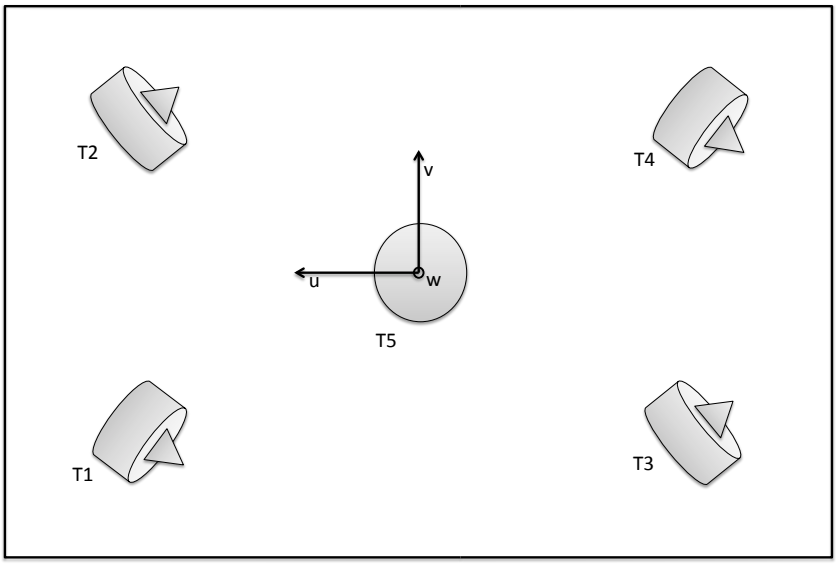


Figure 2: *The configuration of the thrusters as seen from above where the fore is pointing to the left.*

3 System Overview

The Falcon is an Open-Frame ROV similar to the JHUROV in Smallwood and Whitcomb (2003) and the ROMEO in Caccia et al. (2000). A variable configuration makes the ROV highly attractive as it can can be equipped differently depending on the mission. However, this may also alter previously estimated hydrodynamic parameters. The ROV has a cascade control structure where the outer loop consists of the operator which demands a reference speed and a reference heading and the inner loop consists of thruster PID-controllers striving to minimize the speed and heading errors by means of the INS. The dimensions are $0.7 \times 0.6 \times 1.0$m and is symmetrical along its axes. The dry mass of the Falcon is approximately 73kg with the standard sensor payload. The propulsion system consists of five brushless thrusters of which four are horizontal vectored thrusters for motions in surge, sway and yaw and a thruster for motion in heave. Thus, the Falcon is under-actuated as it is only controllable in four degrees of freedom. The thruster configuration can be seen in Fig. 2. The Falcon is equipped with five sensors, these are vector magnetometer, vector gyroscope, vector specific force, doppler velocity log (measuring linear speed) and a depth sensor.

3.1 Kinematics

The kinematics is expressed in the commonly used SNAME notation SNAME (1950). The ROV is a rigid body and its position and orientation can be described using seven coordinates

$$\boldsymbol{\eta} = \begin{bmatrix} x & y & z & \boldsymbol{q}^T \end{bmatrix}^T, \tag{1}$$

where x, y and z denote the position decomposed a local North-East-Down (NED) frame called the e-frame which is a plane tangential to the earth's surface at the current location. This frame is considered to be inertial. Furthermore, $\boldsymbol{q}$ is a unit quaternion

$$\boldsymbol{q} = \begin{bmatrix} q_0 & q_1 & q_2 & q_3 \end{bmatrix}^T. \tag{2}$$

parametrising the orientation of the vehicle w.r.t. the inertial frame, see e.g., Fossen (2002); Kuipers (2002) for details. The velocities

$$\boldsymbol{\nu} = \begin{bmatrix} u & v & w & p & q & r \end{bmatrix}^T, \tag{3}$$

are decomposed in the body fixed frame b. Here, u, v and w denote the velocity in surge, sway and heave and p, q and r denote the angular velocities in roll, pitch and yaw, respectively. The b frame is fixed in the vehicle with the u-axis pointing forewords in the vehicle's direction, v-axis to the port and the w-axis upwards. The origin of the frame will be the approximate centre of mass. The kinematic equations are then

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}(\boldsymbol{\eta})\boldsymbol{\nu}, \tag{4}$$

where $\boldsymbol{J}(\boldsymbol{\eta})$ is the transformation from the body fixed frame to the e frame.

4 ROV Modeling

Hydrodynamics is the terminology of submerged or partially submerged bodies subject to forces and torques. Hydrodynamic model structures can be derived using first principles of physics and some unknown parameters may be identified with e.g., bollard pull or towing tank experiments. The parameters in the models are in general time varying, however an alternative is to estimate constant models for different regions of the ROV movement, such as uniform acceleration, uniform translation and hover, as suggested by Morrison (1994). This approach was considered in Millert and Tayamon (2009) in which several model structures for the Falcon were identified of which some will be described and utilized in the following sections. The six degrees of freedom (DOF) hydrodynamics of the ROV, using notation from Fossen (2002), is

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}(\boldsymbol{\eta})\boldsymbol{\nu}, \tag{5a}$$

$$\boldsymbol{M}\dot{\boldsymbol{\nu}} + \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} + \widetilde{\boldsymbol{w}} \tag{5b}$$

where $\boldsymbol{M}$ is the inertia matrix, $\boldsymbol{C}(\boldsymbol{\nu})$ is the Coriolis and centripetal matrix, $\boldsymbol{D}(\boldsymbol{\nu})$ is the hydrodynamic damping matrix, $\boldsymbol{g}(\boldsymbol{\eta})$ is the hydrostatic restoring force vector,

$\widetilde{w}$ is the vector of environment disturbances, such as currents, and

$$\boldsymbol{\tau} = \begin{bmatrix} \boldsymbol{\tau}_1 \\ \boldsymbol{\tau}_2 \end{bmatrix}, \tag{6}$$

where

$$\boldsymbol{\tau}_1 = \begin{bmatrix} X & Y & Z \end{bmatrix}^T, \tag{7a}$$

$$\boldsymbol{\tau}_2 = \begin{bmatrix} K & M & N \end{bmatrix}^T, \tag{7b}$$

is the vector of forces and torques related to the propulsion. In previous work Aldebjer (2004); Millert and Tayamon (2009) model structures for (5) was identified using free decay experiments and least squares.

The inertia matrix is invertible and therefore the hydrodynamics (5) can be put into the form

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}(\boldsymbol{\eta})\boldsymbol{\nu}, \tag{8a}$$

$$\dot{\boldsymbol{\nu}} = \boldsymbol{M}^{-1}\left(\boldsymbol{\tau} - \boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu} - \boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu} - \boldsymbol{g}(\boldsymbol{\eta}) + \widetilde{\boldsymbol{w}}\right). \tag{8b}$$

Let the state vector be defined as $\boldsymbol{x} = [\boldsymbol{\eta}^T \ \boldsymbol{\nu}^T]^T$ and the input as $\boldsymbol{u} = \boldsymbol{\tau}$ and write (8) in compact form as

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{w}. \tag{9}$$

This notation will be useful later when estimation is concerned in Section 6.

4.1 Hydrodynamic Models

This section presents the hydrodynamic modeling approach that have been used for the different types of forces and torques.

Added Mass and Inertia

The loss of acceleration in water can be modeled as an added mass Aldebjer (2004); Fossen (2002). This applies both for the translation and rotation of the vehicle. The vehicle has three planes of symmetry and it is therefore fair to neglect the coupling terms in the added mass matrix

$$\boldsymbol{M_A} = -\operatorname{diag}\{X_{\dot{u}}, Y_{\dot{v}}, Z_{\dot{w}}, K_{\dot{p}}, M_{\dot{q}}, N_{\dot{r}}\}, \tag{10}$$

which gives the total inertia matrix, $\boldsymbol{M} = \boldsymbol{M_{RB}} + \boldsymbol{M_A}$, where $\boldsymbol{M_{RB}}$ is the rigid body inertia matrix. This approximation is applicable at the low speeds in which the Falcon operates and furthermore, off-diagonal parameters are difficult to estimate. The Coriolis and centripetal matrix $\boldsymbol{C}(\boldsymbol{\nu}) = \boldsymbol{C_{RB}}(\boldsymbol{\nu}) + \boldsymbol{C_A}(\boldsymbol{\nu})$, where $\boldsymbol{C_{RB}}(\boldsymbol{\nu})$ can be calculated from the mass matrix

$$\boldsymbol{C}(\boldsymbol{\nu}) = \begin{bmatrix} \boldsymbol{0} & -S(\boldsymbol{M}_{11}\boldsymbol{\nu}_1 + \boldsymbol{M}_{12}\boldsymbol{\nu}_2) \\ -S(\boldsymbol{M}_{11}\boldsymbol{\nu}_1 + \boldsymbol{M}_{12}\boldsymbol{\nu}_2) & -S(\boldsymbol{M}_{21}\boldsymbol{\nu}_1 + \boldsymbol{M}_{22}\boldsymbol{\nu}_2) \end{bmatrix}, \tag{11}$$

where

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{M}_{11} & \boldsymbol{M}_{12} \\ \boldsymbol{M}_{21} & \boldsymbol{M}_{22} \end{bmatrix}, \qquad S(\boldsymbol{a}) = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix}. \tag{12}$$

This is an off-diagonal structure that introduces desired coupling between the states.

Gravity

The gravity is called a restoring force and it can be modeled as

$$\begin{aligned} \boldsymbol{g}_{1,G} &= R^{be} \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}, \\ \boldsymbol{g}_{2,G} &= \boldsymbol{0} \end{aligned} \tag{13}$$

where $R^{be} = R(\boldsymbol{q})$ denotes a rotation from the e frame to the b frame and $g \approx 9.81\text{m/s}^2$ in Sweden. The force of gravity is affecting the vehicle in the centre of mass and therefore the moment from the gravity force is zero.

Buoyancy

Another restoring force is buoyancy which acts on the vehicle and prevents it from sinking. Buoyancy arises when a body moves the fluid surrounding itself and take its place. Buoyancy forces and torques can be modeled as

$$\boldsymbol{g}_{1,B} = R^{be} \begin{bmatrix} 0 \\ 0 \\ -g\rho V \end{bmatrix}, \tag{14a}$$

$$\boldsymbol{g}_{2,B} = \boldsymbol{r}^{cb} \times \boldsymbol{g}_{1,B}, \tag{14b}$$

where the vector $\boldsymbol{r}^{cb}$ is pointing from the centre of mass to the centre of buoyancy, ρ is the water density and V is the displaced water volume. In most underwater vehicles the centre of buoyancy is above the centre of gravity since this will give a restoring torque on the vehicle which helps stabilizing the vehicle in roll and pitch. In most underwater applications the vehicle has almost the same buoyancy and gravity magnitude since it otherwise would sink to the bottom or reach the surface. Therefore, the buoyancy force is set to $\boldsymbol{g}_{1,B} = -\boldsymbol{g}_{1,G}$, since the vehicle then is neutral in the water. In summary, the restoring force and moments vector is

$$\boldsymbol{g}(\eta) = \begin{bmatrix} \boldsymbol{g}_{1,G} + \boldsymbol{g}_{1,B} \\ \boldsymbol{g}_{2,B} + \boldsymbol{g}_{2,B} \end{bmatrix} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{g}_{2,B} \end{bmatrix}. \tag{15}$$

Drag

The damping forces acting on a marine vessel is described as a sum of potential damping, skin friction, wave drift damping and vortex shedding, these are generally non-linear and difficult to model. At moderate speed it is custom to assume

uncoupled motion, hence, the damping matrix $D(\nu)$ is diagonal. The drag force model matrix is then

$$\begin{aligned} \boldsymbol{D}_{1,D} = &- \operatorname{diag}\{X_u, Y_v, Z_w\} \\ &- \operatorname{diag}\{X_{u|u|}|u|, Y_{v|v|}|v|, Z_{w|w|}|w|\}, \end{aligned} \tag{16}$$

assuming both linear and quadratic damping as in e.g., Eng et al. (2008). As seen in (16) the drag force is assumed to affect the system independently in the three axes. The drag force is assumed to affect the vehicle in the centre of mass and therefore the drag force will not produce any moment. However, the rotational drag will produce a moment according to

$$\begin{aligned} \boldsymbol{D}_{2,D} = &- \operatorname{diag}\{K_p, M_q, N_r\} \\ &- \operatorname{diag}\{K_{p|p|}|p|, M_{q|q|}|q|, N_{r|r|}|r|\}. \end{aligned} \tag{17}$$

The complete damping model is then

$$\boldsymbol{D}(\boldsymbol{\nu}) = \begin{bmatrix} \boldsymbol{D}_{1,D} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{D}_{2,D} \end{bmatrix}. \tag{18}$$

Thrusters

Thruster modeling is a thoroughly studied area with many alternatives proposed in the literature and one of the most common and simplest models describe the output force as quadratic function of the input voltage (or propeller revolution rate) as

$$\tau = Cu|u| \tag{19}$$

where C is a constant and the absolute value accounts for the propeller direction. Previous work Millert and Tayamon (2009) has suggested a slightly different model for the thrusters on this vehicle. This is a dynamic model with a rise time of approximately 120ms which was estimated in bollard pull tests using a load cell measuring the produced force. This dynamics is fast compared to the rest of the system and is therefore neglected. The actual force produced when the vehicle is moving may however differ from stationary experiments to a large extent Indiveri (1998). The force from the thrusters as function of the input signals can be described by

$$\tau_{1,i} = \left(2.98 \cdot 10^{-4} u_i^3 - 0.16 u_i|u_i| + 0.32 u_i\right), \tag{20}$$

where $\tau_{1,i}$ is the force from thruster $i = 1, 2 \ldots, 5$ and u_i is the measured input signal for thruster i. It was noted that the thrusters saturate at 90% of its maximum input producing 127N. The force expressed in the ROV coordinate frame is

$$\boldsymbol{\tau}_{1,i} = \tau_{1,i} \boldsymbol{v}_i \tag{21}$$

$\boldsymbol{v}_i$ is a unit vector pointing in the direction as the thruster exerts its force. This force also create a moment on the vehicle which can be calculated by

$$\boldsymbol{\tau}_{2,i} = \boldsymbol{r}_i^T \times \boldsymbol{\tau}_{1,i}, \tag{22}$$

where $\boldsymbol{r}_i^T$ is the vector pointing from the origin to the location of the i^{th} thruster. All forces and moments generated by the thrusters may be collected as

$$\boldsymbol{\tau}_1 = \sum_{i=1}^{5} \boldsymbol{\tau}_{1,i}, \tag{23a}$$

$$\boldsymbol{\tau}_2 = \sum_{i=1}^{5} \boldsymbol{\tau}_{2,i}. \tag{23b}$$

4.2 Discretization

The hydrodynamic model (8) is discretized with Eulers's method which gives

$$\boldsymbol{\eta}_{t+1} = \boldsymbol{\eta}_t + T\boldsymbol{J}(\boldsymbol{\eta}_t)\boldsymbol{\nu}_t, \tag{24a}$$

$$\boldsymbol{\nu}_{t+1} = \boldsymbol{\nu}_t + T\boldsymbol{M}^{-1}\left(\boldsymbol{\tau}_t - \boldsymbol{C}(\boldsymbol{\nu}_t)\boldsymbol{\nu}_t - \boldsymbol{D}(\boldsymbol{\nu}_t)\boldsymbol{\nu}_t - \boldsymbol{g}(\boldsymbol{\eta}_t)\right) + \boldsymbol{w}_t, \tag{24b}$$

and it is assumed that the control input $\boldsymbol{\tau}$ is constant over the sampling interval T. A simplified notation for (24) is

$$\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}) + \boldsymbol{w}_t, \tag{25}$$

where $\boldsymbol{u} = \boldsymbol{\tau}$ and T is implicit. The process noise is $\boldsymbol{w}_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{Q}_t)$ and a simple discretization is $\boldsymbol{Q}_D \triangleq T\boldsymbol{Q}_t$. The quaternion should also be normalized in each time step whereas otherwise it will not represent an orientation.

4.3 Kinematic model

As a comparison the sensor data will also be processed in an Extended Kalman Filter (EKF) using a kinematic model with inertial measurement signals rather than control inputs. The state vector composed in the e and b frame is

$$\boldsymbol{x} = \begin{bmatrix} (\boldsymbol{p}^e)^T & (\boldsymbol{v}^b)^T & (\boldsymbol{a}^b)^T & (\boldsymbol{q}^{be})^T & (\boldsymbol{\omega})^T \end{bmatrix}^T, \tag{26}$$

denoting the position, velocity, acceleration, the rotation (parametrized with a unit quaternion) and the angular velocity, respectively. As in the hydro dynamic model the velocities and accelerations are expressed in the body frame introducing a few extra non-linearities. On the other hand when using this setup then the same sensor models, see Section 5, can be used. The continuous-time non-linear dynamic model for the states is

$$\dot{\boldsymbol{p}}^e = R^{eb}\boldsymbol{v}^b, \tag{27a}$$

$$\dot{\boldsymbol{v}}^b = \boldsymbol{a}^b - \boldsymbol{\omega} \times \boldsymbol{v}^b, \tag{27b}$$

$$\dot{\boldsymbol{a}}^b = -\boldsymbol{\omega} \times \boldsymbol{a}^b + \boldsymbol{w}_{\boldsymbol{a}_t}, \tag{27c}$$

$$\dot{\boldsymbol{q}}^{be} = \frac{1}{2}S(\boldsymbol{\omega})\boldsymbol{q}, \tag{27d}$$

$$\dot{\boldsymbol{\omega}} = \boldsymbol{w}_{\boldsymbol{\omega}_t}. \tag{27e}$$

where, $\boldsymbol{w}_{\boldsymbol{a}_t}$ and $\boldsymbol{w}_{\boldsymbol{\omega}_t}$ is noise accounting for the unknown jerk and the unknown angular acceleration, respectively. This acceleration state is also used in the hydrodynamic model since it is needed for the accelerometers. Furthermore, $S(\boldsymbol{\omega}_t)\,\boldsymbol{q}_t^{be}$ is the quaternion product of the angular velocity and the unit quaternion in matrix vector notation where $S(\boldsymbol{\omega})$ is defined as

$$S(\boldsymbol{\omega}) = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix}. \tag{28}$$

Again using Euler's method on (12) with sampling interval T gives

$$\boldsymbol{p}_{t+1}^e = \boldsymbol{p}_t^e + T R_t^{eb} \boldsymbol{v}_t^b, \tag{29a}$$

$$\boldsymbol{v}_{t+1}^b = \boldsymbol{v}_t^b + T(\boldsymbol{a}_t^b - \boldsymbol{\omega}_t \times \boldsymbol{v}_t^b) \tag{29b}$$

$$\boldsymbol{a}_{t+1}^b = \boldsymbol{a}_t^b - T(\boldsymbol{\omega}_t \times \boldsymbol{a}_t^b) + T\boldsymbol{w}_{\boldsymbol{a}_t} \tag{29c}$$

$$\boldsymbol{q}_{t+1}^{be} = \boldsymbol{q}_t^{be} + \frac{T}{2} S(\boldsymbol{\omega}_t) \boldsymbol{q}_t^{be} \tag{29d}$$

$$\boldsymbol{\omega}_{t+1} = \boldsymbol{\omega}_t + T\boldsymbol{w}_{\boldsymbol{\omega}_t}, \tag{29e}$$

This kinematic model can be expressed as

$$\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t) + \boldsymbol{w}_t, \tag{30}$$

where $f(\cdot)$ is a nonlinear function of the state and $\boldsymbol{w}_t$ is noise.

5 Sensor Models

The Falcon ROV sensor suite consists of a vector magnetometer, a vector gyroscope, a vector specific force sensor, a Doppler velocity log (DVL) and a hydrostatic pressure sensor (HPS). The typical equations for sensors are

$$y = h(\boldsymbol{x}_t) + \boldsymbol{e}_t, \tag{31}$$

where $h(\cdot)$ is a nonlinear function of the state $\boldsymbol{x}_t$ and $\boldsymbol{e}_t$ is noise where subscript t denotes time dependency. Selection of such sensors can be crucial since size, weight, price, energy consumption and other physical limitations need to be considered. Moreover, cheaper sensors do not offer the same accuracy as in the higher range. Models of strapdown inertial sensors can be found in most books on navigation, see e.g., Titterton and Weston (1997).

5.1 Inertial Measurement Unit and Magnetometer

The inertial measurement unit (IMU) provides measurements of the specific force and angular velocity.

Specific Force

The measured specific force is the sum of the linear acceleration and the force from the gravitational field which is independent of reference frame. The orientation of the sensor plays an important role since the gravitational field will affect the measurements and an error in the estimated sensor orientation is manifested as an acceleration error. Also, the measured force due to the rotational velocity depends on the mounting of the sensor relative to the body rotational centre. This offset is modeled by the vector $\boldsymbol{r}^{IMUb}$ which is pointing to the sensor with the origin at the vehicle's centre of mass. A measurement model for the acceleration is

$$y_{\boldsymbol{a}} = R^{IMUb}\left(\boldsymbol{a}_t^b - R_t^{be}\boldsymbol{g}^e + \boldsymbol{\omega}_t \times \left(\boldsymbol{\omega}_t \times \boldsymbol{r}^{IMUb}\right)\right) + \boldsymbol{e}_{\boldsymbol{a}_t}, \tag{32}$$

where $\boldsymbol{g}^e$ is the gravity vector and $\boldsymbol{r}^{IMUb}$ is the vector from the body origin to the IMU centre and $\boldsymbol{e}_{\boldsymbol{a}_t}$ is measurement noise which also includes the unknown angular acceleration.

Gyroscope

The triad of gyroscopes measures the sensor unit's angular velocity and a measurement model for this sensor is

$$y_{\boldsymbol{\omega}} = R^{IMUb}\boldsymbol{\omega}_t + \boldsymbol{e}_{\boldsymbol{\omega}_t}, \tag{33}$$

where R^{IMUb} is the rotation matrix from the body frame to the IMU frame and $\boldsymbol{e}_{\boldsymbol{\omega}_t}$ is measurement noise. This model is very simple and a common extension is to add a slowly time varying bias to capture sensor drift which may be due to e.g., temperature changes.

Magnetometer

The vector magnetometer is also contained in the IMUs' housing and therefore the origin and rotation w.r.t. the body frame is considered to be the same. The measurement model for the magnetometer is

$$y_{\boldsymbol{m}} = R^{IMUb}R_t^{be}\boldsymbol{n}_{np}^e + \boldsymbol{e}_{\boldsymbol{m}_t}, \tag{34}$$

where R^{IMUb} is the rotation matrix from the body frame to the IMU frame, $\boldsymbol{n}_{np}^e$ is the normalised local earth magnetic field vector and $\boldsymbol{e}_{\boldsymbol{m}_t}$ is the measurement noise. At the location where the measurements were acquired, the magnetic field was $[15661,\ 1103,\ 47978]nT$ denoting north, east and vertical components (NED), respectively.

5.2 Doppler Velocity Log

The Doppler Velocity Log (DVL) measure the linear velocity of the vehicle relative to the seabed. The sensor is an acoustic device that sends sound pings through the water and measures the reflections at the seabed (RD , 2001) using a multibeam echo-sounder. In these reflections there will be a doppler shift which corresponds to the velocity of the vehicle. The output from the sensor is represented

in a sensor fixed coordinate system, *DVL*-frame. An expression is needed for the velocity at the point in which the sensor is mounted. The distance from the vehicle's centre of rotation, which is assumed to coincide with the centre of mass, and the sensor is denoted $\boldsymbol{r}^{DVLb}$. The measurement model for the DVL is

$$y_{\boldsymbol{v}} = R^{DVLb}(\boldsymbol{v}_t^b + \boldsymbol{\omega}_t \times \boldsymbol{r}^{DVLb}) + \boldsymbol{e}_{v_t}, \tag{35}$$

where R^{DVLb} is the rotation matrix from the body frame to the DVL frame, r^{DVLb} is the vector from the vehicle body origin to the DVL origin and $\boldsymbol{e}_{v_t}$ is measurement noise. The measurement error in the DVL is mainly due to a scale error and this error is due to the uncertainty of the speed of sound in the water which depend on the water's salinity.

