



HAL
open science

Smart motion sensor for navigated prosthetic surgery

Göntje Caroline Claasen

► **To cite this version:**

Göntje Caroline Claasen. Smart motion sensor for navigated prosthetic surgery. Other. Ecole Nationale Supérieure des Mines de Paris, 2012. English. NNT : 2012ENMP0008 . pastel-00691192

HAL Id: pastel-00691192

<https://pastel.hal.science/pastel-00691192>

Submitted on 25 Apr 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

École doctorale n°432: Sciences des Métiers de l'Ingénieur (SMI)

Doctorat ParisTech

T H È S E

pour obtenir le grade de docteur délivré par

l'École Nationale Supérieure des Mines de Paris

Spécialité "Mathématique et Automatique"

présentée et soutenue publiquement par

Göntje Caroline CLAASEN

le 17 février 2012

Capteur de mouvement intelligent pour la chirurgie prothétique naviguée

Directeur de thèse: **Philippe MARTIN**

Jury

Mme Isabelle FANTONI-COICHOT, Chargée de recherche HDR,
Heudiasyc UMR CNRS 6599, Université de Technologie de Compiègne

M. Philippe POIGNET, Professeur, LIRMM UM2-CNRS

M. Tarek HAMEL, Professeur, I3S UNSA-CNRS

M. Wilfrid PERRUQUETTI, Professeur, LAGIS FRE CNRS 3303-UL1-ECL

M. Frédéric PICARD, Docteur en médecine, chirurgien orthopédiste,
Golden Jubilee National Hospital, Glasgow

M. Philippe MARTIN, Maître de recherche, Centre Automatique et Systèmes,
MINES ParisTech

Rapporteur

Rapporteur

Examineur

Examineur

Examineur

Examineur

MINES ParisTech

Centre Automatique et Systèmes (CAS)

60, Bd Saint Michel - 75006 Paris - France

**T
H
È
S
E**

Graduate School n°432: Sciences des Métiers de l'Ingénieur (SMI)

ParisTech

P H D T H E S I S

To obtain the Doctor's degree from

École Nationale Supérieure des Mines de Paris

Speciality "Mathématique et Automatique"

defended in public by

Göntje Caroline CLAASEN

on February 17, 2012

**Smart motion sensor
for navigated prosthetic surgery**

Thesis advisor: **Philippe MARTIN**

Committee

Ms. Isabelle FANTONI-COICHOT, Chargée de recherche HDR,
Heudiasyc UMR CNRS 6599, Université de Technologie de Compiègne

Mr. Philippe POIGNET, Professor, LIRMM UM2-CNRS

Mr. Tarek HAMEL, Professor, I3S UNSA-CNRS

Mr. Wilfrid PERRUQUETTI, Professor, LAGIS FRE CNRS 3303-UL1-ECL

Mr. Frédéric PICARD, MD, Orthopaedic Surgeon, Golden Jubilee National Hospital, Glasgow

Mr. Philippe MARTIN, Maître de recherche, Centre Automatique et Systèmes,
MINES ParisTech

Reviewer

Reviewer

Examiner

Examiner

Examiner

Examiner

MINES ParisTech

Centre Automatique et Systèmes (CAS)

60, Bd Saint Michel - 75006 Paris - France

**T
H
È
S
E**

GÖNTJE CAROLINE CLAASEN

Centre Automatique et Systèmes
Unité Mathématiques et Systèmes
MINES ParisTech
60 boulevard St Michel
75272 Paris Cedex
France.

E-mail: caroline.claasen@mines-paristech.fr, goentje.claasen@gmx.de

Key words. - optical-inertial data fusion, Kalman filtering, nonlinear observers, computer-assisted surgery, servo-controlled handheld tool

Mots clés. - fusion de données optiques-inertielles, filtrage de Kalman, observateurs non-linéaires, chirurgie assistée par ordinateur, outil à main asservi

Education is the most powerful weapon which you can use to change the world.

NELSON MANDELA

REMERCIEMENTS

Tout d'abord je tiens à remercier Philippe Martin pour avoir encadré cette thèse pendant trois ans. Merci de votre enthousiasme pour ce projet, pour la recherche en générale et pour les choses qui marchent. Merci pour votre soutien et vos conseils.

Je souhaite remercier Isabelle Fantoni-Coichot ainsi que Philippe Poignet qui ont accepté d'être les rapporteurs de cette thèse. Je remercie également Tarek Hamel, Wilfrid Perruquetti et Frédéric Picard qui m'ont fait l'honneur de participer au jury de soutenance.