6 Estimation

In this section the estimation process using the onboard sensors and the hydrodynamic model is described.

6.1 Filtering

A standard method for non-linear filtering is to apply the Extended Kalman Filter (EKF) to the dynamic equations and the measurements, see e.g., Kailath et al. (2000), and this is often the preferred solution when dealing with attitude estimation using integrated inertial systems Crassidis et al. (2007). The EKF is summarized in Algorithm 1 below.

Algorithm 1 Extended Kalman Filter

Require an initial state, $\hat{x}_{0|0}$, and an initial state covariance, $P_{0|0}$.

1. Time Update

$$\hat{\boldsymbol{x}}_{t|t-1} = f(\hat{\boldsymbol{x}}_{t-1|t-1}, \boldsymbol{u}_t), \tag{36a}$$

$$\boldsymbol{P}_{t|t-1} = \boldsymbol{F}_t \boldsymbol{P}_{t-1|t-1} \boldsymbol{F}_t^T + \boldsymbol{Q}_t, \tag{36b}$$

2. Measurement Update

$$\boldsymbol{K}_t = \boldsymbol{P}_{t|t-1} \boldsymbol{H}_t^T (\boldsymbol{H}_t \boldsymbol{P}_{t|t-1} \boldsymbol{H}_t^T + \boldsymbol{R}_t)^{-1}, \tag{37a}$$

$$\hat{\boldsymbol{x}}_{t|t} = \hat{\boldsymbol{x}}_{t|t-1} + \boldsymbol{K}_t (y_t - h(\hat{\boldsymbol{x}}_{t|t-1})), \tag{37b}$$

$$\boldsymbol{P}_{t|t} = \boldsymbol{P}_{t|t-1} - \boldsymbol{K}_t \boldsymbol{H}_t \boldsymbol{P}_{t|t-1}. \tag{37c}$$

In Algorithm 1 the Jacobian matrices are defined according to

$$\boldsymbol{F}_t \triangleq \left. \frac{\partial f(\boldsymbol{x}_t, \boldsymbol{u}_t)}{\partial \boldsymbol{x}_t} \right|_{(\boldsymbol{x}_t, \boldsymbol{u}_t) = (\hat{\boldsymbol{x}}_{t-1|t-1}, \boldsymbol{u}_t)}, \tag{38}$$

$$\boldsymbol{H}_t \triangleq \left. \frac{\partial h(\boldsymbol{x}_t)}{\partial \boldsymbol{x}_t} \right|_{(\boldsymbol{x}_t) = (\hat{\boldsymbol{x}}_{t|t-1})}, \tag{39}$$

furthermore, $\boldsymbol{Q}_t$ and $\boldsymbol{R}_t$ are the covariance matrices of $\boldsymbol{w}_t$ and $\boldsymbol{e}_t$, respectively.

(a) The lattice

(b) The crayfish cage.

Figure 3: *The lattice marks one end of the reference path and the crayfish cage marks the other end.*

6.2 Synchronization

The sensors and control inputs are sampled at different rates and at different time instants. It is crucial that these signals are synchronized since otherwise the performance of the filter is degraded. The synchronization is handled in a post-processing step using the time-stamps from each sensor and the control inputs. To handle the different sampling rates in the EKF the following is done:

At a time instant t_k

1. Find the next signal (sensor(s) and/or control input) at time t_{k+1}.
2. If it is a control input, perform a time update according to (36) over the interval $T = t_{k+1} - t_k$ using the control input $\boldsymbol{u}_{t_k}$.
3. If it is a sensor signal return to the step 2) and then perform a measurement update according to (37).
4. Return to 1).

7 Results

Data was collected in Lake Vättern, Sweden, under rather good weather conditions. A path, which is marked by a cable, was followed each run where a lattice marks one end of the cable and a cray fish cage marks the other, see Fig. 3. Note that the ROV was only steered such that it finished at the other end of the cable from where it started, i.e., the cable was used as a reference for the operator. The cable's position was measured using a ROV equipped with a GPS receiver which was traveling in surface mode. The path is close to the shore and without any nearby inlets creating currents. Data was processed offline as described in the previous section.

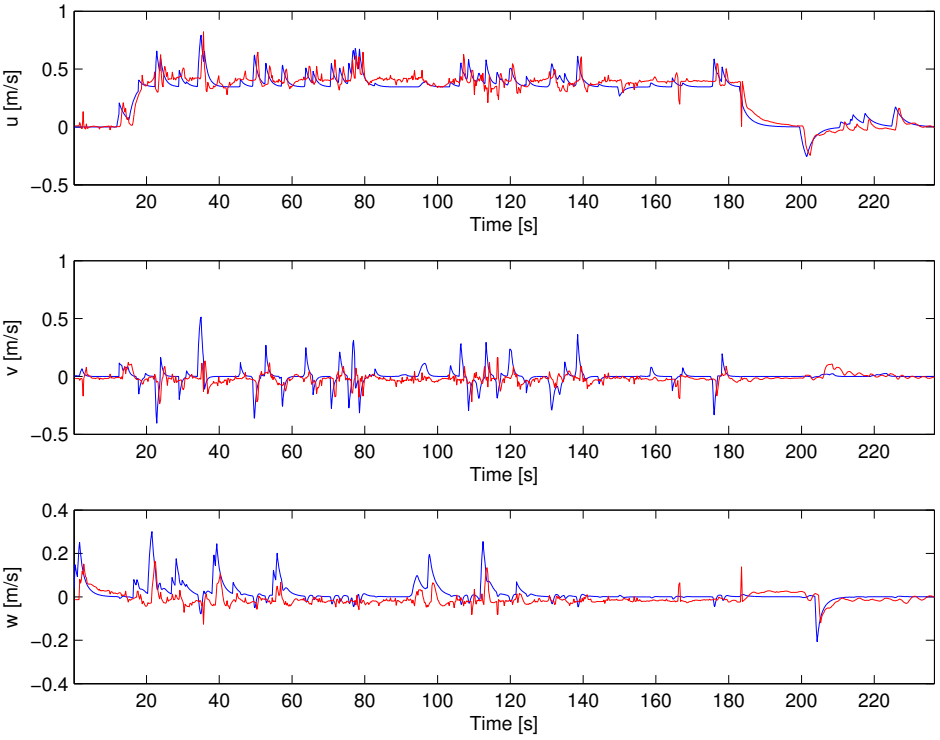


Figure 4: *A comparison of the velocities from the hydrodynamic model in red compared to the sensor readings from the DVL in blue shows that the model is very accurate. Some artifacts may be explained by currents and turbulence not accounted for in the model and there is also noise in the DVL measurements.*

7.1 Velocity Model

The estimated velocity using the hydrodynamic model (8) compared to the velocity measured by the DVL can be seen in Fig. 4. The DVL is fairly accurate and can hence serve as a reliable reference. The model explains both the dynamic behaviour and constant motion very well. Also note the consistency with the measured surge speed when the ROV is moving slightly backwards at about 200s.

7.2 Angular Velocity Model

The simulated yaw rate compared to the gyroscope can be seen Fig. 5 where it is obvious that the hydrodynamic model does not capture the true rate particularly well. An explanation for this could be that the drag model is rather rough approximation since the parameters are in general speed dependent and coupled. With data in which the ROV was doing a clockwise turn at slowly creeping surge

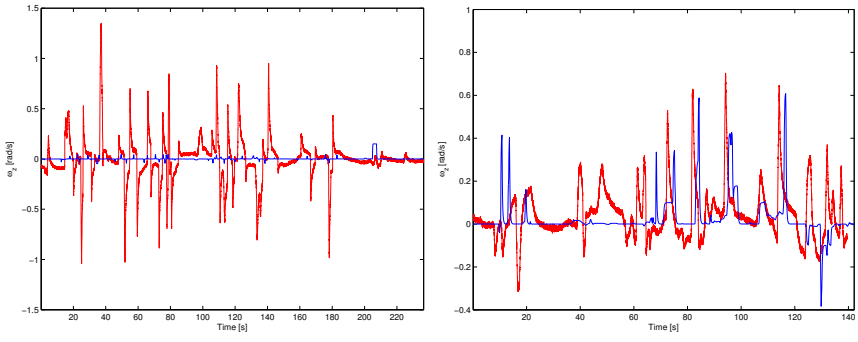


Figure 5: *Left: The yaw rate from the hydrodynamic model in red compared to the sensor reading from the gyroscope in blue shows that the model do not capture the rate dynamics satisfactorily. Right: The yaw rate from a data set with a slowly creeping surge speed is a lot closer to the likely yaw rate.*

speed the yaw rate is much closer to the rate given by the gyroscopes, see Fig. 5.

7.3 Position Estimates

The angular velocity models were simply too poor to be used for estimation of the position, instead the hydrodynamics were aided with the onboard consumer grade inertial and magnetometer sensor MTI from XSens. A view of the trajectory estimated trajectory using the aided hydrodynamic model (8), the kinematic model (12) with DVL and the reference trajectory can be seen in Fig. 6. The reference trajectory marks the coordinates of the cable and it was measured using another ROV equipped with a more sophisticated navigation system. The ROV was not controlled such that it ran along the cable the whole track. The position estimate with the hydrodynamic model is comparable to the kinematic model meaning that the DVL can be aided using the hydrodynamics.

8 Conclusion

We have compared two different Extended Kalman filters for navigation of a ROV. The first one is based on inertial navigation principles supported with a DVL, where a nearly constant velocity model is used for describing the motion. The second one is based on a hydrodynamical model with the five thruster controls as inputs, and where the inertial sensors can be used as feedback. It was shown that the available models for this ROV for the angular dynamics were not good enough to compete with inertial navigation. On the positive side, the translational dynamics turned out to be comparable to the inertial system in accuracy. This is particularly useful in situations when the doppler log is not reliable. In future work it would be interesting to find better models for the angular rates and possibly different ones for different modes as the results indicates.

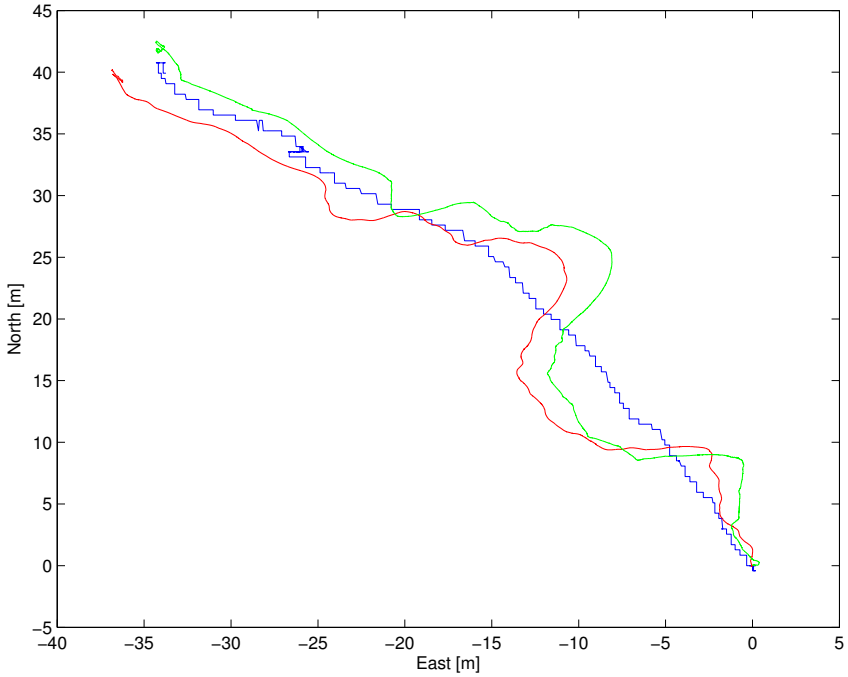


Figure 6: *The trajectory using the hydrodynamic model without the DVL in red, the kinematic model* (12) *using all sensors in green and the blue curve is the reference trajectory.*

Bibliography

L. Aldebjer. Hardware in the loop simulator for remotely operated vehicle. Master's thesis, Chalmers university of technology, December 2004.

A. Alessandri, M. Caccia, G. Indiveri, and G. Veruggio. Application of LS and EKF techniques to the identification of underwater vehicles. In *Proceedings of the IEEE International Conference on Control Applications*, Trieste, Italy, September 1998.

B. Allen, T. Austin, N. Forrester, R. Goldsborough, A. Kukulya, G. Packard, M. Purcell, and R. Stokey. Autonomous Docking Demonstrations with Enhanced REMUS Technology. In *OCEANS 2006*, pages 1–6, sept. 2006. doi: 10.1109/OCEANS.2006.306952.

K. Ånonsen and O. Hallingstad. Terrain Aided Underwater Navigation Using Point Mass and Particle Filters. In *Position, Location, And Navigation Symposium, 2006 IEEE/ION*, pages 1027–1035, 25-27 2006. doi: 10.1109/PLANS.2006.1650705.

M. Caccia, G. Indiveri, and G. Veruggio. Modeling and identification of open-frame variable configuration unmanned underwater vehicles. *Oceanic Engineering, IEEE Journal of*, 25(2):227–240, apr 2000. ISSN 0364-9059. doi: 10.1109/48.838986.

J. Crassidis, F. Markley, and Y. Cheng. A Survey of Nonlinear Attitude Estimation Methods. *AIAA Journal of Guidance, Control, and Dynamics*, 30(1):12–28, Jan.-Feb. 2007.

Y. Eng, W. Lau, E. Low, G. Seet, and C. Chin. Estimation of the Hydrodynamics Coefficients of an ROV using Free Decay Pendulum Motion. *Engineering Letters*, 16(3), August 2008.

R. Eustice, H. Singh, and J. Leonard. Exactly Sparse Delayed-State Filters for View-Based SLAM. *Robotics, IEEE Transactions on*, 22(6):1100–1114, dec. 2006. ISSN 1552-3098. doi: 10.1109/TRO.2006.886264.

K. Fauske, F. Gustafsson, and O. Hegrenaes. Estimation of AUV dynamics for sensor fusion. In *Information Fusion, 2007 10th International Conference on*, pages 1–6, july 2007. doi: 10.1109/ICIF.2007.4408044.

T. I. Fossen. *Marine Control Systems, Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles*. Marine Cybernetics, 1 edition, 2002. ISBN 82-9235-00-2.

G. Indiveri. *Modelling and Identification of Underwater Robotic Systems*. PhD thesis, University of Genova, Italy, 1998.

B. Jalvning, K. Gade, K. Svartveit, A. Willumsen, and R. Sørhagen. DVL Velocity Aiding in the HUGIN 1000 Integrated Inertial Navigation System. *Modeling, Identification & Control*, 25(4):223–236, October 2004.

T. Kailath, A. H. Sayed, and B. Hassibi. *Linear Estimation*. Prentice-Hall, Upper Saddle River, New Jersey, 2000.

R. Karlsson and F. Gustafsson. Bayesian Surface and Underwater Navigation. *IEEE Transactions on Signal Processing*, 54(11):4204–4213, Nov. 2006.

J. Kim, K. Kim, H. Choi, W. Seong, and K.-Y. Lee. Estimation of hydrodynamic coefficients for an AUV using nonlinear observers. *Oceanic Engineering, IEEE Journal of*, 27(4):830–840, Oct. 2002. ISSN 0364-9059. doi: 10.1109/JOE.2002.805098.

J. Kuipers. *Quaternions and Rotations Sequences*. Princeton University Press, 2002.

I. Mahon, S. Williams, O. Pizarro, and M. Johnson-Roberson. Efficient View-Based SLAM Using Visual Loop Closures. *Robotics, IEEE Transactions on*, 24 (5):1002 –1014, oct. 2008. ISSN 1552-3098. doi: 10.1109/TRO.2008.2004888.

M. Mandt, K. Gade, and B. Jalving. Integrating DGPS-USBL position measurements with inertial navigation in the HUGIN 3000 AUV. In *in Proceedings of the 8th Saint Petersburg Conference on Integrated Navigation Sysemts*, Saing Petersburg, Russia, May 2001.

C. Millert and S. Tayamon. Method development for identification of coefficients in a Remotly Operated Vehicle simulator. Master's thesis, Faculty of Science and Technology, Uppsala University, April 2009.

A. Morrison. System identification and state reconstruction for autonomous navigation of an underwater vehicle in an acoustic net. Master's thesis, Massachusetts Institute of Technology, 1994.

F. Pappalardi, S. Dunham, M. LeBlang, T. Jones, J. Bangert, and G. Kaplan. Alternatives to GPS. In *OCEANS, 2001. MTS/IEEE Conference and Exhibition*, volume 3, pages 1452–1459, Honolulu, HI, USA, Nov 2001. doi: 10.1109/OCEANS.2001.968047.

Navigator ADCP/DVL Technical Manual. RD Instruments, 2001.

M. A. Skoglund, K. Jönsson, and F. Gustafsson. Modeling and sensor fusion of a remotely operated underwater vehicle. In *Proceedings of the 15th International Conference on Information Fusion (FUSION), Singapore, 9-12 July 2012*, pages 947–954. IEEE, 2012.

D. Smallwood and L. Whitcomb. Adaptive identification of dynamically positioned underwater robotic vehicles. *Control Systems Technology, IEEE Transactions on*, 11(4):505–515, July 2003. ISSN 1063-6536. doi: 10.1109/TCST.2003.813377.

SNAME. Nomenclature for treating the motion of a submerged body through fluid. Technical Report 1-5, Society of Naval Architects and Marine Engineers (SNAME), April 1950.

A. Taino, R. Sutton, A. Lozowicki, and W. Naeem. Observer Kalman filter identification of an autonomous underwater vehicle. *Control Engineering Practice*, 15(6):727–739, 2007.

D. Titterton and J. Weston. *Strapdown inertial navigation technology*. IEE, 1997.

Paper D

Initialisation and Estimation Methods for Batch Optimisation of Inertial/Visual SLAM

Authors: Martin A. Skoglund, Zoran Sjanic and Fredrik Gustafsson

Edited version of the paper:

M. A. Skoglund, Z. Sjanic, and F. Gustafsson. Initialisation and estimation methods for batch optimisation of inertial/visual SLAM. Submitted to IEEE Transactions on Aerospace and Electronic Systems, June 2014.

Preliminary version:

Technical Report LiTH-ISY-R-3065, Dept. of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden.

Initialisation and Estimation Methods for Batch Optimisation of Inertial/Visual SLAM

Martin A. Skoglund*, Zoran Sjanic* ** and Fredrik Gustafsson*

*Dept. of Electrical Engineering,
Linköping University,
SE–581 83 Linköping, Sweden
{ms, zoran,
fredrik}@isy.liu.se

**Dept. of Flight Data and Navigation,
Saab Aeronautics
SE-581 88 Linköping, Sweden

Abstract

Inertial/visual SLAM aims at estimating the pose of a camera and a map of landmarks detected in the images, using support from an inertial measurement unit (IMU). Some of the most competitive approaches for SLAM and its computer vision counterpart, Structure From Motion (SFM), are based on batch formulations such as GraphSLAM or Bundle Adjustment (BA). A major challenge in the implementation is the initialisation since these problems are inherently nonlinear and nonconvex. We propose a multi-step algorithm that solves a series of simple and almost uncoupled problems, often leading to linear solutions. It is believed that this leads to a robust algorithm which is simple to implement and fast to execute. The initialisation method is demonstrated on simulated data and a small feasibility study on experimental data using an industrial robot, to get access to ground truth, is also performed.

1 Introduction

Simultaneous Localization And Mapping (SLAM) concerns the problem of jointly estimating the trajectory of a moving platform and a map of landmarks observed from the platform. The platform trajectory is inherently constrained by its motion dynamics, so model-based filtering approaches have been used extensively, in particular in the robotics area Durrant-Whyte and Bailey (2006); Bailey and Durrant-Whyte (2006). Further, measurements from an Inertial Measurement Unit (IMU) enable a more accurate motion model based on inertial navigation principles Titterton and Weston (1997). Besides providing more reliable predictions of the platform position, the IMU also provides a metric scale to the trajectory and the map.

The first filtering SLAM methods were relying on Extended Kalman Filters (EKF) P. Moutarlier and R. Chatila (1989); Smith et al. (1990) which scale badly with the

dimension of the map whereas particle methods FastSLAM Montemerlo et al. (2002, 2003) does not scale well with the dimension of the vehicle state. More recent methods are based on batch formulations Dellaert, F. and Kaess, M. (2006); Thrun and Montemerlo (2006); Grisetti et al. (2007); Jung and Taylor (2001) with clever data structures allowing real-time execution M. Kaess and A. Ranganathan and F. Dellaert (2008).

SLAM batch approaches in computer vision are known as Bundle Adjustment (BA), Hartley and Zisserman (2004); Triggs et al. (2000); Agarwal et al. (2009) which dates back to the work of Brown (1958). BA is used as a final polish to improve an estimate or as intermediate step in sequential algorithms such as Structure From Motion (SFM) Fitzgibbon and Zisserman (1998); Taylor et al. (1991). When BA and SLAM is formulated as Maximum Likelihood (ML) or Maximum a Posteriori (MAP) with a Gaussian noise assumption the results is a nonlinear least squares (NLS) estimation problem. However, BA methods normally do not utilise a motion model and other sensors, besides cameras, are rare. For instance, the visual sequential methods Konolige and Agrawal (2008); Strasdat et al. (2010) approximate the BA by keeping a set of keyframes selected heuristically. The work of Klein and Murray (2007) combines local feature tracking and BA in two threads and is still considered state-of-the-art.

GraphSLAM was established in Lu and Milios (1997) as a solution to batch SLAM via a pose graph and was further explored by Thrun and Montemerlo (2006); Grisetti et al. (2007) and many others. Pose graphs can be formally obtained by marginalisation of landmark parameters introducing relative pose constraints through shared observations. This is an approximation since the optimisation variables only consider the pose graph and not the landmarks. These methods have been proven very useful for 2D ground robotics with laser scanners but have not been used extensively on BA problems. Methods to initialise such systems in a linear manner, similar to the ones used in computer vision, were previously explored in Carlone et al. (2011); Dellaert and Stroupe (2002).

The working horse of these algorithms is Gauss-Newton like optimisation solvers with efficient structure utilisation and domain specific tricks as the back-bone for scaling to huge problems. As with all nonlinear optimisation schemes, NLS-SLAM, BA, and GraphSLAM rely on accurate initialisation to avoid the many local minima in their corresponding NLS cost functions.

Inertial/visual SLAM Kneip et al. (2011b); Martinelli (2012); Bryson et al. (2009); Lupton and Sukkarieh (2012) is an attempt to unify the advantages of the SLAM solutions from the computer vision and robotics communities, where both camera frames and IMU measurements are utilised. See Figure 1 for an illustration of the setup. It is the purpose of this contribution to provide a general, robust and fast initialisation algorithm for inertial/visual SLAM. It is a multi-step algorithm, similar to work in Sinha et al. (2010) but which only treats camera based SFM, thus ad-hoc in its nature, but each step has a natural problem formulation and a simple solution. The intention of this algorithm is that it can facilitate faster and simpler solutions to the impressing applications that already exist for

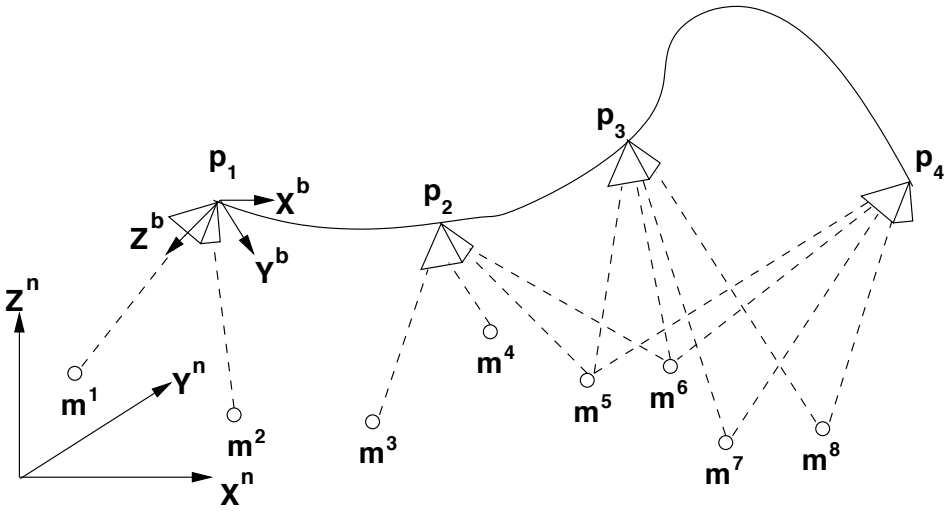


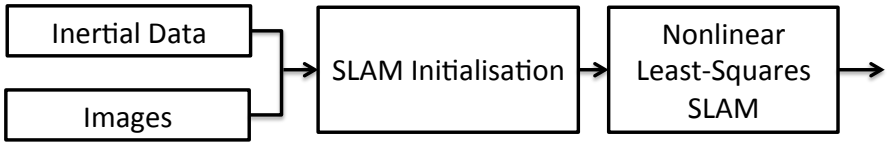
Figure 1: *The setup with inertial and visual sensors. The camera is observing the environment represented by point landmarks, $m^1, \ldots, m^8$, and the inertial sensors are measuring accelerations and angular rates in the body coordinate system, (b), which moves together with the camera. Also, a global, fixed navigation coordinate system, (n), is drawn.*

camera-only SLAM. Optionally, the nonlinear refinement can be skipped if the results from the initialisation are already satisfactory.

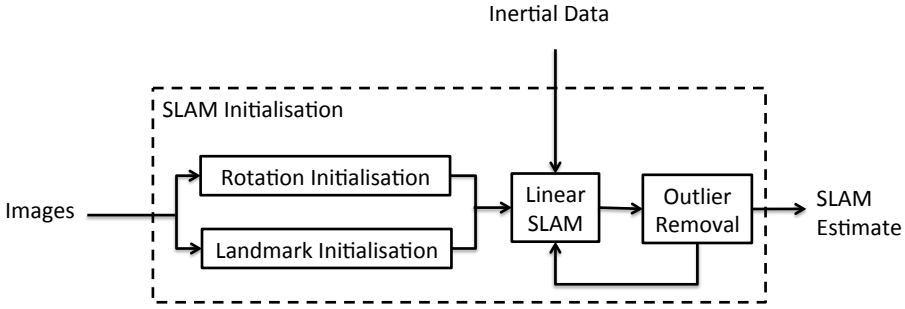
The initialisation is utilising that the reprojection error is conditionally linear given that the platform's rotations are known. In this linear formulation the camera measurement noise is parameter dependent, see Sinha et al. (2010); Martinelli (2012); Kneip et al. (2011b), and is treated in an iterative fashion.

The data association, which is also an important and difficult problem, is based on clustering, see for instance Hastie et al. (2009), of feature tracks. The feature tracks, see e.g., Thormählen et al. (2008), are estimated as a linear program formulation of the assignment problem. Outliers are efficiently eliminated using an iterative procedure on the reprojection errors using the IMU data.

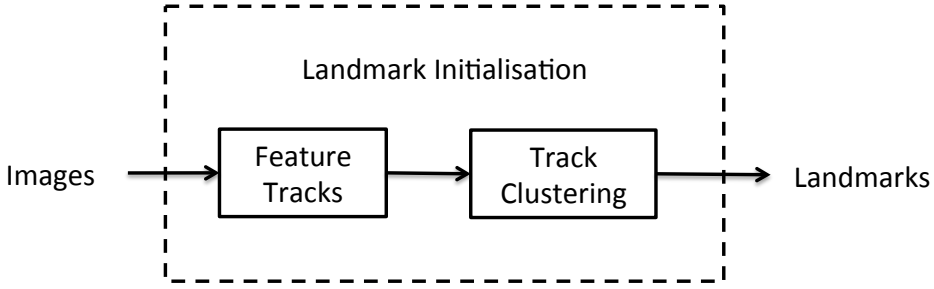
The block diagrams in Figure 2a serve as illustrative overview of the proposed method. The initialisation procedure computes a set of landmarks with its corresponding measurements (this includes the loop closure candidates), an estimate of the trajectory and the velocity using IMU data and a camera-only rotation estimate. The flow of this procedure is illustrated in Figure 2b. The landmark initialisation in Figure 2b represents the image processing operations such as feature tracking and track clustering. Feature tracks are extracted from matching correspondence pairs in the image sequence. The feature tracks are then clustered based upon their average feature descriptor in order to find loop closure candidates. The clustered feature tracks are then used to initialise 3D point land-



(a) High level abstraction of the computational flow.



(b) Camera and inertial data initialisation procedure.



(c) Image processing procedure.

Figure 2: *An overview of the proposed method. Most blocks corresponds to a subsection in Section 3 and Section 5 with the same name*

marks.

The paper is organised as follows; Section 4 describes the models of the different sensors used in the formulation of the problem. Section 3 handles the initialisation procedure based on the almost linear formulation of the visual/inertial SLAM problem. Here, all steps for the initialisation of the trajectory, orientation, landmarks and data association are described. Each subsection here represents a particular block in Figures 2b and 2c. In Section 5 a nonlinear refinement method is used where the initial point is given by the proposed initialisation procedure. In Section 5 some comments and discussion that motivate the linear initialisation method are given. Finally, results on both simulated and real data are shown in Sections 6 and 8.2 respectively, and some conclusions and future work are discussed in Section 9.

2 Models

The sensors of interest are monocular camera and 6-DOF inertial sensors, gyroscopes and accelerometers, contained in a single unit. A standard Cartesian 3D point landmark parametrisation is used and its measurement is given by the pinhole projection model. In this work we assume that both the camera and the relative pose of the camera optical center with respect to the center of the IMU are calibrated. The camera calibration implies that image pixel coordinates can be transformed to metric coordinates and all the inertial measurements can be assumed to measure the camera's rotation and acceleration.

2.1 Position and Orientation

Given the accelerations, $a = [a^x, a^y, a^z]^T$, and angular velocities, $\omega = [\omega^x, \omega^y, \omega^z]^T$, of a moving and rotating object expressed in the non-moving frame, the so called navigation (or world or earth) frame. The the position, velocity and orientation (parametrised as unit quaternion $q = [q_0, q_1, q_2, q_3]^T$, $q^T q = 1$) of the object in the navigation frame, $[p, v, q]$, can be written as a discrete time dynamic model as

$$p_t = p_{t-1} + T v_{t-1} + \frac{T^2}{2} a_t \tag{1a}$$

$$v_t = v_{t-1} + T a_t \tag{1b}$$

$$q_t = \exp\left(\frac{T}{2} S_\omega(\omega_t)\right) q_{t-1} \tag{1c}$$

where T is the sampling time, the skew-symmetric matrix

$$S_\omega(\omega) = \begin{bmatrix} 0 & -\omega^x & -\omega^y & -\omega^z \\ \omega^x & 0 & \omega^z & -\omega^y \\ \omega^y & -\omega^z & 0 & \omega^x \\ \omega^z & \omega^y & -\omega^x & 0 \end{bmatrix}, \tag{2}$$

parametrises the quaternion dynamics and here $\exp(\cdot)$ denotes the matrix exponential.

2.2 IMU Measurements

The IMU measures the specific force and rotation speed in a frame attached to the IMU body frame, denoted b. Usually these measurements are imperfect and contain both biases and measurement noise. The biases are assumed constant and this is usually only a good approximation for a short period of time since in practice biases will vary due to e.g., temperature. Under these assumptions the measurements can then be described as

$$y_t^a = \mathsf{R}^{be}(q_t)(a_t^e - g^e) + b_a + e_t^a \tag{3a}$$

$$y_t^\omega = \omega_t + b_\omega + e_t^\omega \tag{3b}$$

where $g^e = [0, 0, -g]$ is the local gravity vector expressed in the navigation frame, and $g \approx 9.82$, $\mathsf{R}^{be}(q)$ is the rotation matrix parametrisation of the quaternion and the measurement noises are assumed i.i.d. Gaussian with zero mean and time invariant covariances R_a and R_ω, i.e., $e_t^a \sim \mathcal{N}(0, R_a)$ and $e_t^\omega \sim \mathcal{N}(0, R_\omega)$.

2.3 Camera Measurements

The monocular camera is modeled as a standard pinhole camera, see cf. Hartley and Zisserman (2004). The camera calibration matrix and lens distortion need to be estimated prior to usage. Since the calibration and distortion are known the distorted pixels can be pre-multiplied with the inverse of the camera matrix and distortion can be compensated for. Thus, the camera then works as a projective map in Euclidean space, $P : \mathbb{R}^3 \to \mathbb{R}^2$. The projection is defined as $P([X, Y, Z]^T) = [X/Z, Y/Z]^T$ and normalised camera measurement $y_t^m = [u_t, v_t]^T$ of a landmark, m, at time t is then

$$y_t^m = P(\mathsf{R}^{ce}(q_t)(m - p_t)) + e_t^m \tag{4}$$

which relates the absolute pose of the camera w.r.t., the 3D location of the point. The measurement noise is assumed i.i.d. Gaussian, $e_t^m \sim \mathcal{N}(0, R_m)$. The correspondence variables at time t, c_t^i, encodes the measurement-landmark assignment, $y_t^i \leftrightarrow m^j$, which gives a subset of all M landmarks at time t, $\mathsf{M}_t = \{m^j\}$, $j \in \{1, \ldots, M | \, c_t^i = j\}$. At time t the stacked measurement equation is then

$$\underbrace{\begin{bmatrix} u_t^1 \\ v_t^1 \\ \vdots \\ u_t^{\mathsf{N}_y} \\ v_t^{\mathsf{N}_y} \end{bmatrix}}_{y_t} = \underbrace{\begin{bmatrix} P(\mathsf{R}^{ce}(q_t)(m^{c_t^1} - p_t)) \\ \vdots \\ P(\mathsf{R}^{ce}(q_t)(m^{c_t^{\mathsf{N}_y}} - p_t)) \end{bmatrix}}_{h_t(x_t, \theta)} + e_t^m, \tag{5}$$

where c_t^i denotes the index of the corresponding landmark and N_y is the number of measurements, which of course varies over time. Methods for estimation of correspondence variables are discussed in Section 3.2.

3 SLAM Initialisation

In this section a method intended for initialisation of monocular visual/inertial SLAM from sequential data is described. The output of the method is a landmark map and the motion of the platform. It also establishes local correspondences via assignment variables using image features descriptors. Classical algorithms that solve assignment problems are the Hungarian (Munkres) algorithm Kuhn (1955) and the popular Auction algorithm Bertsekas (1991). Here a slightly different approach is adopted which results in a sequence of linear optimisation problems. In the landmark initialisation procedure we use appearance based correspondence matching, see e.g., Cummins and Newman (2010); Ho and Newman (2006). It aims at finding similar features corresponding to the same physical object in different images. Appearance based matching relies on feature descriptors that are distinctive and holds some invariance properties. For instance, image intensity invariance can be important in outdoor environments where lighting conditions may change and matching over large baselines requires invariance against scale, rotation and possibly invariance against change of viewpoint is desirable. In the following subsections we will describe the total initialisation procedure in detail and provide algorithms that implement these steps.

3.1 Feature Tracks

Feature tracks Thormählen et al. (2008) are established from the appearance of correspondences over multiple views by a matching scheme. Feature descriptor vectors, $\boldsymbol{f}$, from the popular Scale-Invariant Feature Transform (SIFT) Lowe (1999) are used for establishing correspondences. Given a sequence of images I_t, $t \in \{1, \ldots, K\}$, the feature matching problem consists of assigning a subset of feature measurements from image I_t, $\boldsymbol{f}_t^i$, $i \in \{1, \ldots, N_t\}$, to a subset of feature measurements from image I_{t+1}, $\boldsymbol{f}_{t+1}^j$, $j \in \{1, \ldots, M_{t+1}\}$, such that each measurement gets assigned to exactly one, unique, other measurement. In a manner similar to measurement-landmark assignment described before these assignments are also encoded by correspondence variables (which are binary in this case), $c_t^{ij} \in \{0, 1\}$, which are collected into c_t and the assignments for all images are collected into $\boldsymbol{c}$. Furthermore, each assignment is associated with a matching cost G_t^{ij} as

$$\mathsf{G}_t^{ij} = -\|\boldsymbol{f}_t^i - \boldsymbol{f}_{t+1}^j\|_2^{-1}, \tag{6}$$

which is the negative inverse Euclidean distance between the feature descriptor vectors. The costs are used to construct a matrix and to find pairwise matches in the image sequence. This is done by solving the assignment problem which can

be formulated as the following binary program (BP)

$$\begin{aligned} \widehat{c}_t = \arg\min_{c_t^{ij}} & \sum_{i=1}^{N_t} \sum_{j=1}^{M_{t+1}} \mathsf{G}_t^{ij} c_t^{ij} \\ \text{subject to} & \sum_{i=1}^{N_t} c_t^{ij} \leq 1, \forall j \\ & \sum_{j=1}^{M_{t+1}} c_t^{ij} \leq 1, \forall i \\ & c_t^{ij} \in \{0,1\} \end{aligned} \tag{7}$$

which is typically hard to solve. A standard method is to relax the binary constraints $c_t^{ij} \in \{0,1\}$ to $0 \leq c_t^{ij} \leq 1$. This relaxation gives that (7) becomes a linear program (LP) which is much easier to solve. A compact representation of the relaxed BP assignment problem (7) on matrix form is (omitting time index for readability)

$$\begin{aligned} \widehat{\bar{c}} = \arg\min_{\bar{c}} & \ \bar{\mathsf{G}}^T \bar{c} \\ \text{subject to } & A\bar{c} \leq \mathbf{1}_{(N+M)\times 1} \\ & \mathbf{0}_{NM\times 1} \leq \bar{c} \leq \mathbf{1}_{NM\times 1} \end{aligned} \tag{8}$$

where $\bar{\mathsf{G}}$ and $\bar{c}$ are the vectorised versions of the matrices G and c where columns are stacked on top of each other. The matrices $\mathbf{1}_{i\times j}$ and $\mathbf{0}_{i\times j}$ are the $i \times j$ matrix of ones and zeros respectively. Matrix A, which has dimension $(N+M) \times NM$, has a specific structure as follows: The first M rows look like

$$A_1 = I_M \otimes \mathbf{1}_{1\times N} \tag{9a}$$

and the last N rows look like

$$A_2 = \mathbf{1}_{1\times M} \otimes I_N \tag{9b}$$

and $A = [A_1^T \ A_2^T]^T$. $\otimes$ represents the Kronecker's matrix product. This constraint matrix A is totally unimodular, that is, all possible square sub-matrices are unimodular i.e., having determinant equal to ± 1. An important observation is that the matrix of (relaxed) constraints is unimodular. This means that the LP problem is integral, i.e., its optimum has an integer value corresponding to the optimum of the original BP problem Papadimitriou and Steiglitz (1982). This means the assignment problems are simple since good and fast LP solvers are readily available. In this work Gurobi Optimizer Gurobi Optimization Inc (2013) is used. The computational bottleneck for these problems is creating the cost matrix G since each element must be calculated.

The solution to the assignment problem will always use all the measurements from the smaller set no matter how bad the fit is. The reason is because the cost will always decrease by assigning one more variable, no matter how small the

decrease is. This is not a desired behavior since these matches can in principle be arbitrarily bad. It is therefore necessary to model features which are unique for each measurement such that they do not end up being assigned. One way of doing this is to add a regularisation term to the assignment cost as

$$\mathsf{G}_t^{ij} = -\|\boldsymbol{f}_t^i - \boldsymbol{f}_{t+1}^j\|_2^{-1} + \eta, \tag{10}$$

where $\eta > 0$ is a tuning parameter which controls the rejection of the excess assignments that are bad. Thus, η will force the cost of certain, unlikely, assignments to become positive, implying that those assignments will never be chosen since they would increase the total cost.

3.2 Track Clustering

Feature tracks are defined as a time sequence of pairwise matching feature correspondences $\mathsf{C}_{t:t+s} = [\boldsymbol{f}_t, \boldsymbol{f}_{t+1}, \ldots, \boldsymbol{f}_{t+s}]$. The minimum length of a track is then a pair because a feature without a match is not useful. The length of the tracks has a twofold interpretation; a feature descriptor is unique with respect to others in the sequence, i.e., the feature has a unique surrounding, and the other case is when the camera is stationary and thus the scene has been observed for a long time. However, in case of a moving camera, feature tracks may be lost due to e.g., occlusion or change of perspective. Therefore, new feature tracks may represent previously initiated tracks. To cater for this a track clustering scheme is employed joining tracks that may represent observations of the same feature. For simplicity of calculation, each track is represented by the mean value, $\bar{\mathsf{C}}$, of all the descriptors that constitute that track. The distance between tracks used for clustering is then

$$d_{ij} = \|\bar{\mathsf{C}}^i - \bar{\mathsf{C}}^j\|_2. \tag{11}$$

Since tracks have a temporal meaning and each feature can be measured only once in each image, valid clusters of tracks contain only time-disjoint tracks. That is, $\mathsf{C}_{i:j}$ and $\mathsf{C}_{k:l}$ are allowed to be clustered together only if $\{i : j\} \cap \{k : l\} = \emptyset$. Time-disjoint clustering is illustrated in Figure 3. Solving for these constraints can be done by simply removing time-overlapping tracks within clusters. Since the amount of landmarks is unknown, a clustering method where the number of clusters is not explicitly given is used. One such method is single-linkage clustering where the clustering stops when some condition on the between-cluster distance is fulfilled, Hastie et al. (2009). This distance is viewed as a tuning parameter. Another benefit of the single-linkage clustering is its speed, since there are good implementations available. Furthermore, data reduction and automatic loop closure detection is obtained since loops are defined by clusters containing more than one track. This implies that the between-cluster distance used for termination of the clustering controls the quality of the loop closures; if stopping too soon, there will be many small clusters and some loop closures will be missed and if stopping too late the risk of clustering wrong tracks together is increased. The track clustering algorithm is summarised in Algorithm 1. In this way, a set of landmarks has been obtained representing the initial map of the environment. Er-

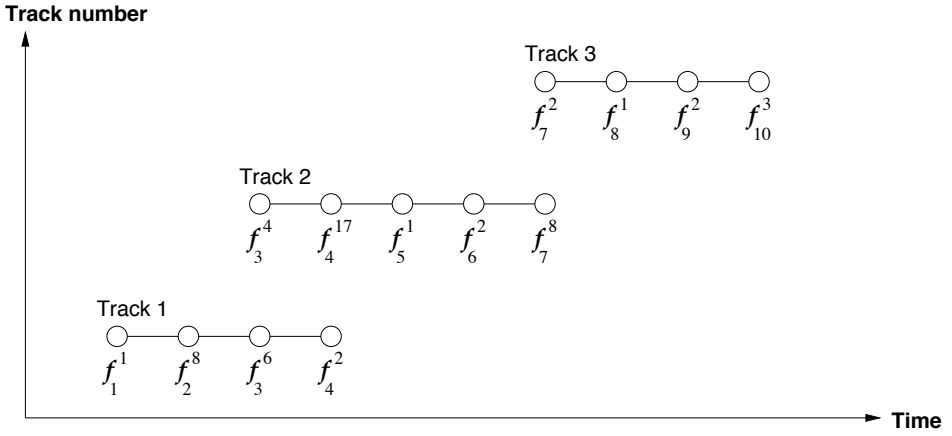


Figure 3: *Example of time-disjoint track clustering where only Track 1 and Track 3 are allowed to be clustered. Subscript figures on the descriptors, f, are the time indices and superscript figures are the enumeration of features at each time instant.*

Algorithm 1 Track Clustering

Input: The set of all tracks $\{\mathsf{C}^i\}_{i=1}^{N_\mathsf{C}}$, where $\mathsf{C}^i = [f_k, f_{k+1}, \ldots, f_{k+l}]$, $l - k > 1$.
Output: Track clusters $\mathbf{C}$

1: Compute track means:
   $\bar{\mathsf{C}}^i = \frac{1}{l-k} \sum_{t=k}^{l} f_t,\ i = 1, \ldots, N_\mathsf{C}$
2: Cluster the tracks:
   $\tilde{\mathbf{C}} = Cluster\_data(\bar{\mathsf{C}}^i),\ i = 1, \ldots, N_\mathsf{C}$
3: **for** all clusters **do**
4: **if** tracks within the cluster are time disjoint **then**
5: keep the cluster
6: **end if**
7: **end for**
8: Remaining clusters are $\mathbf{C}$

rors introduced, outlier measurements, in the clustering and in the feature tracks should be removed. This will be done according to Algorithm 3.

3.3 Rotation Initialisation

From sets of correspondences, estimated as in Section 3.1, for each image pair in the image sequence the relative transformations (up to a scale of the translation) can be obtained. This is done with the Eight-Point Algorithm, see e.g., Hartley and Zisserman (2004), resulting in a rotation sequence

$$\mathbf{R}^c = \{\mathrm{R}_i^c\}_{i=1}^{K-1}, \tag{12}$$

where R_i^c is the relative rotation of the cameras, (c), from time i to $i+1$. The global rotation, from world to camera, (ce), for any time, j, can be calculated by the matrix product

$$\mathrm{R}_j^{ce} = \prod_{i=j}^{0} \mathrm{R}_i^c \,. \tag{13}$$

Note that rotation matrices do not generally commute meaning that the product must be done in the reverse time order. Also note that the rotations are initialised with camera only and the gyro is not used. This is because the dominating error from the gyro is bias giving drifting rotation estimates. Errors from camera estimated rotations have more random like behaviour resulting in the random walk errors. There may also be correspondence errors in the feature tracks resulting in bad rotations. The sensitivity to initial rotation errors are further analysed in Section 6.2.

3.4 Linear SLAM

Methods of 3D structure estimation using linear methods and image point correspondences are well known. The basic idea is to form an overdetermined triangulation problem, which is linear in the unknowns, and solve it by linear least squares. This is essentially the Direct Linear Transformation (DLT) Abdel-Aziz and Karara (1971) which may work well in practice. However, instead of minimising the discrepancy between measured image coordinates and the back projection of points, an algebraic error without good geometrical interpretation is minimised Hartley and Zisserman (2004). It is therefore common to proceed with a nonlinear optimisation over the reprojection residuals with the linear method as a starting point. Given the correct weighting of the measurements, then the linear method minimises the reprojection error, see for instance Zhang and Kanade (1998). This weighting does however depend on the unknown depth of the points but iterative re-weighting often improves the linear solution.

The projection is a convex operation and this can be exploited in many ways. Optimal approaches to reconstruction consider reprojection errors under the L_∞ norm Hartley and Schaffalitzky (2004); Kahl and Henrion (2007) since it preserves (quasi)-convexity, see Olsson and Kahl (2010); Kahl (2005) and have a single optimum which is typically not the case for the L_2 cost. These approaches

assume outlier free measurements because otherwise the maximum error will correspond to an outlier and by using this insight an iterative method for outlier removal using L_∞ was proposed Sim and Hartley (2006), however, it does not scale to large problems Agarwal et al. (2008). In common, it is assumed that rotations are known beforehand and in many situations this is a reasonable alternative since rotations may be estimated from point correspondences. Known (or error-free) orientation was considered for fusion of vision and inertial sensors in Martinelli (2012) and for the case of visual odometry with inertial measurements in Kneip et al. (2011a,b). The assumption of known orientation will also be used here which results in an almost linear method. Assuming known rotations it is possible to rewrite (11) in the following way (omitting time index)

$$\delta = \mathsf{R}^{ce}(q)(m-p) \tag{14}$$

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \frac{\delta_x}{\delta_z} \\ \frac{\delta_y}{\delta_z} \end{bmatrix} + \begin{bmatrix} e_u \\ e_v \end{bmatrix} \Rightarrow \tag{15}$$

$$\begin{bmatrix} u\delta_z \\ v\delta_z \end{bmatrix} = \begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix} + \begin{bmatrix} e_u\delta_z \\ e_v\delta_z \end{bmatrix} \tag{16}$$

where $\delta(p, m) = [\delta_x,\ \delta_y,\ \delta_z]^T$ is the difference between landmark and camera positions expressed in the camera coordinate system. Equation (10) is linear in the unknown parameters m and p, but has noise that is dependent on the parameters. With δ explicit (10) becomes

$$\mathsf{R}_{3,:}(m-p)\begin{bmatrix} u-e_u \\ v-e_v \end{bmatrix} = \begin{bmatrix} \mathsf{R}_{1,:}(m-p) \\ \mathsf{R}_{2,:}(m-p) \end{bmatrix}, \tag{17}$$

where $\mathsf{R}_{i,:}$ denotes the i:th row of the rotation matrix R^{ce}. The accelerometer measurements with bias are as in Section 4.1

$$y_t^a = \mathsf{R}_t^{ce}(a_t - g^e) + b_a + e_t^a \tag{18}$$

where g^e, b_a are assumed constant and $e_t^a \sim \mathcal{N}(0, R_a)$. Usually, the sampling rate of an IMU is faster than a camera which can be handled in a straightforward fashion e.g., by averaging accelerations between the camera samples

$$y_t^a = \frac{1}{(S_t - S_{t-1})} \sum_{s=S_{t-1}}^{S_t} y_s^a, \tag{19}$$

where S_t maps the time indices between the camera and IMU. Treating the accelerations, its bias, initial velocity and landmarks as unknown parameters and

Algorithm 2 Iterative Reweighted Least Squares

Input: IMU measurements, $y^a_{1:N}, (\omega_{1:N})$, feature measurements, $y^m_{1:N}$, rotations $\mathsf{R}^{ec}_{1:N}$, data associations $\boldsymbol{c}$, initial parameters $\theta_0 = [a_{1:N}, v_0, b_a, m]^T$, number of iterations K

Output: Parameter estimates θ_{init}

1: **for** $k := 1 \dots K$ **do**
2: Create the WLS problem according to (21) with $\tilde{R}_m = \delta_z(\theta_{k-1})^2 R_m$
3: Solve the WLS problem giving θ_k
4: **end for**
5: $\theta_{\text{init}} := \theta_k$

defining $\theta = [a_{1:N}, v_0, b_a, m]^T$, the following formulation is proposed

$$\begin{aligned}\theta_{\text{init}} = \arg\min_{\theta} \sum_{t=1}^{N} &\|y^a_t - \mathsf{R}^{ce}_t(a_t - g^e) - b_a\|^2_{R_a^{-1}} + \\ &\|y^m_t \delta_z(p_t, m) - \delta_{x,y}(p_t, m)\|^2_{\tilde{R}_m^{-1}} \\ \text{subject to } &\begin{bmatrix} p_t \\ v_t \end{bmatrix} = F^t \begin{bmatrix} p_0 \\ v_0 \end{bmatrix} + \sum_{i=1}^{t} F^{i-1} B a_i, \\ &F = \begin{bmatrix} I_3 & TI_3 \\ 0 & I_3 \end{bmatrix},\ B = \begin{bmatrix} \frac{T^2}{2} I_3 \\ TI_3 \end{bmatrix},\end{aligned} \tag{20}$$

where $\tilde{R}_m = \delta_z(p^0_t, m^0)^2 R_m$, and superscript 0 indicates that the parameters used for weighting are fixed for each iteration. The constraints represent the second order linear dynamics introduced in (9). In turn, this can also be interpreted as a second order interpolation of the trajectory. The formulation in (20) would be a constrained weighted linear least squares (WLS) problem if the measurement noise in the camera did not depend on the landmark. Now since any p_t can be expressed as a (linear) function of v_0 and $a_{1:t}$ the constraint can be directly substituted into the cost function resulting in the following unconstrained problem

$$\begin{aligned}\theta_{\text{init}} = \arg\min_{\theta} \sum_{t=1}^{N} &\|y^a_t - \mathsf{R}^{ce}_t(a_t - g^e) - b_a\|^2_{R_a^{-1}} + \\ &\|y^m_t \delta_z(v_0, a_{1:t}, m) - \delta_{x,y}(v_0, a_{1:t}, m)\|^2_{\tilde{R}_m^{-1}},\end{aligned} \tag{21}$$

where $\tilde{R}_m = \delta_z(v_0, a_{1:t}, m)^2 R_m$. The only difference between this problem and usual WLS is the parameter dependent noise for the landmark measurements. This can be treated in an iterative fashion where δ_z is used for weighting the noise covariance which is evaluated using the parameter values from the previous iteration. This approach is also known as Iterative Reweighted Least Squares (IRLS) which is a well known method, see e.g., Björck (1996). The procedure is described in Algorithm 2. This approach usually converges after a few iterations and in our implementation three ($K = 3$) iterations were a suitable choice.

Algorithm 3 Iterative Outlier Rejection

Input: IMU measurements, $y^a_{1:N}$, feature measurements, $y^m_{1:N}$, rotations $\mathsf{R}^{ec}_{1:N}$, data associations $\boldsymbol{c}$, initial parameters $a_{1:N}$, v_0, b_a, m, rejection threshold λ
Output: Data associations $\boldsymbol{c}$

terminate := **false**
$k := 0$
$\boldsymbol{c}^k := \boldsymbol{c}$
while not *terminate* **do**
  Solve the problem according to Algorithm 2 given the assignments $\boldsymbol{c}^k$ and all the parameters and produce a set of landmark residuals for each image i, ε^i_m
  for each image i **do**
    $\bar{\varepsilon} := \varepsilon^i_m \setminus \max(\varepsilon^i_m)$
    if all $\max(\varepsilon^i_m) \oslash \bar{\varepsilon} > \lambda$ **then**
      remove the assignment that is associated with $\max(\varepsilon^i_m)$ from $\boldsymbol{c}^k$
    end if
  end for
  if no assignments removed from $\boldsymbol{c}^k$ **then**
    terminate := **true**
  else
    $k := k + 1$
  end if
end while
$\boldsymbol{c} := \boldsymbol{c}^k$

The problem (21) can be augmented with linear terms for initial gyro bias estimation, $\|q_t - \frac{T}{2} S_\omega(\omega_t + b_\omega) q_{t-1}\|^2_{R_q^{-1}}$, which is the first order approximation of (9c). Using the bilinear relation $S_\omega(b)q = \tilde{S}_q(q)b$, this can be written as $\|q_t - \frac{T}{2} S_\omega(\omega_t) q_{t-1} - \frac{T}{2} \tilde{S}_q(q_{t-1}) b_\omega\|^2_{R_q^{-1}}$, which is also a linear function of b_ω, since rotations are assumed to be known. These terms are decoupled from the rest of the parameters and the WLS problem defined by these can be solved separately if required.

3.5 Iterative Outlier Removal

The landmark initialisation produced by Algorithm 1 will introduce erroneous associations due to the unavoidable ambiguity of the feature descriptors. These associations should be considered outliers. It is difficult to discriminate outliers based on descriptors alone. However, given the IMU data, which describe the motion independently of cluster appearance, a strategy for inertial based outlier rejection can be devised according to the pseudo-code in Algorithm 3. This procedure will terminate when all of the residuals are of similar size, where similar is defined here by the rejection threshold λ. The operator $\oslash$ denotes element-wise division.

4 Nonlinear Least-Squares SLAM

Accelerations, initial velocity, landmarks and biases estimated with the linear method described in Section 3.4 are used as an initial value for NLS. Here the reprojection error is formulated in its original form, that is

$$y_t^m = P(\mathrm{R}^{ce}(q_t)(m - p_t)) + e_t \tag{22}$$

where the operator P is defined as in (11). The second addition is that rotations are not fixed any more, but are estimated together with the rest of the parameters. This is done by adding the angular velocities $\omega_{1:N}$ to the parameter set. Now, the measurement relation from Section 4.1 can be used

$$y_t^\omega = \omega_t + b_\omega + e_t^\omega. \tag{23}$$

and the rotations can be calculated by using the relation (9c) as

$$q_t = \left[\prod_{k=1}^{t} \exp\left(\frac{T}{2} S_\omega(\omega_k)\right)\right] q_0, \tag{24}$$

where q_0 is assumed given. These two modifications give the new parameter vector $\theta = [a_{1:N}, \omega_{1:N}, v_0, b_a, b_\omega, m]^T$ and the new nonlinear least squares (NLS) formulation of the problem according to

$$\begin{aligned}\hat{\theta} = \arg\min_\theta \sum_{t=1}^{N} & \|y_t^a - \mathrm{R}_t^{ce}(a_t - g^e) - b_a\|^2_{R_a^{-1}} + \\ & \|y_t^\omega - \omega_t - b_\omega\|^2_{R_\omega^{-1}} + \\ & \left\| y_t^m - \frac{\delta_{x,y}(v_0, a_{1:t}, \omega_{1:t}, m)}{\delta_z(v_0, a_{1:t}, \omega_{1:t}, m)} \right\|^2_{R_m^{-1}}\end{aligned} \tag{25}$$

and R_t^{ce} is now a function of $\omega_{1:t}$ and δ is defined in (14). This problem can be solved efficiently with e.g., a standard Levenberg-Marquardt solver, Nocedal and Wright (2006).

5 Heuristic Motivation of the Linear Initialisation

In tracking and navigation the measurement models are often nonlinear and so are most SLAM systems stemming from e.g., transformations between reference frames by rotations, perspective divide, among others. In practice this means that there exist local minima which should be avoided. In order to reach the global minimum the initialisation point should be in the proximity of the global minimum or at least the function should be monotone between the initial point and the global minimum and even better is of course if it is also convex along this direction. In fact, it is sufficient that the function is convex on a path that the minimisation procedure will take in order to end up in the global minimum. Here we propose an initialisation procedure based on the almost linear method. We will use a simple heuristics to motivate that the initial point created in this

manner is better than an initial point created using available measurements.

The definition of convex function $f(\theta): \mathbb{R}^n \to \mathbb{R}$ is

$$f(\lambda\theta_1 + (1-\lambda)\theta_2) \leq \lambda f(\theta_1) + (1-\lambda) f(\theta_2), \tag{26a}$$

$$0 \leq \lambda \leq 1, \tag{26b}$$

$$\forall \theta_1, \theta_2 \in \mathbf{dom}\ f \subset \mathbb{R}^n. \tag{26c}$$

Geometrically, this is interpreted as the hyperplane that lies between points $(\theta_1, f(\theta_1))$ and $(\theta_2, f(\theta_2))$ is always above the function f. If this is fulfilled for all θ then the function is convex. Convex functions have a property that there is only one global minimum, so any minimisation procedure can be used to obtain that. For example linear least squares problems fall into this category. However, many functions, although non-convex on the whole domain, are convex on a subset of the domain, usually in the proximity of the local minima. This can be motivated with the fact that Taylor expansion around the local minimum will be quadratic function plus a rest term of a higher degree. If this rest term is not dominating over the quadratic term, the function is (locally) convex in this region. As stated before, in order to apply a local minimisation procedure, and successfully obtain the global minimum, the path between the initial point and minimum should also fulfill the convexity property (26).

The main idea is to check the local convexity of the NLS cost function given the initial point produced with the linear initialisation procedure. This can be done approximately by using a dense sampling of the cost function along the search direction as given by the initialisation and then evaluate if the path is convex according to (26). In addition, we also require that the search direction in the initial point, p_0, is well aligned with the direction from the initial point to the true solution, $\theta^* - \theta_0$. The intuition behind this is that an initial search direction is crucial for convergence to a good solution. This is determined with the angle between these two directions defined as

$$\gamma = \arccos\left(\frac{p_0^T(\theta^* - \theta_0)}{\|p_0\|_2 \, \|\theta^* - \theta_0\|_2}\right), \tag{27a}$$

$$p_0 = -(J_0^T J_0)^{-1} J_0^T \varepsilon_0, \tag{27b}$$

where J_0 is a Jacobian matrix of the cost function evaluated in θ_0 and ε_0 is the residual in θ_0. These criteria will be evaluated in a Monte Carlo fashion for the linear initialisation procedure and compared to other initialisation approaches, for example using measurements only.

6 Monte Carlo Simulations

Monte Carlo (MC) simulations are used to evaluate the whole initialisation approach. That is, to find out if the linear method gives a good starting point for the nonlinear optimisation and if the outlier rejection procedure seems reasonable. For the initialisation of the landmarks, i.e., the clustering approach, only

Initialisation method	Linear	Naïve
Average initial angle γ [°]	17.7	66.6
Average error $\|\hat{\theta} - \theta^*\|/\mathbf{dim}(\theta^*)$	$1.6 \cdot 10^{-3}$	$221 \cdot 10^{-3}$
# of non convex paths	0	0

Table 1: *MC simulation results for the initialisation method, see Section 6.1, where the measurement noise was varying. The angle γ is defined in* (27a)*, $\hat{\theta}$ is defined in* (25) *and θ^* is the true solution.*

real data are used since it is complicated to make simulated SIFT features and these results are presented in Section 8.2.

6.1 Efficiency of the Linear Initialisation

Here, the method according to Section 3.4 is evaluated. In the first set of simulations a trajectory and a set of landmarks are created and different white Gaussian noise is simulated and added to the measurements for each MC simulation. No outliers are present. Initial rotation used for the linear optimisation is also randomly perturbed but fixed together with the trajectory and the scene. The linear solution produced in this way is compared with the naïve guess where the initial point is created from the measurements and randomised landmark positions.

The first set of MC simulations consist of 50 simulations performed on a data set with 20 landmarks and 30 time points. Landmarks were placed in a general configuration, with varying distance to the camera. The noise standard deviation was fixed and is set to $\sigma_a = 10^{-3}$ m/s^2, $\sigma_m = 10^{-4}$ m and $\sigma_\omega = .5°$/s. The results are listed in Table 1. Both the proposed initialisation method and the naïve one have convex paths between the initial and the true point, but the initial angle between the search direction and the direction to the true solution is much larger (approximately four times) for the measurement initialised optimisation. This causes the average error of the solution to be much larger (approximately 140 times) for the naïve initialisation.

6.2 Sensitivity to Initial Rotation Errors

In the second set, consisting of 3 times 25 realisations, the trajectory and the scene are fixed as before, no noise is added to the measurements and no outliers are present. Here the initial rotations are varied randomly with different perturbation magnitude 0.1, 1 and 5 [°/s]. The solutions to the linear initialisations given different rotation perturbations are then used to solve the non-linear optimisation problem and the results are in Table 2. Here it can be seen that both error in the linear solution as well as initial angle grow with the magnitude of the error in rotation. As a consequence the number of non-convex paths also increases with the perturbation magnitude and the average error of the estimate is large. The reason for the very large average for the perturbation of 1 degree per second is the presence of 4 really bad solutions. With these removed the value is 0.45 which is more reasonable. This evaluation shows the importance of the

Perturbation magnitude [°/s]	0.1	1	5
Average initial angle γ [°]	1.64	40.2	84.0
Average error $\|\hat{\theta} - \theta^*\|/\mathbf{dim}(\theta^*)$	$8.34 \cdot 10^{-9}$	69.1	1.0
Average error $\|\theta_{\text{init}} - \theta^*\|/\mathbf{dim}(\theta^*)$	0.08	0.63	0.78
# of non convex paths	0	3	18

Table 2: *MC simulation results for the initialisation method, see Section 6.2, where the initial rotation error was varying. The angle γ is defined in* (27a)*, the NLS estimate $\hat{\theta}$ is defined in* (25)*, the initial estimate θ_{init} is defined in* (21) *and θ^* is the true solution.*

initial estimation of the rotations.

6.3 Iterative Outlier Removal

The third set is used to evaluate the performance of the outlier rejection method proposed in Algorithm 3. Here the outlier measurement rate is varied while the trajectory, scene and the initial rotations are fixed. The performance is evaluated according to the amount of outlier measurements that are left and the amount of inlier measurements that are removed.

Results are presented in Table 3, in this case the scene is fixed to 30 landmarks in 21 images and the outlier rate is varied, both as a number of landmark outliers and as a number of images where outliers are present. The outliers are created by randomly flipping a pair of associations in an image. For example, if the number of landmarks to be outliers are two, then in all images where outliers are present the measurements from these two landmarks are flipped with two other randomly chosen landmarks. This strategy is chosen because it is the behaviour of the data association method employed here. Three different rejection thresholds are used, $\lambda = \{3, 5, 8\}$, in the experiments. The result shows that, as a general trend and as expected, the rejection threshold governs the amount of outliers that are left and number of inliers that are rejected. It is of general interest to reject as many outliers as possible and to keep as many inliers as possible. The lower threshold means better outlier rejection, but the price is that more inliers are also rejected. This also emphasises the importance of having many landmark measurements in order to be resilient to removing inliers. Note that this statistics is conservative in the way that only measurements that are created as outliers are considered as true outliers. In many cases all measurements for a landmark that is creating outliers are removed, implying that this landmark is no longer deemed as usable and it can be excluded from the statistics. This means that results in Table 3 would be somewhat better if these are taken into account.

$\lambda = 3$	Outliers left [%]			Inliers removed [%]		
IO \ LO	0.07	0.13	0.2	0.07	0.13	0.2
0.05	0	0	0	13	22	22
0.10	0	0	0	18	23	31
0.14	0	0	0	17	26	41
0.19	0	6.3	0	29	36	40
0.24	0	0	0	20	20	52
0.29	0	0	0	20	46	48
0.33	0	0	2.4	23	44	52
0.38	6.3	0	2.1	42	50	54
$\lambda = 5$	Outliers left [%]			Inliers removed [%]		
IO \ LO	0.07	0.13	0.2	0.07	0.13	0.2
0.05	0	50	0	9.1	12	14
0.10	0	0	0	11	21	32
0.14	0	8.3	5.6	17	25	42
0.19	0	0	0	21	34	38
0.24	0	5	0	19	25	54
0.29	0	0	0	16	36	49
0.33	7	0	0	21	34	51
0.38	6.3	0	0	39	35	54
$\lambda = 8$	Outliers left [%]			Inliers removed [%]		
IO \ LO	0.07	0.13	0.2	0.07	0.13	0.2
0.05	0	0	8.3	8.1	9.2	15
0.10	0	0	0	17	20	27
0.14	0	0	11	16	18	26
0.19	0	0	0	15	25	35
0.24	0	5	10	22	34	58
0.29	8.3	0	0	16	25	42
0.33	0	0	2.4	27	42	38
0.38	6.3	0	8.3	34	31	66

Table 3: *Outlier rejection simulation results. IO is the fraction of images in which outliers are present. LO is the fraction of the landmark set in which outliers are present. Three different rejection thresholds λ are used.*

Figure 4: *Sensor unit containing both camera and inertial sensors.*

7 Real Data Experiments

In the real data experiments, a sensor unit, see Figure 3, equipped with monocular monochrome VGA camera (Pointgrey firefly) and three axis inertial sensors (gyroscopes and accelerometers) was used. It contains also magnetometers and a temperature sensor which is used for internal calibration. In one of the experiments the sensor unit was mounted at the tool position of an IRB-1400 industrial robot from ABB for the purpose of an accurately known ground truth. Also, a small scene with objects of known size was created so that the estimated scene could be compared with respect to its size. In the second experiment, a free-hand movement of the camera is used in a room, and no accurate ground truth is available. In this case the accuracy is evaluated based on the approximate measurements in the room and approximately "knowing" where camera was. Prior to usage, the camera was calibrated using the toolbox Bouguet (2010) and the relative pose of the camera centre with respect to the IMU centre was calibrated as described in Hol et al. (2010). A open source SIFT implementation from Vedaldi and Fulkerson (2008) was used.

In Figure 4 the trajectory estimate from the linear initialisation and the nonlinear refinement are plotted together with the ground truth trajectory for the first data set. An image with the feature measurements is illustrated in Figure 5. Velocity in x, y-plane is shown in Figure 5. Since the true rotations are known, errors in quaternions and Euler angles are depicted in Figures 6 and 9 respectively. Landmark estimates for initial and nonlinear estimation are shown in Figure 10.

For the second data set (free-hand run), an example image together with plotted features is showed in Figure 11. The resulting trajectory estimate is shown in Figure 12. The x, y-plane velocity, quaternions, Euler angles and landmark estimates

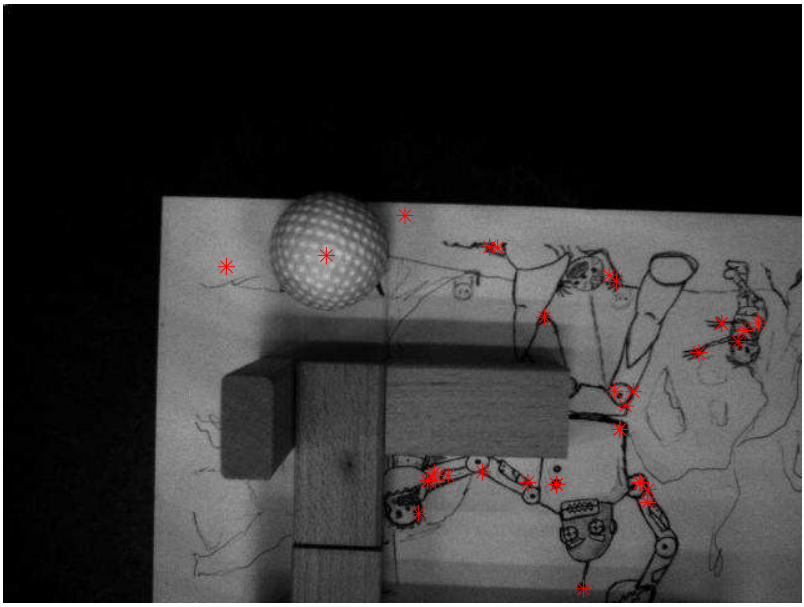


Figure 5: *Example image from the robot run with extracted features shown as red stars.*

are plotted in Figures 13, 14, 15 and 16 respectively.

In both cases the nonlinear refinement improves the initial estimate for both rotations and kinematic parameters. In the first data set the landmark estimate is much better than the initial estimate, and for the second data set this is harder to assess, but the positions of the landmarks look reasonable given the environment. For example we see that there are three levels in the bookshelf with distinct features, which can be seen in both Figure 11 and Figure 16.

7.1 Clustering Results

For the landmark initialisation approach (track clustering) a helicopter data set made with a Yamaha Rmax helicopter is used. In this data set only the images were available, i.e., no IMU, implying that no outlier rejection nor complete SLAM could be done and this data set is only used to evaluate the clustering performance. The flight is performed in a circle, meaning that the helicopter visits the same place. The total length of one such loop was 325 images taken in 4 Hz giving the total time of 81.25 s. An example of the loop-closure is illustrated in Figure 17 where two tracks, one from the beginning and one from the end of the loop, are clustered together. With this cluster tuning the amount of loop-closures is about 4%. This should be compared to the amount of images showing the overlapping environment which were 7% of the total amount of images. For the other two data sets this number is higher, especially for the free-hand run

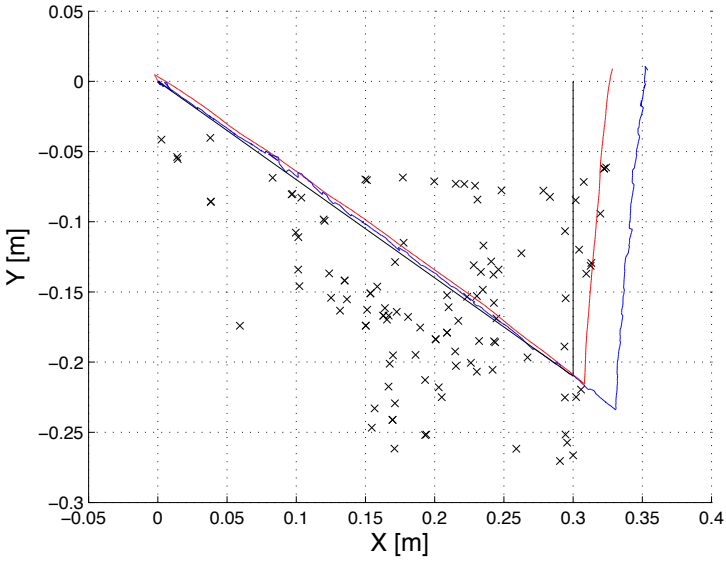


Figure 6: *Trajectory from the linear estimation (green), from nonlinear refinement (red) and ground truth (blue).*

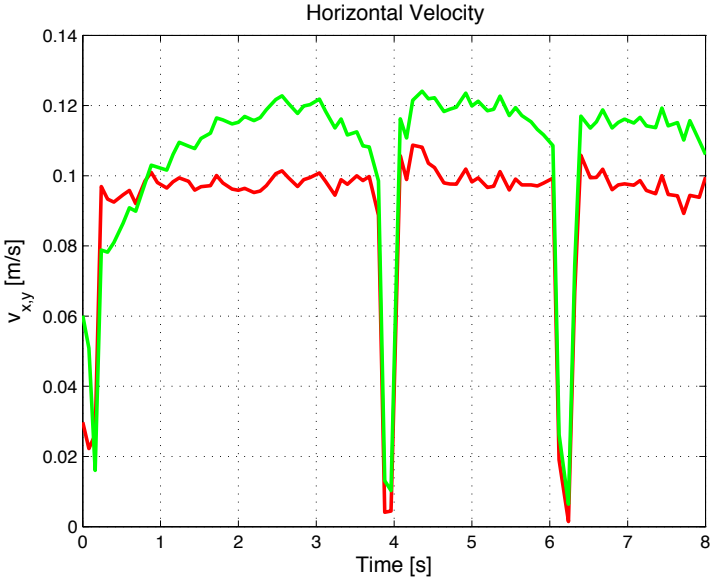


Figure 7: *Estimated horizontal velocity from the linear estimation (green) and from the nonlinear refinement (red). True velocity is* 0.1 m/s *except in three time points when it is zero.*

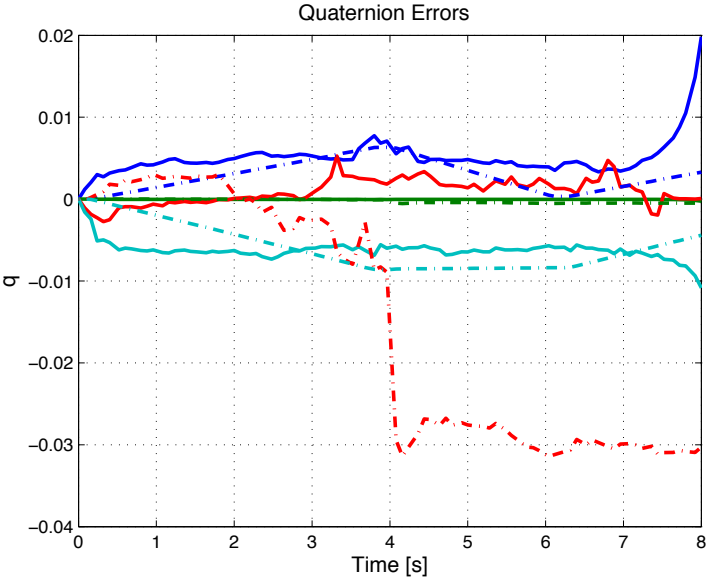


Figure 8: *Estimated quaternion error from camera (dash-dotted) and after nonlinear refinement (solid).*

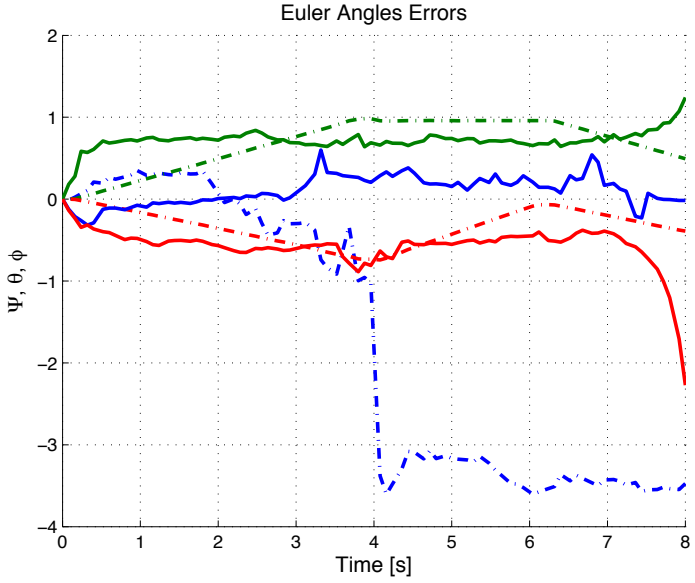


Figure 9: *Estimated Euler angles error in degrees from camera (dash-dotted) and after nonlinear refinement (solid). Blue is the yaw, green is the pitch and red is the roll angle.*

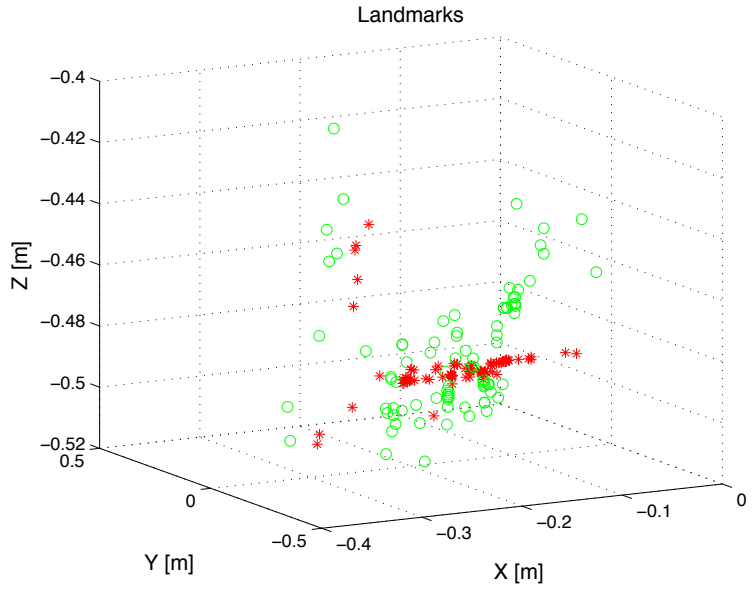


Figure 10: *Landmark estimates from the linear estimation (green) and from nonlinear refinement (red). Most of the landmarks should lie on the* −0.5 m *plane and some should be higher up.*

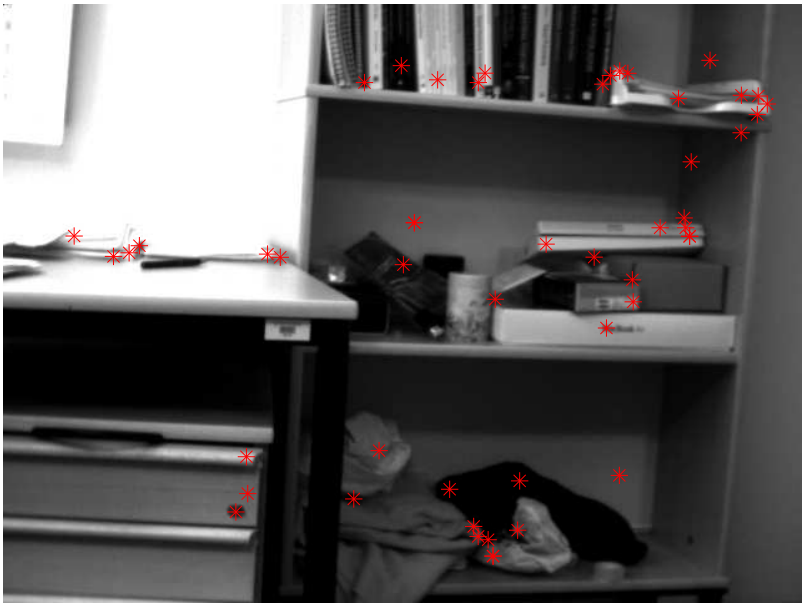


Figure 11: *Example image from the free-hand run with extracted features shown as red stars.*

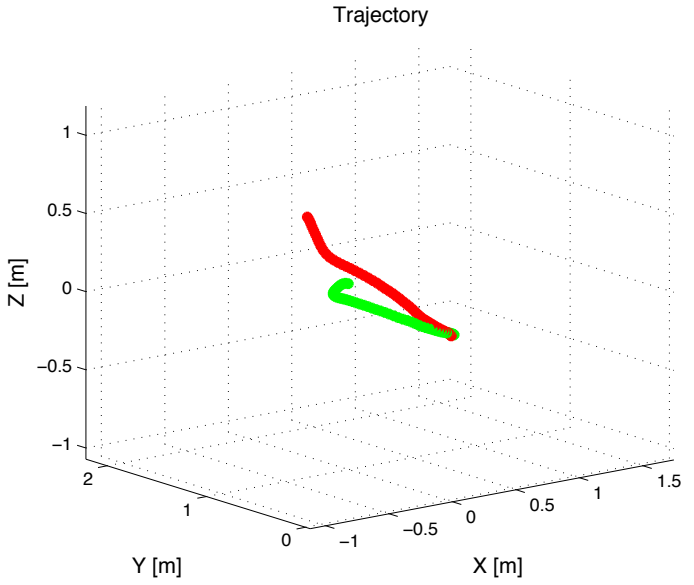


Figure 12: *Trajectory from the linear estimation (green), from nonlinear refinement (red).*

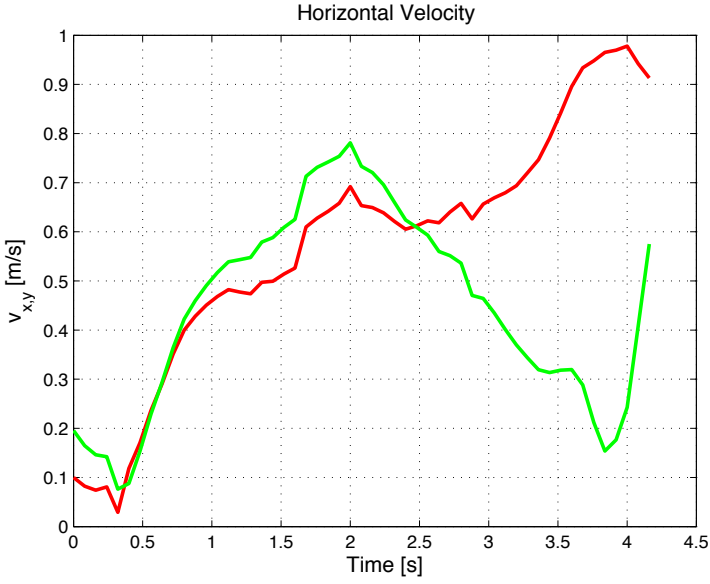


Figure 13: *Estimated horizontal velocity from the linear estimation (green) and from the nonlinear refinement (red).*

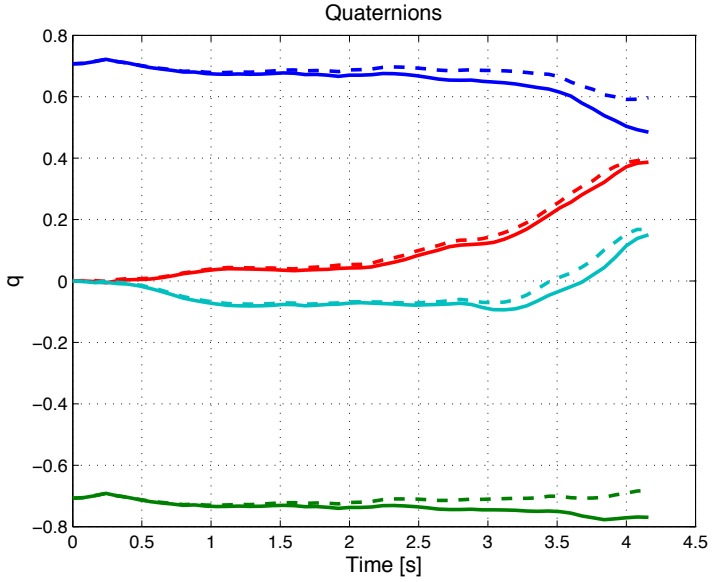


Figure 14: *Estimated rotations from camera (dashed) and after nonlinear refinement (solid).*

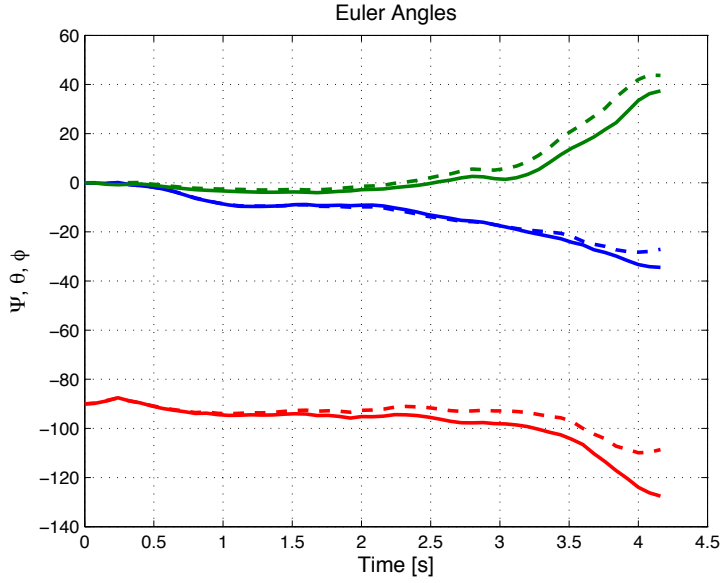


Figure 15: *Estimated Euler angles in degrees from camera (dash-dotted) and after nonlinear refinement (solid). Blue is the yaw, green is the pitch and red is the roll angle.*

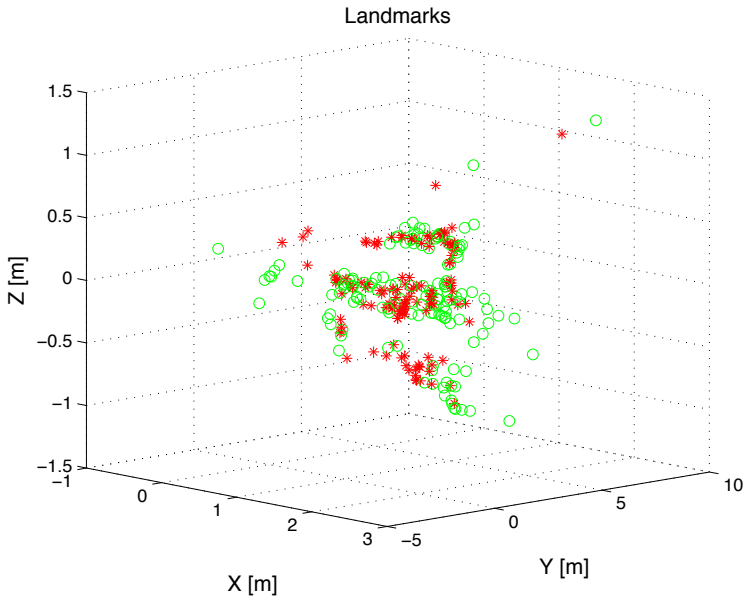


Figure 16: *Landmark estimates from the linear estimation (green) and from nonlinear refinement (red). Three distinct layers can be recognised which basically come from the bookshelf's levels.*

data set, where large parts of the environment were visible the whole time. In general, most of the clusterings are local loops where for example a feature is lost for a few images and is then tracked again. One such example can be seen in Figure 18.

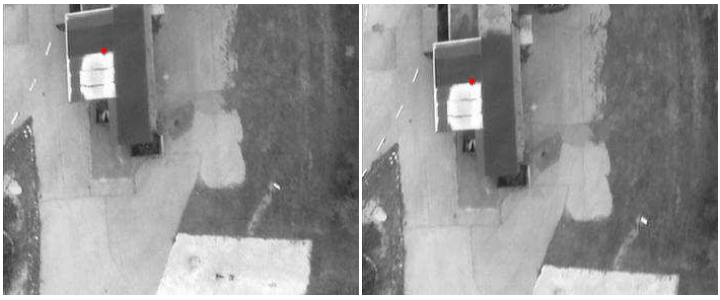
Even some outliers are introduced in the clustering process, which is, as explained earlier, expected, and one such is depicted in Figure 19. The total amount of outlier landmark was about 10%, which is similar for the other two data sets. Many of the outliers are caused by the too similar environment, for the example in Figure 19 the road edge looks similar to SIFT and the descriptors are too similar. This will cause the erroneous clustering.

8 Conclusions and Future Work

In this work we presented a method for initialisation of optimisation based visual/inertial SLAM on batch form. This sensor combination makes it possible to obtain full Euclidian reconstruction of the environment and trajectory. The method is based on a multistage strategy where visual methods, such as the Eight-Point Algorithm, feature extraction and clustering of feature tracks, are used for rotation and landmark initialisation. Inertial data are used for data association including outlier rejection and initialisation of trajectory and landmark location pa-

(a) *Image 7.* ***(b)*** *Image 8.*

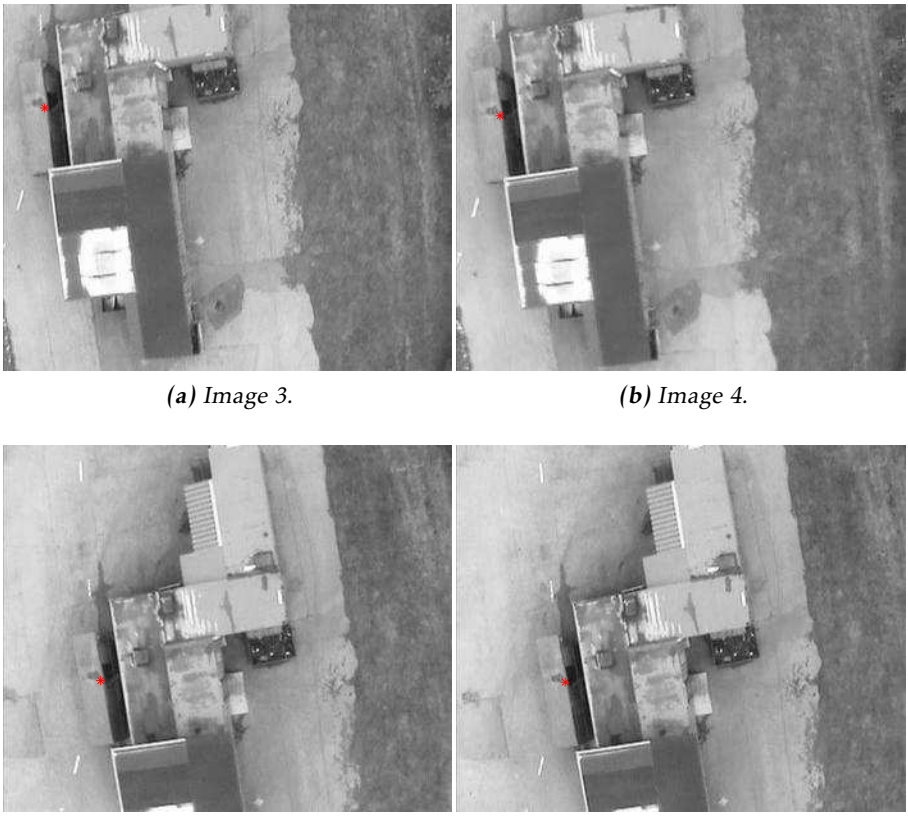
(c) *Image 9.* ***(d)*** *Image 321.*



(e) *Image 322.* ***(f)*** *Image 323.*

(g) *Image 324.* ***(h)*** *Image 325.*

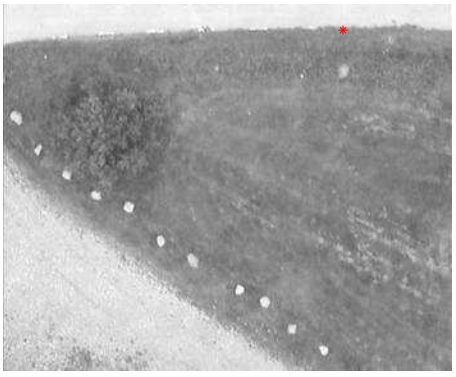
Figure 17: *Example of successful loop-closure clustering with the helicopter data. The tracked feature is marked with the red star.*



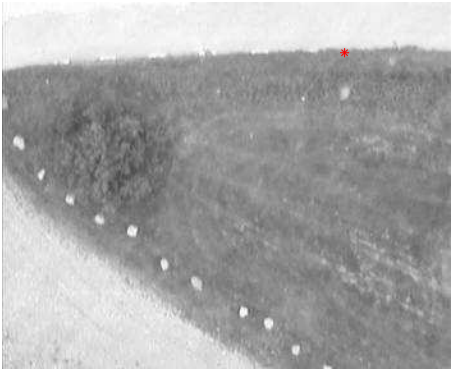
(a) *Image 3.* ***(b)*** *Image 4.*

(c) *Image 12.* ***(d)*** *Image 13.*

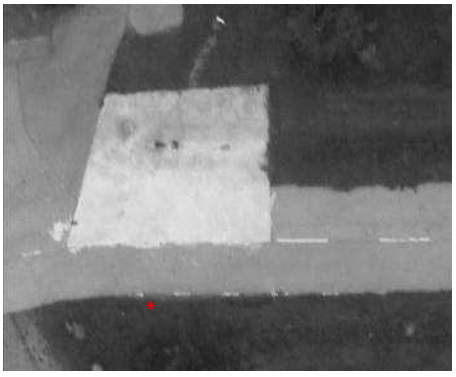
Figure 18: *Example of clustering creating local loops with the helicopter data. The tracked feature is marked with the red star.*



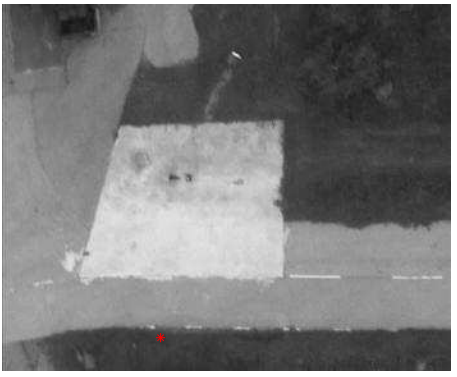
(a) *Image 299.*



(b) *Image 300.*



(c) *Image 314.*



(d) *Image 315.*

Figure 19: *Example of erroneous clustering causing an outlier with the helicopter data. The tracked feature is marked with the red star.*

rameters. The method exploits the conditional linearity of visual/inertial SLAM. The experiments done on the simulated and real data sets show that the initialisation method gives better starting point for the subsequent full nonlinear optimisation than naïve initialisation with measurements only.

Also, the landmark initialisation method based on clustering of the tracked features gives quite promising results where many possible loop-closures are identified while the amount of wrong associations is rather low. This allows for the iterative outlier rejection method with aid from the inertial data. Even this method shows good results with efficient outlier removal while keeping the inlier amount relatively high.

It must be pointed out that this approach requires a large amount of landmark measurements in order to produce good results, i.e., the equation system must be highly overdetermined. On top of that, since a camera is a bearings-only sensor, there is also a demand for sufficient viewpoint change, also known as parallax, in order to accurately estimate landmark position. The trajectory estimation can be compensated with the inertial data, but even there a good SNR is required.

In the future it can be interesting to use proper constrained clustering algorithm instead of discarding clusters with overlapping times, as it is done now. Also alternative feature detectors might be used to see if descriptors from those have different behaviour compared to SIFT.

Bibliography

Y. Abdel-Aziz and H. Karara. Direct Linear Transformation from Comparator to Object Space Coordinates in Close-Range Photogrammetry. In *Proceedings of the American Society of Photogrammetry Symposium on Close-Range Photogrammetry*, pages 1–18, Falls Church, VA, USA, 1971.

S. Agarwal, N. Snavely, and S. M. Seitz. Fast algorithms for L_∞ problems in multiview geometry. In *Computer Vision and Pattern Recognition*, pages 1–8, 2008. doi: 10.1109/CVPR.2008.4587713.

S. Agarwal, N. Snavely, I. Simon, S. Seitz, and R. Szeliski. Building Rome in a Day. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 72 –79, oct 2009. doi: 10.1109/ICCV.2009.5459148.

T. Bailey and H. Durrant-Whyte. Simultaneous Localization and Mapping (SLAM): Part II. *IEEE Robotics & Automation Magazine*, 13(3):108–117, Sept. 2006.

D. P. Bertsekas. An Auction Algorithm for Shortest Paths. *SIAM Journal on Optimization*, 1(4):425–447, 1991. ISSN 10526234. doi: DOI:10.1137/0801026. URL `http://dx.doi.org/doi/10.1137/0801026`.

Å. Björck. *Numerical Methods for Least Squares Problems*. SIAM, 1996. ISBN 0-89871-360-9.

J.-Y. Bouguet. Camera Calibration Toolbox for Matlab. `www.vision.caltech.edu/bouguetj/calib_doc/`, 2010.

D. C. Brown. A solution to the general problem of multiple station analytical stereo triangulation. Technical Report Technical Report No. 43 (or AFMTC TR 58-8), Technical Report RCA-MTP Data Reduction, Patrick Airforce Base, Florida, 1958.

M. Bryson, M. Johnson-Roberson, and S. Sukkarieh. Airborne smoothing and mapping using vision and inertial sensors. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 3143–3148, Kobe, Japan, 2009. IEEE Press. ISBN 978-1-4244-2788-8.

L. Carlone, R. Aragues, J. Castellanos, and B. Bona. A Linear Approximation for Graph-based Simultaneous Localization and Mapping. In *Proceedings of Robotics: Science and Systems*, Los Angeles, CA, USA, June 2011.

M. Cummins and P. Newman. Appearance-only SLAM at large scale with FAB-MAP 2.0. *The International Journal of Robotics Research*, 2010. doi: 10.1177/0278364910385483. URL `http://ijr.sagepub.com/content/early/2010/11/11/0278364910385483.abstract`.

F. Dellaert and A. W. Stroupe. Linear 2D Localization and Mapping for Single and Multiple Robot Scenarios. In *ICRA*, pages 688–694. IEEE, 2002. ISBN 0-7803-7273-5.

Dellaert, F. and Kaess, M. Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing. *International Journal of Robotics Research*, 25(12):1181–1203, 2006. ISSN 0278-3649. doi: 10.1177/0278364906072768. URL http://dx.doi.org/10.1177/0278364906072768.

H. Durrant-Whyte and T. Bailey. Simultaneous Localization and Mapping: Part I. 13(12):99–110, June 2006.

A. W. Fitzgibbon and A. Zisserman. Automatic Camera Recovery for Closed or Open Image Sequences. In *ECCV (1)*, pages 311–326, 1998.

G. Grisetti, C. Stachniss, S. Grzonka, and W. Burgard. A tree parameterization for efficiently computing maximum likelihood maps using gradient descent. In *Proceedings of Robotics: Science and Systems (RSS)*, Atlanta, GA, USA, 2007.

Gurobi Optimization Inc. Gurobi optimizer reference manual, 2013. URL http://www.gurobi.com.

R. Hartley and F. Schaffalitzky. L_∞; minimization in geometric reconstruction problems. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–504 – I–509 Vol.1, june-2 july 2004. doi: 10.1109/CVPR.2004.1315073.

R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004. ISBN 0-521-54051-8.

T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, 2 edition, 2009. ISBN 0387952845.

K. L. Ho and P. M. Newman. Loop closure detection in SLAM by combining visual and spatial appearance. *Robotics and Autonomous Systems*, 54(9):740–749, 2006.

J. Hol, T. B. Schön, and F. Gustafsson. Modeling and Calibration of Inertial and Vision Sensors. *The international journal of robotics research*, 29(2):231–244, 2010.

S.-H. Jung and C. Taylor. Camera trajectory estimation using inertial sensor measurements and structure from motion results. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 2, pages II–732–II–737 vol.2, 2001. doi: 10.1109/CVPR.2001.991037.

F. Kahl. Multiple view geometry and the L_∞-norm. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1002 – 1009 Vol. 2, oct. 2005. doi: 10.1109/ICCV.2005.163.

F. Kahl and D. Henrion. Globally Optimal Estimates for Geometric Reconstruction Problems. *Int. J. Comput. Vision*, 74(1):3–15, Aug. 2007. ISSN 0920-

5691. doi: 10.1007/s11263-006-0015-y. URL http://dx.doi.org/10.1007/s11263-006-0015-y.

G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*, Nara, Japan, November 2007.

L. Kneip, M. Chli, and R. Siegwart. Robust Real-Time Visual Odometry with a Single Camera and an IMU. In *Proceedings of the British Machine Vision Conference*, pages 16.1–16.11. BMVA Press, 2011a. ISBN 1-901725-43-X. http://dx.doi.org/10.5244/C.25.16.

L. Kneip, A. Martinelli, S. Weiss, D. Scaramuzza, and R. Siegwart. Closed-form solution for absolute scale velocity determination combining inertial measurements and a single feature correspondence. In *Proceedings of the International Conference on Robotics and Automation*, Shanghai, China, May 2011b. URL http://hal.inria.fr/hal-00641772.

K. Konolige and M. Agrawal. FrameSLAM: From bundle adjustment to real-time visual mapping. 24(5):1066–1077, 2008.

H. W. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955. ISSN 1931-9193. doi: 10.1002/nav.3800020109. URL http://dx.doi.org/10.1002/nav.3800020109.

D. Lowe. Object Recognition from Local Scale-Invariant Features. In *Proceedings of the Seventh International Conference on Computer Vision (ICCV'99)*, pages 1150–1157, Corfu, Greece, 1999.

F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997.

T. Lupton and S. Sukkarieh. Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions. 28(1):61 –76, Feb. 2012. ISSN 1552-3098. doi: 10.1109/TRO.2011.2170332.

M. Kaess and A. Ranganathan and F. Dellaert. iSAM: Incremental smoothing and mapping. 24(6):1365–1378, Dec. 2008.

A. Martinelli. Vision and imu data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination. 28(1):44 –60, Feb. 2012. ISSN 1552-3098. doi: 10.1109/TRO.2011.2160468.

M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, page 593–598, Edmonton, Alberta, Canada, 28-1 July/Aug. 2002.

M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proceedings of the Sixteenth International Joint*

Conference on Artificial Intelligence (IJCAI), Acapulco, Mexico, 9-15 Aug. 2003.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 2nd edition, 2006. ISBN 978-0-387-40065-5.

C. Olsson and F. Kahl. Generalized Convexity in Multiple View Geometry. *Journal of Mathematical Imaging and Vision*, 38:35–51, 2010. ISSN 0924-9907. URL http://dx.doi.org/10.1007/s10851-010-0207-5. 10.1007/s10851-010-0207-5.

P. Moutarlier and R. Chatila. Stochastic multisensory data fusion for mobile robot location and environment modelling. In *5th International Symposium on Robotics Research*, pages 207–216, Tokyo, Japan, 1989.

C. H. Papadimitriou and K. Steiglitz. *Combinatorial Optimization: algorithms and complexity*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1982. ISBN 0-13-152462-3.

K. Sim and R. Hartley. Removing Outliers Using The L_∞ Norm. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 485 – 494, june 2006. doi: 10.1109/CVPR.2006.253.

S. N. Sinha, D. Steedly, and R. Szeliski. A Multi-staged Linear Approach to Structure from Motion. In *In RMLE-ECCV workshop, 2010.*, 2010.

M. A. Skoglund, Z. Sjanic, and F. Gustafsson. Initialisation and estimation methods for batch optimisation of inertial/visual SLAM. Submitted to, June 2014.

R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In *Autonomous robot vehicles*, pages 167–193. Springer-Verlag New York, Inc., New York, NY, USA, 1990. ISBN 0-387-97240-4.

H. Strasdat, J. M. M. Montiel, and A. Davison. Real-time monocular SLAM: Why filter? In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2657–2664, May 2010. doi: 10.1109/ROBOT.2010.5509636.

C. Taylor, D. Kriegman, and P. Anandan. Structure and Motion in Two Dimensions from Multiple Images: A Least Squares Approach. In *Proceedings of the IEEE Workshop on Visual Motion*, pages 242 –248, Princeton, NJ, USA, October 1991.

T. Thormählen, N. Hasler, M. Wand, and H.-P. Seidel. Merging of Feature Tracks for Camera Motion Estimation from Video. In *Proceedings of the 5th European Conference on Visual Media Production (CVMP 2008)*, pages 1 – 8, Hertfordshire, UK, Jan. 2008. IET.

S. Thrun and M. Montemerlo. The GraphSLAM algorithm with applications to large-scale mapping of urban structures. *International Journal on Robotics Research*, 25(5):403–430, 2006.

D. Titterton and J. Weston. *Strapdown Inertial Navigation Technology*. IEE radar, sonar, navigation and avionics series. Peter Peregrinus Ltd., Stevenage, UK, 1997. ISBN 0863413587.

B. Triggs, P. Mclauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment - a modern synthesis. In B. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, volume 1883 of *Lecture Notes in Computer Science*, pages 298–372. Springer-Verlag, 2000.

A. Vedaldi and B. Fulkerson. VLFeat: An Open and Portable Library of Computer Vision Algorithms, 2008. URL `http://www.vlfeat.org/`.

Z. Zhang and T. Kanade. Determining the epipolar geometry and its uncertainty: A review. *International Journal of Computer Vision*, 27:161–195, 1998.

Paper E

EM-SLAM with Inertial/Visual Applications

Authors: Zoran Sjanic , Martin A. Skoglund and Fredrik Gustafsson

Edited version of the paper:

Z. Sjanic, M. A. Skoglund, and F. Gustafsson. EM-SLAM with inertial/visual applications. Submitted to IEEE Transactions on Aerospace and Electronic Systems, June 2014.

EM-SLAM with Inertial/Visual Applications

Zoran Sjanic* **, Martin A. Skoglund* and Fredrik Gustafsson*

*Dept. of Electrical Engineering,
Linköping University,
SE–581 83 Linköping, Sweden
{zoran, ms,
fredrik}@isy.liu.se

**Dept. of Flight Data and Navigation,
Saab Aeronautics
SE-581 88 Linköping, Sweden

Abstract

The general Simultaneous Localisation and Mapping (SLAM) problem aims at estimating the state of a moving platform simultaneously with building a map of the local environment. There are essentially three classes of algorithms. EKF-SLAM and FastSLAM solve the problem on-line, while Nonlinear Least Squares (NLS) is a batch method. All of them scales badly with either the state dimension, the map dimension or the batch length. We investigate the EM algorithm for solving a generalized version of the NLS problem. This EM-SLAM algorithm solves two simpler problems iteratively, hence it scales much better with dimensions. The iterations switch between state estimation, where we propose an Extended Rauch-Tung-Striebel smoother, and map estimation, where a quasi-Newton method is suggested. The proposed method is evaluated in real experiments and also in simulations on a platform with a monocular camera attached to an inertial measurement unit. It is demonstrated to produce lower RMSE than with a standard Levenberg-Marquardt solver of NLS problem, at a computational cost that increases considerably slower.

1 Introduction

The aim in Simultaneous Localisation and Mapping (SLAM) is to estimate a moving platform's position and orientation while mapping the observed environment. In the seminal work of Smith et al. (1990) the idea of a stochastic map was presented and was first used in P. Moutarlier and R. Chatila (1989), where the estimate is computed with an Extended Kalman Filter (EKF) and some later ones are J.E. Guviant and E.M. Nebot (2001); J. J. Leonard and H. Jacob and S. Feder (2000). Another popular approach is the FastSLAM method, Montemerlo et al. (2002, 2003), which uses particle filters which are known to handle nonlinearities very well. However, these approaches have some downsides, like constantly increasing covariance matrix size for the EKF-SLAM, making it computationally infeasible for large data sets. Additionally, linearisation errors can contribute to

the inconsistencies in the estimates, Bailey et al. (2006a). However, FastSLAM estimates can also be inconsistent due to particle depletion Bailey et al. (2006b) and FastSLAM also scales badly with the state dimension of the moving platform. Recently, some approaches based on optimisation, and Nonlinear Least-Squares (NLS) in particular, have been proposed. These methods solve batch problems, i.e., a smoothed estimate is obtained, but for online applications a moving horizon strategy could be applied. A downside of NLS is the quadratic scaling in batch length. Popular approaches applying optimisation are e.g., GraphSLAM, Thrun et al. (2005) and Square Root Smoothing and Mapping, $\sqrt{\text{SAM}}$, Dellaert, F. and Kaess, M. (2006). A common denominator of these approaches is that they solve some form of Maximum Likelihood (ML) problem. Also it is usual that the sparsity of the problems is utilised enabling fast computations.

SLAM is a general class of problems where the combination of sensors vary and one sensor which have gained in popularity is the camera, and in particular monocular camera. Methods based on camera only has been known in the computer vision society as the structure from motion (SfM) problem for quite some time, see e.g., Fitzgibbon and Zisserman (1998); Taylor et al. (1991). The structure and motion recovered from SfM will have unknown universal scale, since the camera suffers from the depth ambiguity problem. In other words, given a motion of the camera, we cannot say if our velocity was large and the scene was far away or if the velocity was low and the scene was close to the camera. One way to solve this problem and to resolve the universal scale is to add some kind of velocity measurement, and the inertial measurement unit (IMU) measuring accelerations and angular velocities is one such way Bryson et al. (2009); Kneip et al. (2011a,b); Martinelli (2012); Lupton and Sukkarieh (2012).

Here, we propose a SLAM formulation in ML form which is based on the Expectation Maximisation algorithm Dempster et al. (1977). The method is evaluated on, but not restricted to, a system with a monocular camera and IMU sensors. In an EM setting, so called latent, or hidden, variables are introduced in order to solve ML problems that can be difficult. This is achieved by splitting the problem into two simpler problems, one where expectation with respect to the conditional density of the latent variables has to be calculated and one where a certain function needs to be maximised with respect to the parameters. These two steps are then repeated until convergence. This motivates the name of the method. In EM-SLAM, the map is viewed as the unknown parameter and the platform states, such as position and orientation, are considered to be the latent variables. As a simplified and intuitive motivation for this separation we can consider two simpler problems; one with known map and the other one with known trajectory and orientation. The first problem is then simply the navigation problem with known landmarks. The second problem is known as the triangulation problem, i.e., finding the landmark positions given the known platform positions and camera observations. See e.g., Hartley and Sturm (1997) for an example of triangulation application. Each of these problems are rather straightforward to solve separately but hard to solve combined. By separating the variables in the proposed way we basically split the SLAM problem into the above-mentioned two simpler

problems. In the first of these problems some approximations are necessary in order to implement the algorithm. In the conditional expectation step the latent variables are assumed to have Gaussian distribution and that they can be well approximated with an Extended Rauch-Tung-Striebel (E-RTS) smoother, Rauch et al. (1965). The maximization step is solved using a quasi-Newton method. The proposed method is compared with the NLS formulation which can be seen as a straightforward ML formulation where both the state sequence and the map are seen as parameters. This reference method is solved using the Levenberg-Marquardt algorithm Nocedal and Wright (2006). The comparison is done for both the performance of the estimation and for the complexity of each approach on both the simulations and small real data experiment.

The paper is organised as follows; in Section 2 the Expectation Maximisation algorithm is explained more detailed and application to SLAM is described in Section 3. The dynamical and measurement models specific to visual/inertial SLAM are introduced in Section 4 and an alternative method of solving ML SLAM problem, NLS, is explained in Section 5. Comparison between EM-SLAM and NLS-SLAM is discussed in Section 6, and a brief explanation about obtaining an initial estimate for the landmarks is given in Section 7. Finally, results, conclusions and future work are discussed in Section 8 and Section 9.

2 Expectation Maximisation

Maximum Likelihood in its basic form is a batch method which takes a set of observations $Y = \{y_1, \dots, y_N\}$, where the index denotes time, and aims at finding the maximum likelihood (ML) estimate of the parameters θ from the measurement likelihoods as

$$\hat{\theta}^{ML} = \arg\max_{\theta} \; p_\theta(Y), \tag{1}$$

which can be solved by considering minimizing the sum of the negative measurement log-likelihoods. Naturally, $p_\theta(Y)$ is the probability density function parametrized by the unknown θ. Often, the maximization of (1) can be very difficult and the key idea with Expectation Maximisation is to consider the joint density $p_\theta(Y, X)$, where $X = \{x_1, \dots, x_N\}$ are latent variables. Then, by splitting this density into two coupled, and hopefully easier, problems the parameters and the latent variables can be solved for in an iterative manner. The first step is the *Expectation step*, commonly denoted *E-step*, where the expectation of the joint log-likelihood, $\log p_\theta(Y, X)$, with respect to the density of the latent variable conditioned on all the measurements, $p_{\theta_k}(X|Y)$, is computed. The expectation $\mathbf{E}_{\theta_k}\{\log p_\theta(X, Y)|Y\}$ will be a function, called $\mathcal{Q}(\theta, \theta_k)$, of the parameter vector θ as

$$\mathcal{Q}(\theta, \theta_k) = \int p_{\theta_k}(X|Y) \log p_\theta(Y, X) \, \mathrm{d}X. \tag{2}$$

Note that the conditional density of the latent variables, $p_{\theta_k}(X|Y)$, is computed using the previous estimate of the parameters, θ_k, which is also emphasised in the

notation. In the *Maximisation step* or *M-step*, the Q function obtained in the E-step is maximized with respect to the parameter θ obtaining new estimate θ_{k+1}. These two steps are repeated until some convergence criterion is met, usually when the change in the parameter or likelihood value is below a certain threshold.

For an explanation of the EM algorithm applied to dynamical systems see e.g., Schön (2009), and how it can be used in system identification is exemplified in Wills et al. (2010), where a particle smoother is used to calculate the conditional expectation in the E-step. In Ghahramani and Roweis (1999); Duncan and Gyongy (2006) nonlinear dynamical models are treated using EM where the E-step is calculated using an Extended Kalman smoother, which is the same approach that will be used here. All these EM variants are formulated as batch methods, but there are also online EM methods which typically use sequential Monte Carlo and stochastic approximation methods Ozkan et al. (2012); Le Corff et al. (2011). However, in these online approaches it is assumed that either the joint log-likelihood belongs to the exponential family or that the state is low dimensional and can be well approximated with particle methods. In our case these assumptions are not met which will require other approximations to be applied.

3 EM-SLAM

We formulate the visual/inertial SLAM problem by defining a state space model as

$$x_t = f(x_{t-1}, u_t, w_t), \tag{3a}$$

$$y_t = h_t(x_t, \theta) + e_t, \tag{3b}$$

where the measurement noise, e_t, is considered white and Gaussian with mean zero and covariance R while the process noise, w_t, is considered white with mean zero and covariance Q. f describes the state transition function and u_t are considered to be inputs given by the inertial sensors from which pose and velocity, $x = [p, v, q]^T$, are computed. The measurements y_t here are the camera measurements i.e., features extracted from images and h is the measurement function relating measurements, states and parameters. The parameter vector θ, consists of landmark coordinates in three dimensions. These models will be defined in detail in Section 4. The most significant difference, as opposed to traditional SLAM state space model formulation, is that the map is seen as a parameter which parametrises the measurement equation and in turn, the measurement likelihood function. This formulation is quite natural since the map is time independent and is naturally seen as a parameter and not part of the state vector. Furthermore, the conditional expectation step is assumed to be well approximated by an Extended Rauch-Tung-Striebel (E-RTS) smoother. E-RTS is a straightforward modification of the standard RTS smoother, Rauch et al. (1965), by using the Extended Kalman Filter instead of the Kalman Filter in the forward filtering step, while the backward smoothing step is the same as in the original RTS smoother.

The state space formulation above constitutes the basis for the ML formulation that is naturally put into EM setting, i.e., it is straightforward to define the joint likelihood $p_\theta(X, Y)$. Here the platform states, X, are considered to be latent variables. By using the Markov properties this density can be written as

$$p_\theta(Y, X) = \prod_{t=1}^{N} p_\theta(y_t|x_t)p(x_t|x_{t-1}). \tag{4}$$

Notice that the process model does not depend explicitly on the parameter θ, which will simplify the calculations significantly as will be shown in the next section.

Next, both the E-step and the M-step will be explained in detail with all derivations and approximations used.

3.1 E-step

Given the joint likelihood from (4) the expectation step gets the following form

$$\mathcal{Q}(\theta, \theta_k) = \mathbf{E}_{\theta_k}\left\{\log\left[\prod_{t=1}^{N} p_\theta(y_t|x_t)p(x_t|x_{t-1})\right]\middle| Y\right\}, \tag{5}$$

where the measurement likelihood is given by the PDF

$$p_\theta(y_t|x_t) = p_\theta(e_t) = p_\theta(y_t - h_t(x_t, \theta)), \tag{6}$$

and the state transition density, $p(x_t|x_{t-1})$, does not depend on θ. Assuming that the likelihood has Gaussian distribution the expectation (5) becomes

$$\begin{aligned}
&\mathcal{Q}(\theta, \theta_k) = \text{const.} - \\
&\mathbf{E}_{\theta_k}\left\{\sum_{t=1}^{N} \frac{1}{2}\|y_t - h_t(x_t, \theta)\|^2_{R_t^{-1}} + \log p(x_t|x_{t-1})\middle| Y\right\} \\
&= -\sum_{t=1}^{N} \mathbf{E}_{\theta_k}\left\{\frac{1}{2}\|y_t - h_t(x_t, \theta)\|^2_{R_t^{-1}}\middle| Y\right\} + \text{const.}
\end{aligned} \tag{7}$$

where all the terms not depending on θ are lumped into a constant term, which will not affect the optimisation in the subsequent step. Due to the nonlinear nature of the measurement function, see Section 4.2, there is no closed form solution. Thus, some approximations are necessary and one such is

$$\begin{aligned}
\mathcal{Q}(\theta, \theta_k) \approx \text{const.} - \frac{1}{2}\sum_{t=1}^{N}\Big(&\|y_t - h_t(\hat{x}_{t|N}, \theta)\|^2_{R^{-1}} + \\
&\text{Tr}(R_t^{-1}\nabla_x h_t(\hat{x}_{t|N}, \theta)P^s_{t|N}(\nabla_x h_t(\hat{x}_{t|N}, \theta))^T)\Big)
\end{aligned} \tag{8}$$

Here, $\hat{x}_{t|N}$ is the smoothed estimate of the latent variable and $P^s_{t|N}$ is its covariance. The smoothed estimate is obtained with an E-RTS smoother which is summarised in Algorithm 1. The trace term can be thought of as a regularisation term to

Algorithm 1 Extended Rauch-Tung-Striebel Smoother (E-RTS)

Input: measurements $Y = \{y_1, \ldots, y_N\}$, inputs $U = \{u_1, \ldots, u_N\}$, covariance matrices Q and R, parameter estimate θ_k

Output: smoothed state estimates $\hat{x}^s = \{\hat{x}_{1|N}, \ldots, \hat{x}_{N|N}\}$, covariances $P^s_{1:N|N}$

1: Run a forward Extended Kalman filter (EKF) where measurement equation uses fixed value of the parameter $\theta = \theta_k$, and store time and measurement updates for states, $\hat{x}_{t|t}$, $\hat{x}_{t|t-1}$, the covariances $P_{t|t}$, $P_{t|t-1}$ and the Jacobians of the dynamics, $F_{t-1} = \frac{\partial}{\partial x} f(x_{t-1}, u_t, w_t)|_{x_{t-1}=\hat{x}_{t-1|t-1}, w_t=0}$, defined in (3a).

2: $P^s_{N|N} := P_{N|N}$

3: **for** $t = N : 2$ **do**

$S_{t-1} := P_{t-1|t-1} F^T_{t-1} P^{-1}_{t|t-1}$

4: $\hat{x}_{t-1|N} := \hat{x}_{t-1|t-1} + S_{t-1}(\hat{x}_{t|N} - \hat{x}_{t|t-1})$

$P^s_{t-1|N} := P_{t-1|t-1} + S_{t-1}(P^s_{t|N} - P_{t|t-1}) S^T_{t-1}$

5: **end for**

compensate for the usage of the estimated latent variables instead of the true ones. If the true ones have been used that term would vanish and only the nonlinear least squares part had to be solved. See Appendix for derivation of (8).

3.2 M-step

Maximisation of the $\mathcal{Q}$-function can be done using standard optimisation software. Optimisation software usually assumes that the cost function should be minimised, which can easily be obtained by defining a new function as $-\mathcal{Q}(\theta, \theta_k)$. In the continuation the minimisation of the function will be considered. An important special case is linear systems since then the minimisation step can be solved by linear least-squares. As for our particular setting, the function to be minimised is a nonlinear function of the parameters and nonlinear methods need to be used. We use a quasi-Newton method called BFGS, Nocedal and Wright (2006) since it is quite efficient but other choices are also possible. In this method, the inverse Hessian of the function to be optimised is recursively approximated using the gradient information. Also note that the inverse Hessian could be interpreted as an approximation of the map covariance. The BFGS algorithm is summarised in Algorithm 2.

4 Models

In this section the models in (3) will be specified. The sensors of interest are monocular camera and 6-DOF inertial sensors, i.e., gyroscopes and accelerometers, contained in a single sensor package. To reduce the state and parameter space the inertial sensors are considered as inputs to a process model. A minimal 3D point landmark parametrisation is used and its measurement is given by a

Algorithm 2 M-step (Quasi-Newton minimisation method with BFGS Hessian update)

Input: smoothed states $\hat{x}^s$, measurements Y, initial parameters θ_k, initial inverse Hessian approximation $B_0 = \lambda I,\ \lambda > 0$, termination threshold ε.
Output: θ_{k+1}.

$i := 0$
terminate := **false**
$\theta^i := \theta_k$
while not *terminate* **do**
  Compute search direction:
  $p_i := -B_i \nabla_\theta \mathcal{Q}(\theta^i, \theta_k)$
  Update the parameter:
  $\theta^{i+1} := \theta^i + \alpha_i p_i$
  where α_i is the step length computed by line search ensuring decrease in cost
  Compute:
  $s_i = \theta_{i+1} - \theta_i$
  $r_i = \nabla_\theta \mathcal{Q}(\theta^{i+1}, \theta_k) - \nabla_\theta \mathcal{Q}(\theta^i, \theta_k)$
  Update the inverse Hessian
  $B_{i+1} := \left(I - \frac{s_i r_i^T}{r_i^T s_i}\right) B_i \left(I - \frac{r_i s_i^T}{r_i^T s_i}\right) + \frac{s_i s_i^T}{r_i^T s_i}$
  if $\|\nabla_\theta \mathcal{Q}(\theta^{i+1}, \theta_k)\| < \varepsilon$ **then**
    terminate := **true**
  else
    $i := i + 1$
  end if
end while
$\theta_{k+1} := \theta^{i+1}$

pinhole projection model. Also, since, in general, an IMU has a higher sampling rate than a camera, a multirate system model is obtained. Basically, it implies that several state updates are performed, using the process model, between the measurements. This however poses no limitations for the methods presented here, since the E-RTS can easily handle multirate models.

4.1 IMU Parametrisation

The models for the gyroscopes and accelerometers are simple as they are only considered to be inputs to the process model. The gyroscope signals are denoted $u^\omega = [u_x^\omega, u_y^\omega, u_z^\omega]^T$ where the subscript refers to each axis of the body frame. Similarly the accelerometer signals are denoted $u^\mathrm{a} = [u_x^\mathrm{a}, u_y^\mathrm{a}, u_z^\mathrm{a}]^T$ which are also given in the sensor body frame. A discretised process model for the position velocity and rotation, $[p, v, q]$, in the local, inertial, navigation frame is then,

$$p_t = p_{t-1} + T v_{t-1} + \frac{T^2}{2} \mathsf{R}^T(q_{t-1}) \left(u_t^\mathrm{a} + g^b + w_t^\mathrm{a} \right) \tag{9a}$$

$$v_t = v_{t-1} + T \, \mathsf{R}^T(q_{t-1}) \left(u_t^\mathrm{a} + g^b + w_t^\mathrm{a} \right) \tag{9b}$$

$$q_t = \exp\left(\frac{T}{2} S_\omega(u_t^\omega + w_t^\omega) \right) q_{t-1} \tag{9c}$$

where the T denotes the sampling interval, $\mathsf{R}(q)$ is a rotation matrix parametrisation of the unit quaternion $q = [q_0, q_1, q_2, q_3]^T$ which describes the rotation from navigation to body frame, $g^b = \mathsf{R}(q) g^n$, is the gravity expressed in the body frame, $g^n = [0, 0, -g]$ is the local gravity vector expressed in the inertial frame where $g \approx 9.82$ and $\exp(\cdot)$ is here considered as the matrix exponential. The noise terms are assumed Gaussian and independent $[(w_t^\mathrm{a})^T, (w_t^\omega)^T]^T = w_t \sim \mathcal{N}(0, \operatorname{diag} Q_\mathrm{a}, Q_\omega)$. The skew-symmetric matrix

$$S_\omega(u^\omega) = \begin{bmatrix} 0 & -u_x^\omega & -u_y^\omega & -u_z^\omega \\ u_x^\omega & 0 & u_z^\omega & -u_y^\omega \\ u_y^\omega & -u_z^\omega & 0 & u_x^\omega \\ u_z^\omega & u_y^\omega & -u_x^\omega & 0 \end{bmatrix}, \tag{10}$$

parametrises the quaternion dynamics. This parametrisation is very similar to reduced-dimension observers in Rugh (1996).

4.2 Camera Measurements

The monocular camera is modeled as a standard pinhole camera, see cf. Hartley and Zisserman (2004). The camera calibration matrix and lens distortion was estimated prior to usage. Since the calibration and distortion are known the undistorted pixels can be pre-multiplied with the inverse of the camera matrix, thus the camera then works as a projective map in Euclidean space, $P : \mathbb{R}^3 \to \mathbb{R}^2$. The projection is defined as $P([X, Y, Z]) = [X/Z, Y/Z]$ and the Z coordinate is assumed positive and non-zero since otherwise the point would be behind the camera. Then a normalised camera measurement $y_t^m = [u_t, v_t]^T$ of a landmark,

m, at time t is

$$y_t^m = P(\mathsf{R}(q_t)(m - p_t)) + e_t^m \tag{11}$$

which relates the pose (position and orientation) of the camera to the 3D location of the point. The measurement noise is assumed i.i.d. Gaussian, $e_t^m = [e_t^u, e_t^v]^T \sim \mathcal{N}(0, R_m)$. The correspondence variables at time t, c_t^i, encode the measurement-landmark assignment, $y_t^i \leftrightarrow m^j$, which gives a subset of all M landmarks at time t, $\mathsf{M}_t = \{m^j\},\ j \in \{1, \ldots, M | \, c_t^i = j\}$. At time t the stacked measurement equation is then

$$\underbrace{\begin{bmatrix} u_t^1 \\ v_t^1 \\ \vdots \\ u_t^{\mathsf{N}_y} \\ v_t^{\mathsf{N}_y} \end{bmatrix}}_{y_t^m} = \underbrace{\begin{bmatrix} P(\mathsf{R}(q_t)(m^{c_t^1} - p_t)) \\ \vdots \\ P(\mathsf{R}(q_t)(m^{c_t^{\mathsf{N}_y}} - p_t)) \end{bmatrix}}_{h_t(x_t, \theta)} + e_t^m, \tag{12}$$

where c_t^i denotes the index of the corresponding landmark and N_y is the number of measurements, which of course varies over time. In this paper the correspondences assumed correctly solved for in the initialiation step (see Section 7) but in practice there will always be outliers of some kind. This is a strong assumption which should be treated carefully since faulty associations will bias the SLAM estimate. Interesting approaches to data association was exploited in e.g., Bibby and Reid (2007); Dellaert et al. (2003) which both make use of the EM algorithm to estimate correspondences.

5 Nonlinear Least-Squares

Another way of solving the ML SLAM problem is to consider all the interesting parameters explicitly instead of having position, velocity and orientation as hidden variables. In this case the parameter vector θ will consist of all unknown parameters, that is landmarks, accelerations in navigation frame and rate gyros. The dynamics for the velocity and position is in this case used as explicit constraints. In this setting it is also possible to include biases for accelerations and angular rates as parameters, which was avoided in the EM formulation. This is because the problem greatly simplifies if the parameters affect only the measurement relation, as already explained in Section 3. Note however, that these extra terms can be put in the state vector.

The measurement models for accelerations and angular rates are then

$$y_t^a = \mathsf{R}(q_t)(a_t - g^e) + b_a + e_t^a \tag{13a}$$

$$y_t^\omega = \omega_t + b_\omega + e_t^\omega \tag{13b}$$

and camera measurements are defined as in Equation (12)

$$y_t^m = h_t(p_t, q_t, \mathsf{M}_t) + e_t^m. \tag{14}$$

The unknown parameters are then accelerations, $a_{1:N}$, angular rates, $\omega_{1:N}$, initial velocity v_0, acceleration bias, b_a, angular velocity bias, b_ω, and landmark positions m. Under the assumption that all noises are Gaussian and white, i.e., $e_t^i \sim \mathcal{N}(0, R_i)$, the corresponding negative log-likelihood becomes

$$\begin{aligned} -\log p_\theta(Y) = \sum_{t=1}^{N} & \|y_t^a - \mathsf{R}(q_t)(a_t - g^e) - b_a\|^2_{R_a^{-1}} + \\ & \|y_t^\omega - \omega_t - b_\omega\|^2_{R_\omega^{-1}} + \\ & \|y_t^m - h_t(p_t, q_t, \mathsf{M}_t)\|^2_{R_m^{-1}}. \end{aligned} \tag{15}$$

where $\theta = [a_{1:N}^T, \omega_{1:N}^T, v_0^T, b_a^T, b_\omega^T, m^T]^T$, q_t is a function of $\omega_{1:t}$ and p_t is a function of v_0 and $a_{1:t}$. Maximum likelihood minimisation problem can now be formulated as

$$\hat{\theta}^{ML} = \arg\min_\theta \; -\log p_\theta(Y) \tag{16a}$$

$$\text{subject to} \quad \begin{bmatrix} p_t \\ v_t \end{bmatrix} = F^t \begin{bmatrix} p_0 \\ v_0 \end{bmatrix} + \sum_{i=1}^{t} F^{i-1} B a_i, \tag{16b}$$

$$F = \begin{bmatrix} I_3 & TI_3 \\ 0 & I_3 \end{bmatrix}, \quad B = \begin{bmatrix} \frac{T^2}{2} I_3 \\ TI_3 \end{bmatrix},$$

$$q_t = \left[\prod_{k=1}^{t} \exp\left(\frac{T}{2} S_\omega(\omega_k) \right) \right] q_0. \tag{16c}$$

The constraints can actually be removed by expanding them and substituting them into the cost function giving an unconstrained problem. This problem is solved with e.g., standard Levenberg-Marquardt solver. The estimate obtained in this way will be used to compare to the estimate obtained with the EM-SLAM method.

6 Computation Complexity

The main difference between NLS and EM approach is the number of parameters. While NLS has both landmarks and platform's motion as parameters, EM considers the motion as latent variables. Seen the other way around, the ML problem in (1) can be considered as a marginalised version of (15), where motion is integrated out. As such, the complexity of the EM approach is actually lower than the NLS approach. To see this, consider the problem sizes of the two approaches, given N time instances where landmarks are observed, M landmarks and N_m landmark measurements, the NLS problem will have $6(N-1)+3M+3$ variables (6 more if biases are also included) and $6(N-1)+2N_m$ measurements. So, the size

of the problem grows both with the number of landmarks and time and it grows quadratically in time. For the EM approach, the E-step is realised as an E-RTS smoother with constant size state vector (unlike traditional EKF-SLAM for example), meaning that the complexity increases linearly in the forward pass with the amount of time points, even if a multirate model is considered, since the measurements are assumed independent and can be processed iteratively. The backward pass of E-RTS has a worst case cubical complexity in the state dimension (= 10). For the M-step, the size of the problem to be solved is $3M$ variables and $2N_m$ measurements with a slightly more difficult problem than NLS. BFGS only uses gradient information and an approximate inverse Hessian model resulting in a worst case quadratic complexity (disregarding function evaluations) in the map size, $3M$, at each iteration. Furthermore, it is quite efficient having superlinear convergence properties Nocedal and Wright (2006). This means that each step in the EM-SLAM will eventually be cheaper in total when the number of time steps grows. For large maps, the limited memory version L-BFGS is recommended having complexity which is $\mathcal{O}(3Mm)$ for both storage and computations, where m is the number of iterations.

7 Obtaining an Initial Estimate

Both EM and NLS-SLAM need an initial value of the parameters in order to do the iterations. This initial value is also important for the performance of the methods, since both formulations are non-linear and non-convex. The initialisation can be performed by simply randomising parameter values but that can lead to solutions that are stuck in local minima. A better estimate of the initial values can be obtained by noting that the NLS-SLAM problem, defined in (16), is actually almost linear if rotations are fixed, Martinelli (2012). In that case (13b) is not needed any more and (13a) is linear in parameters. For the landmark measurements consider the projection according to Equation (11) which for fixed rotations can be rewritten as

$$\begin{bmatrix} [u_t \, \mathsf{R}_{3,:}(q_t) - \mathsf{R}_{1,:}(q_t)](m - p_t) \\ [v_t \, \mathsf{R}_{3,:}(q_t) - \mathsf{R}_{2,:}(q_t)](m - p_t) \end{bmatrix} = \begin{bmatrix} \mathsf{R}_{3,:}(q_t)(m - p_t)e_t^u \\ \mathsf{R}_{3,:}(q_t)(m - p_t)e_t^v \end{bmatrix}, \tag{17}$$

where $\mathsf{R}_{i,:}(q_t)$ denotes the i:th row of the rotation matrix. The only thing that makes this equation non-linear is the parameter dependent noise term. However this formulation leads to a well known Iterative Reweighted Least Squares (IRLS) method which is solved efficiently, see e.g., Björck (1996). The accuracy of the estimate obtained in this way is dependent of the fixed rotations, but it still constitutes a much better initial value for the EM and NLS-SLAM then simply random values, see Skoglund et al. (2013) for more details. The initial rotations can be obtained in several ways, for example simply by integrating rate gyros using Equation (9c), or by some camera based method like 8-point, see e.g., Hartley and Zisserman (2004). The first method works quite fine if the gyro bias is small, while the latter one demands that the scene geometry is beneficial.

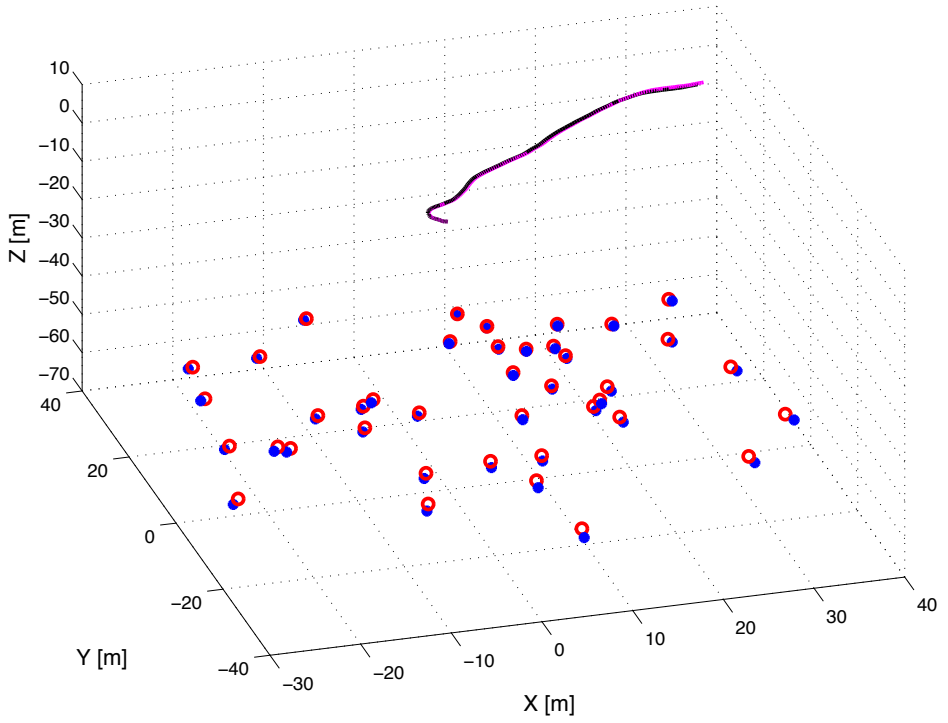


Figure 1: *The setup used in the MC simulations. True trajectory is in black and true landmarks are red circles. One of 30 simulation results is depicted as magenta trajectory and blue landmarks.*

8 Results

Evaluation of the proposed method is carried out on both simulated and experimental data.

8.1 Simulations

Simulations give the ability to choose noise levels, correspondences, the true parameters and the true accuracy of the method. Monte Carlo (MC) simulations with 30 different measurement noise realisations have been performed in order to evaluate the performance of the proposed method and to compare EM-SLAM with the NLS-SLAM. In Figure 1 the setup used for the simulations is illustrated. The true trajectory is in black and true landmarks are represented with red circles. One of the resulting trajectory (magenta) and landmark (blue stars) estimates is also plotted. Table 1 shows the average of the landmark estimation error for the two methods, while in Figure 2 the RMSE of the trajectory, for both methods, is plotted. In general it can be seen that the EM-SLAM method performs slightly better in average than the NLS-SLAM method.

Method	EM-SLAM	NLS-SLAM
Mean $\|\hat{\theta} - \theta^*\|/\mathbf{dim}(\theta^*)$	0.11	0.19

Table 1: *MC simulation results for the varying measurement noise (30 realisations). Note that θ contains only landmarks in this case.*

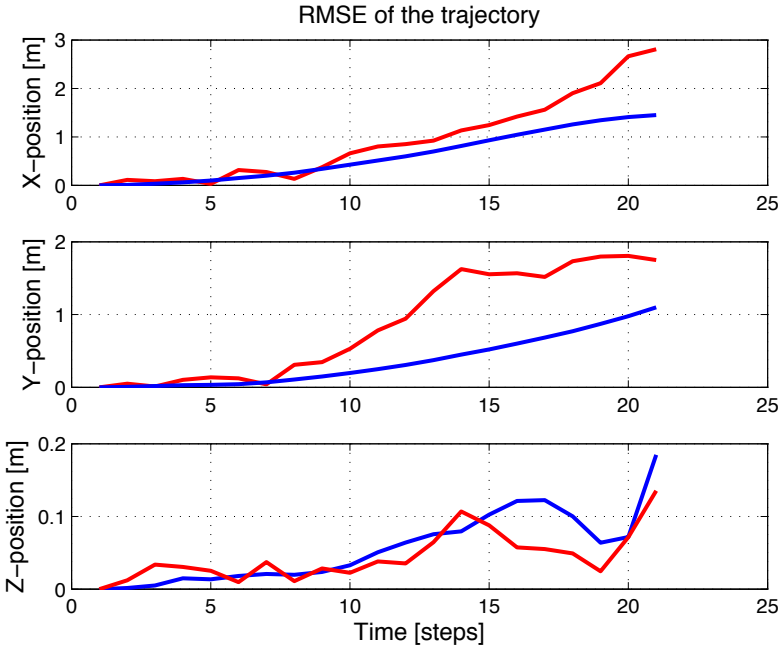


Figure 2: *RMSE of the EM-SLAM and NLS-SLAM estimated trajectories, EM in blue, NLS in red. 30 MC simulations are used.*

Figure 3: *Sensor unit containing both camera and inertial sensors.*

8.2 Real Data Experiments

In the experiments, a sensor unit, see Figure 3, equipped with monocular monochrome VGA camera (Pointgrey firefly) and three axis inertial sensors (gyroscopes and accelerometers) is used. The sensor unit also contains magnetometers, which are not used here, and a temperature sensor which is used for obtaining internal calibration of the inertial sensors. For the purpose of an accurately known ground truth the sensor unit was mounted at the tool position of an IRB-1400 industrial robot from ABB. Also, a small scene with objects of known size was created so that the estimated scene could be compared with respect to its size. Prior to usage, the camera was calibrated using the toolbox Bouguet (2010) and the relative pose of the camera centre with respect to the IMU centre was calibrated as described in Hol et al. (2010). A open source SIFT implementation from Vedaldi and Fulkerson (2008) was used to extract the features used as camera measurements. The results of the estimation are depicted in Figures 4 to 6. We see that both NLS and EM methods have similar performance on this data set. EM performs somewhat worse for the rotations which might be explained by the lack of biases as opposed to NLS.

9 Conclusions and Future Work

In this work we presented a Maximum Likelihood method for solving inertial/visual SLAM problem based on the EM algorithm. The particular structure of the SLAM

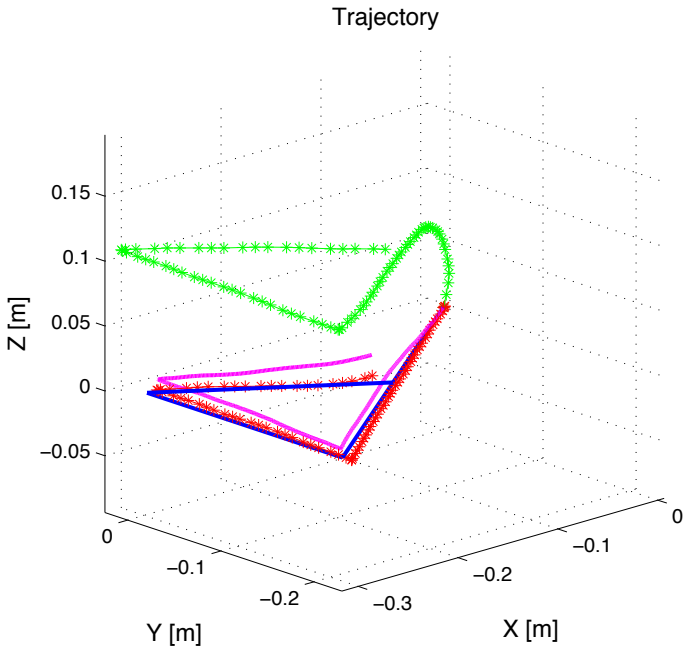


Figure 4: *Trajectory from the linear initialisation (green), EM (magenta), NLS (red) and ground truth (blue).*

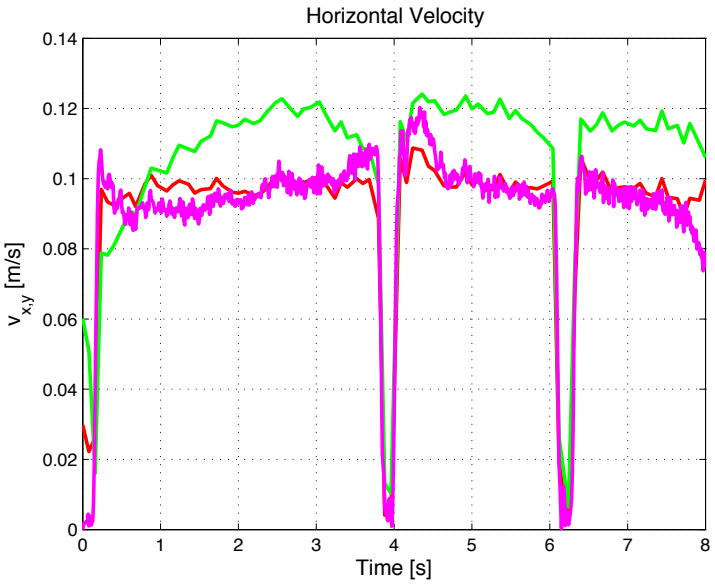


Figure 5: *Estimated horizontal velocity from the linear initialisation (green), EM (magenta) and NLS (red).*

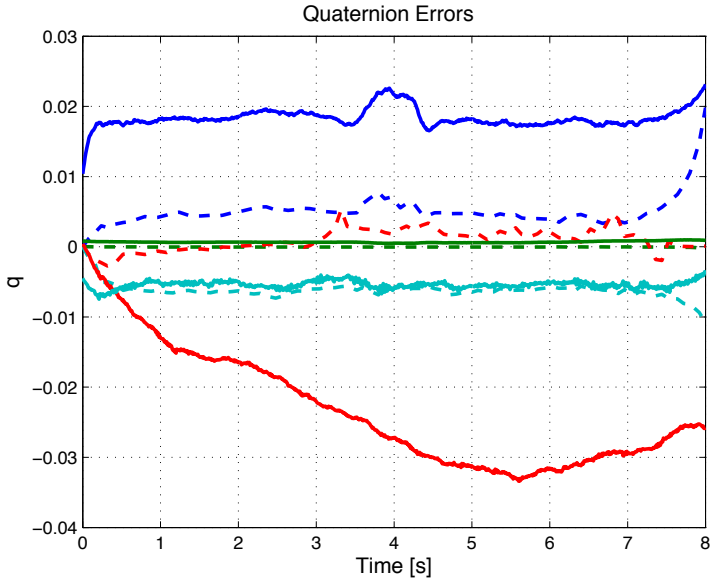


Figure 6: *Estimated quaternion error from EM (solid) and NLS (dashed).*

problem, where landmarks are seen as static parameters while the platform's motion is introduced as latent variables, makes the EM scheme a natural way to formulate the problem. This gives better scaling of the problem compared to both FastSLAM, EKF-SLAM and NLS. In particular, a qualitative analysis of the computational complexity is made comparing EM-SLAM and NLS. Furthermore, it is also shown that somewhat better results are obtained for EM-SLAM than the straightforward formulation in the form of NLS. These results are demonstrated on both simulated and real data sets,

In the future work it would be interesting to change E-RTS smoother estimate of the states with the particle smoother estimate, since it may handle nonlinearities in the models better and see if the performance improves.

Appendix

Given a smoothed estimate of the latent variables, $\hat{x}^s = \hat{x}_{1:N|N}$ the measurement function $h(x, \theta)$ can be linearised around these as

$$h(x,\theta) \approx \underbrace{h(\hat{x}^s,\theta)}_{\hat{h}} + \underbrace{\nabla_x h(\hat{x}^s,\theta)}_{H}\underbrace{(x-\hat{x}^s)}_{\tilde{x}}. \tag{18}$$

Using this approximation and expanding the norm in (7) for one time instant, while dropping the time index for readability, we obtain

$$\begin{aligned}
&\|y-\hat{h}-H\tilde{x}\|^2_{R^{-1}} = (y-\hat{h}-H\tilde{x})^T R^{-1}(y-\hat{h}-H\tilde{x}) = \\
&y^T R^{-1} y - y^T R^{-1}\hat{h} - y^T R^{-1} H\tilde{x} - \\
&\hat{h}^T R^{-1} y + \hat{h}^T R^{-1}\hat{h} + \hat{h}^T R^{-1} H\tilde{x} - \\
&(H\tilde{x})^T R^{-1} y + (H\tilde{x})^T R^{-1}\hat{h} + (H\tilde{x})^T R^{-1}(H\tilde{x})
\end{aligned} \tag{19}$$

and taking the expected value

$$\begin{aligned}
&\mathbf{E}_{\theta_k}\{\|y-\hat{h}-H\tilde{x}\|^2_{R^{-1}}\} = y^T R^{-1} y - y^T R^{-1}\hat{h} - \\
&\hat{h}^T R^{-1} y + \hat{h}^T R^{-1}\hat{h} + \mathbf{E}_{\theta_k}\{(H\tilde{x})^T R^{-1}(H\tilde{x})|Y\},
\end{aligned} \tag{20}$$

since all terms with only $\tilde{x}$ evaluate to zero under the assumption $(\tilde{x}|Y) \sim \mathcal{N}(0, P^s)$. Because $(H\tilde{x})^T R^{-1}(H\tilde{x})$ is scalar, it is equal to its trace and by using the trace rule $\mathrm{Tr}(A^T BA) = \mathrm{Tr}(BAA^T)$ together with the linearity of the trace and expectation operators, the last term becomes

$$\begin{aligned}
\mathbf{E}_{\theta_k}\{(H\tilde{x})^T R^{-1}(H\tilde{x})|Y\} &= \mathbf{E}_{\theta_k}\{\mathrm{Tr}((H\tilde{x})^T R^{-1}(H\tilde{x}))|Y\} \\
&= \mathbf{E}_{\theta_k}\{\mathrm{Tr}(R^{-1}H\tilde{x}\tilde{x}^T H^T)|Y\} \\
&= \mathrm{Tr}(R^{-1}H\mathbf{E}_{\theta_k}\{\tilde{x}\tilde{x}^T|Y\}H^T) \\
&= \mathrm{Tr}(R^{-1}HP^s H^T)
\end{aligned} \tag{21}$$

which results in

$$\begin{aligned}\mathcal{Q}(\theta, \theta_k) \approx \text{const.} - \\ \frac{1}{2}\sum_{t=1}^{N}\left(\|y_t - \hat{h}_t\|^2_{R^{-1}} + \text{Tr}(R^{-1}H_t P^s_{t|N} H_t^T)\right) = \\ \text{const.} - \frac{1}{2}\sum_{t=1}^{N}\Big(\|y_t - h_t(\hat{x}_{t|N}, \theta)\|^2_{R^{-1}} + \\ \text{Tr}(R^{-1}\nabla_x h_t(\hat{x}_{t|N}, \theta) P^s_{t|N}(\nabla_x h_t(\hat{x}_{t|N}, \theta))^T)\Big) \end{aligned} \tag{22}$$

which is the expression in (8).

Bibliography

T. Bailey, J. Nieto, J. E. Guivant, M. Stevens, and E. M. Nebot. Consistency of the EKF-SLAM algorithm. In *Proceedings of the International Conference on Intelligent Robots and Systems (IROS)*, pages 3562–3568, Beijing, China, 2006a. doi: 10.1109/IROS.2006.281644. URL http://dx.doi.org/10.1109/IROS.2006.281644.

T. Bailey, J. Nieto, and E. M. Nebot. Consistency of the FastSLAM Algorithm. In *International Conference on Robotics and Automation (ICRA)*, pages 424–429, Orlando, Florida, USA, 2006b.

C. Bibby and I. Reid. Simultaneous localisation and mapping in dynamic environments (SLAMIDE) with reversible data association. In *Proceedings of Robotics: Science and Systems*, Atlanta, GA, USA, 27-30 June 2007.

Å. Björck. *Numerical Methods for Least Squares Problems*. SIAM, 1996. ISBN 0-89871-360-9.

J.-Y. Bouguet. Camera Calibration Toolbox for Matlab. www.vision.caltech.edu/bouguetj/calib_doc/, 2010.

M. Bryson, M. Johnson-Roberson, and S. Sukkarieh. Airborne smoothing and mapping using vision and inertial sensors. In *Proceedings of the International Conference on Robotics and Automation (ICRA)*, pages 3143–3148, Kobe, Japan, 2009. IEEE Press. ISBN 978-1-4244-2788-8.

F. Dellaert, S. Seitz, C. Thorpe, and S. Thrun. EM, MCMC, and Chain Flipping for Structure from Motion with Unknown Correspondence. *Machine Learning*, 50(1-2):45–71, 2003.

Dellaert, F. and Kaess, M. Square Root SAM: Simultaneous Localization and Mapping via Square Root Information Smoothing. *International Journal of Robotics Research*, 25(12):1181–1203, 2006. ISSN 0278-3649. doi: 10.1177/0278364906072768. URL http://dx.doi.org/10.1177/0278364906072768.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977. ISSN 00359246. doi: 10.2307/2984875. URL http://web.mit.edu/6.435/www/Dempster77.pdf.

S. Duncan and M. Gyongy. Using the EM algorithm to estimate the disease parameters for smallpox in 17th century London. In *Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, 2006 IEEE*, pages 3312 –3317, oct. 2006. doi: 10.1109/CACSD-CCA-ISIC.2006.4777169.

A. W. Fitzgibbon and A. Zisserman. Automatic Camera Recovery for Closed or Open Image Sequences. In *ECCV (1)*, pages 311–326, 1998.

Z. Ghahramani and S. T. Roweis. Learning Nonlinear Dynamical Systems using an EM Algorithm. In *Advances in Neural Information Processing Systems*, volume 11, pages 599–605. MIT Press, 1999.

R. I. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146 – 157, 1997. ISSN 1077-3142. doi: 10.1006/cviu.1997.0547. URL http://www.sciencedirect.com/science/article/pii/S1077314297905476.

R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004. ISBN 0-521-54051-8.

J. Hol, T. B. Schön, and F. Gustafsson. Modeling and Calibration of Inertial and Vision Sensors. *The international journal of robotics research*, 29(2):231–244, 2010.

J. J. Leonard and H. Jacob and S. Feder. A Computationally Efficient Method for Large-Scale Concurrent Mapping and Localization. In *Proceedings of the Ninth International Symposium on Robotics Research*, pages 169–176, Salt Lake City, Utah, 2000. Springer-Verlag.

J.E. Guviant and E.M. Nebot. Optimization of the simultaneous localization and map-building algorithm for real-time implementation. *IEEE Transactions on Robotics and Automation*, 17(3):242–257, June 2001. ISSN 1042-296X.

L. Kneip, M. Chli, and R. Siegwart. Robust Real-Time Visual Odometry with a Single Camera and an IMU. In *Proceedings of the British Machine Vision Conference*, pages 16.1–16.11. BMVA Press, 2011a. ISBN 1-901725-43-X. http://dx.doi.org/10.5244/C.25.16.

L. Kneip, A. Martinelli, S. Weiss, D. Scaramuzza, and R. Siegwart. Closed-form solution for absolute scale velocity determination combining inertial measurements and a single feature correspondence. In *Proceedings of the International Conference on Robotics and Automation*, Shanghai, China, May 2011b. URL http://hal.inria.fr/hal-00641772.

S. Le Corff, G. Fort, and E. Moulines. Online Expectation Maximization algorithm to solve the SLAM problem. In *Statistical Signal Processing Workshop (SSP), 2011 IEEE*, pages 225 –228, june 2011. doi: 10.1109/SSP.2011.5967666.

T. Lupton and S. Sukkarieh. Visual-inertial-aided navigation for high-dynamic motion in built environments without initial conditions. 28(1):61 –76, Feb. 2012. ISSN 1552-3098. doi: 10.1109/TRO.2011.2170332.

A. Martinelli. Vision and imu data fusion: Closed-form solutions for attitude, speed, absolute scale, and bias determination. 28(1):44 –60, Feb. 2012. ISSN 1552-3098. doi: 10.1109/TRO.2011.2160468.

M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM: A factored solution to the simultaneous localization and mapping problem. In *Proceed-*

ings of the AAAI National Conference on Artificial Intelligence, page 593–598, Edmonton, Alberta, Canada, 28-1 July/Aug. 2002.

M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, Acapulco, Mexico, 9-15 Aug. 2003.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, New York, 2nd edition, 2006. ISBN 978-0-387-40065-5.

E. Ozkan, C. Fritsche, and F. Gustafsson. Online EM algorithm for joint state and mixture measurement noise estimation. In *Information Fusion (FUSION), 2012 15th International Conference on*, pages 1935 –1940, july 2012.

P. Moutarlier and R. Chatila. Stochastic multisensory data fusion for mobile robot location and environment modelling. In *5th International Symposium on Robotics Research*, pages 207–216, Tokyo, Japan, 1989.

H. E. Rauch, S. C. T., and T. F. Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, 3(8):1445–1450, 1965.

W. J. Rugh. *Linear System Theory*. Prentice Hall, Engelwood Cliffs, NJ, 2nd ed., 1996.

T. B. Schön. An Explanation of the Expectation Maximization Algorithm. Technical Report LiTH-ISY-R-2915, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden, Aug. 2009.

Z. Sjanic, M. A. Skoglund, and F. Gustafsson. EM-SLAM with inertial/visual applications. Submitted to, June 2014.

M. A. Skoglund, Z. Sjanic, and F. Gustafsson. Initialisation and Estimation Methods for Batch Optimisation of Inertial/Visual SLAM. Technical Report LiTH-ISY-R-3065, Department of Electrical Engineering, Linköping University, 2013.

R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In *Autonomous robot vehicles*, pages 167–193. Springer-Verlag New York, Inc., New York, NY, USA, 1990. ISBN 0-387-97240-4.

C. Taylor, D. Kriegman, and P. Anandan. Structure and Motion in Two Dimensions from Multiple Images: A Least Squares Approach. In *Proceedings of the IEEE Workshop on Visual Motion*, pages 242 –248, Princeton, NJ, USA, October 1991.

S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2005. ISBN 0262201623.

A. Vedaldi and B. Fulkerson. VLFeat: An Open and Portable Library of Computer Vision Algorithms, 2008. URL `http://www.vlfeat.org/`.

A. Wills, B. Ninness, and T. Schön. Estimating State-Space Models in Innovations Form using the Expectation Maximisation Algorithm. In *The 49th IEEE Conference on Decision and Control (CDC)*, Atlanta, USA, Dec. 2010.

PhD Dissertations
Division of Automatic Control
Linköping University

M. Millnert: Identification and control of systems subject to abrupt changes. Thesis No. 82, 1982. ISBN 91-7372-542-0.

A. J. M. van Overbeek: On-line structure selection for the identification of multivariable systems. Thesis No. 86, 1982. ISBN 91-7372-586-2.

B. Bengtsson: On some control problems for queues. Thesis No. 87, 1982. ISBN 91-7372-593-5.

S. Ljung: Fast algorithms for integral equations and least squares identification problems. Thesis No. 93, 1983. ISBN 91-7372-641-9.

H. Jonson: A Newton method for solving non-linear optimal control problems with general constraints. Thesis No. 104, 1983. ISBN 91-7372-718-0.

E. Trulsson: Adaptive control based on explicit criterion minimization. Thesis No. 106, 1983. ISBN 91-7372-728-8.

K. Nordström: Uncertainty, robustness and sensitivity reduction in the design of single input control systems. Thesis No. 162, 1987. ISBN 91-7870-170-8.

B. Wahlberg: On the identification and approximation of linear systems. Thesis No. 163, 1987. ISBN 91-7870-175-9.

S. Gunnarsson: Frequency domain aspects of modeling and control in adaptive systems. Thesis No. 194, 1988. ISBN 91-7870-380-8.

A. Isaksson: On system identification in one and two dimensions with signal processing applications. Thesis No. 196, 1988. ISBN 91-7870-383-2.

M. Viberg: Subspace fitting concepts in sensor array processing. Thesis No. 217, 1989. ISBN 91-7870-529-0.

K. Forsman: Constructive commutative algebra in nonlinear control theory. Thesis No. 261, 1991. ISBN 91-7870-827-3.

F. Gustafsson: Estimation of discrete parameters in linear systems. Thesis No. 271, 1992. ISBN 91-7870-876-1.

P. Nagy: Tools for knowledge-based signal processing with applications to system identification. Thesis No. 280, 1992. ISBN 91-7870-962-8.

T. Svensson: Mathematical tools and software for analysis and design of nonlinear control systems. Thesis No. 285, 1992. ISBN 91-7870-989-X.

S. Andersson: On dimension reduction in sensor array signal processing. Thesis No. 290, 1992. ISBN 91-7871-015-4.

H. Hjalmarsson: Aspects on incomplete modeling in system identification. Thesis No. 298, 1993. ISBN 91-7871-070-7.

I. Klein: Automatic synthesis of sequential control schemes. Thesis No. 305, 1993. ISBN 91-7871-090-1.

J.-E. Strömberg: A mode switching modelling philosophy. Thesis No. 353, 1994. ISBN 91-7871-430-3.

K. Wang Chen: Transformation and symbolic calculations in filtering and control. Thesis No. 361, 1994. ISBN 91-7871-467-2.

T. McKelvey: Identification of state-space models from time and frequency data. Thesis No. 380, 1995. ISBN 91-7871-531-8.

J. Sjöberg: Non-linear system identification with neural networks. Thesis No. 381, 1995. ISBN 91-7871-534-2.

R. Germundsson: Symbolic systems – theory, computation and applications. Thesis No. 389, 1995. ISBN 91-7871-578-4.

P. Pucar: Modeling and segmentation using multiple models. Thesis No. 405, 1995. ISBN 91-7871-627-6.
H. Fortell: Algebraic approaches to normal forms and zero dynamics. Thesis No. 407, 1995. ISBN 91-7871-629-2.
A. Helmersson: Methods for robust gain scheduling. Thesis No. 406, 1995. ISBN 91-7871-628-4.
P. Lindskog: Methods, algorithms and tools for system identification based on prior knowledge. Thesis No. 436, 1996. ISBN 91-7871-424-8.
J. Gunnarsson: Symbolic methods and tools for discrete event dynamic systems. Thesis No. 477, 1997. ISBN 91-7871-917-8.
M. Jirstrand: Constructive methods for inequality constraints in control. Thesis No. 527, 1998. ISBN 91-7219-187-2.
U. Forssell: Closed-loop identification: Methods, theory, and applications. Thesis No. 566, 1999. ISBN 91-7219-432-4.
A. Stenman: Model on demand: Algorithms, analysis and applications. Thesis No. 571, 1999. ISBN 91-7219-450-2.
N. Bergman: Recursive Bayesian estimation: Navigation and tracking applications. Thesis No. 579, 1999. ISBN 91-7219-473-1.
K. Edström: Switched bond graphs: Simulation and analysis. Thesis No. 586, 1999. ISBN 91-7219-493-6.
M. Larsson: Behavioral and structural model based approaches to discrete diagnosis. Thesis No. 608, 1999. ISBN 91-7219-615-5.
F. Gunnarsson: Power control in cellular radio systems: Analysis, design and estimation. Thesis No. 623, 2000. ISBN 91-7219-689-0.
V. Einarsson: Model checking methods for mode switching systems. Thesis No. 652, 2000. ISBN 91-7219-836-2.
M. Norrlöf: Iterative learning control: Analysis, design, and experiments. Thesis No. 653, 2000. ISBN 91-7219-837-0.
F. Tjärnström: Variance expressions and model reduction in system identification. Thesis No. 730, 2002. ISBN 91-7373-253-2.
J. Löfberg: Minimax approaches to robust model predictive control. Thesis No. 812, 2003. ISBN 91-7373-622-8.
J. Roll: Local and piecewise affine approaches to system identification. Thesis No. 802, 2003. ISBN 91-7373-608-2.
J. Elbornsson: Analysis, estimation and compensation of mismatch effects in A/D converters. Thesis No. 811, 2003. ISBN 91-7373-621-X.
O. Härkegård: Backstepping and control allocation with applications to flight control. Thesis No. 820, 2003. ISBN 91-7373-647-3.
R. Wallin: Optimization algorithms for system analysis and identification. Thesis No. 919, 2004. ISBN 91-85297-19-4.
D. Lindgren: Projection methods for classification and identification. Thesis No. 915, 2005. ISBN 91-85297-06-2.
R. Karlsson: Particle Filtering for Positioning and Tracking Applications. Thesis No. 924, 2005. ISBN 91-85297-34-8.
J. Jansson: Collision Avoidance Theory with Applications to Automotive Collision Mitigation. Thesis No. 950, 2005. ISBN 91-85299-45-6.
E. Geijer Lundin: Uplink Load in CDMA Cellular Radio Systems. Thesis No. 977, 2005. ISBN 91-85457-49-3.
M. Enqvist: Linear Models of Nonlinear Systems. Thesis No. 985, 2005. ISBN 91-85457-64-7.
T. B. Schön: Estimation of Nonlinear Dynamic Systems — Theory and Applications. Thesis No. 998, 2006. ISBN 91-85497-03-7.

I. Lind: Regressor and Structure Selection — Uses of ANOVA in System Identification. Thesis No. 1012, 2006. ISBN 91-85523-98-4.

J. Gillberg: Frequency Domain Identification of Continuous-Time Systems Reconstruction and Robustness. Thesis No. 1031, 2006. ISBN 91-85523-34-8.

M. Gerdin: Identification and Estimation for Models Described by Differential-Algebraic Equations. Thesis No. 1046, 2006. ISBN 91-85643-87-4.

C. Grönwall: Ground Object Recognition using Laser Radar Data – Geometric Fitting, Performance Analysis, and Applications. Thesis No. 1055, 2006. ISBN 91-85643-53-X.

A. Eidehall: Tracking and threat assessment for automotive collision avoidance. Thesis No. 1066, 2007. ISBN 91-85643-10-6.

F. Eng: Non-Uniform Sampling in Statistical Signal Processing. Thesis No. 1082, 2007. ISBN 978-91-85715-49-7.

E. Wernholt: Multivariable Frequency-Domain Identification of Industrial Robots. Thesis No. 1138, 2007. ISBN 978-91-85895-72-4.

D. Axehill: Integer Quadratic Programming for Control and Communication. Thesis No. 1158, 2008. ISBN 978-91-85523-03-0.

G. Hendeby: Performance and Implementation Aspects of Nonlinear Filtering. Thesis No. 1161, 2008. ISBN 978-91-7393-979-9.

J. Sjöberg: Optimal Control and Model Reduction of Nonlinear DAE Models. Thesis No. 1166, 2008. ISBN 978-91-7393-964-5.

D. Törnqvist: Estimation and Detection with Applications to Navigation. Thesis No. 1216, 2008. ISBN 978-91-7393-785-6.

P-J. Nordlund: Efficient Estimation and Detection Methods for Airborne Applications. Thesis No. 1231, 2008. ISBN 978-91-7393-720-7.

H. Tidefelt: Differential-algebraic equations and matrix-valued singular perturbation. Thesis No. 1292, 2009. ISBN 978-91-7393-479-4.

H. Ohlsson: Regularization for Sparseness and Smoothness — Applications in System Identification and Signal Processing. Thesis No. 1351, 2010. ISBN 978-91-7393-287-5.

S. Moberg: Modeling and Control of Flexible Manipulators. Thesis No. 1349, 2010. ISBN 978-91-7393-289-9.

J. Wallén: Estimation-based iterative learning control. Thesis No. 1358, 2011. ISBN 978-91-7393-255-4.

J. Hol: Sensor Fusion and Calibration of Inertial Sensors, Vision, Ultra-Wideband and GPS. Thesis No. 1368, 2011. ISBN 978-91-7393-197-7.

D. Ankelhed: On the Design of Low Order H-infinity Controllers. Thesis No. 1371, 2011. ISBN 978-91-7393-157-1.

C. Lundquist: Sensor Fusion for Automotive Applications. Thesis No. 1409, 2011. ISBN 978-91-7393-023-9.

P. Skoglar: Tracking and Planning for Surveillance Applications. Thesis No. 1432, 2012. ISBN 978-91-7519-941-2.

K. Granström: Extended target tracking using PHD filters. Thesis No. 1476, 2012. ISBN 978-91-7519-796-8.

C. Lyzell: Structural Reformulations in System Identification. Thesis No. 1475, 2012. ISBN 978-91-7519-800-2.

J. Callmer: Autonomous Localization in Unknown Environments. Thesis No. 1520, 2013. ISBN 978-91-7519-620-6.

D. Petersson: A Nonlinear Optimization Approach to H2-Optimal Modeling and Control. Thesis No. 1528, 2013. ISBN 978-91-7519-567-4.

Z. Sjanic: Navigation and Mapping for Aerial Vehicles Based on Inertial and Imaging Sensors. Thesis No. 1533, 2013. ISBN 978-91-7519-553-7.

F. Lindsten: Particle Filters and Markov Chains for Learning of Dynamical Systems. Thesis No. 1530, 2013. ISBN 978-91-7519-559-9.
P. Axelsson: Sensor Fusion and Control Applied to Industrial Manipulators. Thesis No. 1585, 2014. ISBN 978-91-7519-368-7.
A. Carvalho Bittencourt: Modeling and Diagnosis of Friction and Wear in Industrial Robots. Thesis No. 1617, 2014. ISBN 978-91-7519-251-2.