Un grand merci à tous les membres du Centre Automatique et Systèmes pour leur soutien et leurs conseils et pour avoir répondu à mes nombreuses questions sur la France et la langue française. Merci à mes camarades de thèse Eric, Florent et Pierre-Jean. Merci également à Erwan pour son aide.

I would like to thank Frédéric Picard and Angela Deakin from the Golden Jubilee Hospital in Glasgow for having introduced me to the world of orthopaedic surgery.

Danke meiner Familie, besonders meinen Eltern und Großeltern, der Familie Bezanson und natürlich Gregor für ihre immerwährende Unterstützung, die dieses Abenteuer erst möglich gemacht hat.

Je remercie le Fonds AXA pour la Recherche pour le financement de cette thèse.

Smart motion sensor
for navigated prosthetic surgery

RÉSUMÉ

Nous présentons un système de tracking optique-inertiel qui consiste en deux caméras stationnaires et une Sensor Unit avec des marqueurs optiques et une centrale inertielle. La Sensor Unit est fixée sur l'objet suivi et sa position et son orientation sont déterminées par un algorithme de fusion de données. Le système de tracking est destiné à asservir un outil à main dans un système de chirurgie naviguée ou assistée par ordinateur. L'algorithme de fusion de données intègre les données des différents capteurs, c'est-à-dire les données optiques des caméras et les données inertielles des accéléromètres et gyroscopes. Nous présentons différents algorithmes qui rendent possible un tracking à grande bande passante avec au moins 200Hz avec des temps de latence bas grâce à une approche directe et des filtres dits invariants qui prennent en compte les symétries du système. Grâce à ces propriétés, le système de tracking satisfait les conditions pour l'application désirée. Le système a été implémenté et testé avec succès avec un dispositif expérimental.

ABSTRACT

We present an optical-inertial tracking system which consists of two stationary cameras and a Sensor Unit with optical markers and an inertial measurement unit (IMU). This Sensor Unit is attached to the object being tracked and its position and orientation are determined by a data fusion algorithm. The tracking system is to be used for servo-controlling a handheld tool in a navigated or computer-assisted surgery system. The data fusion algorithm integrates data from the different sensors, that is optical data from the cameras and inertial data from accelerometers and gyroscopes. We present different algorithms which ensure high-bandwidth tracking with at least 200Hz with low latencies by using a direct approach and so-called invariant filters which take into account system symmetries. Through these features, the tracking system meets the requirements for being used in the desired application. The system was successfully implemented and tested with an experimental setup.

Contents

1	Introduction	1
1.1	Navigated Surgery Systems	2
1.2	Handheld Tools for Navigated Surgery	5
1.3	Smart Handheld Tool	6
1.4	Tracking for the Smart Handheld Tool	7
1.5	Servo-Control for 1D model	9
1.6	Outline	14
1.7	Publications	15
2	Optical-Inertial Tracking System	17
2.1	System Setup	17
2.1.1	Optical System	17
2.1.2	Inertial Sensors	22
2.2	Mathematical Model	24
2.2.1	Coordinate systems	24
2.2.2	Quaternions	24
2.2.3	Dynamics Model	27
2.2.4	Output Model	27
2.2.5	Noise Models	28
2.2.6	Complete Model	31
3	State of the Art	33
3.1	Computer Vision and Optical Tracking	33
3.1.1	Monocular Tracking	34
3.1.2	Stereo Tracking	37
3.1.3	Problems/Disadvantages	50
3.2	Optical-Inertial Tracking Systems	50
3.2.1	Systems Presented in the Literature	50
3.2.2	Motivation of Our Approach	53
3.3	Calibration	54

CONTENTS

3.3.1	Camera Calibration	54
3.3.2	IMU Calibration	55
3.3.3	Optical-Inertial Calibration	57
4	Data Fusion	59
4.1	Motivation	59
4.2	Extended Kalman Filter	61
4.2.1	Continuous EKF	61
4.2.2	Multiplicative EKF (MEKF)	62
4.2.3	Continuous-Discrete EKF	63
4.3	Data Fusion for Optical-Inertial Tracking	64
4.3.1	System Observability	64
4.3.2	Direct and Indirect Approaches	65
4.3.3	MEKF	65
4.3.4	Right- and Left-Invariant EKF	67
4.3.5	Covariance Parameters	77
4.3.6	Continuous-Discrete and Multi-rate	77
4.4	RIEKF for Calibration	79
4.4.1	Influence of calibration errors on the RIEKF	80
4.4.2	Calibration of Marker-Body Rotation with RIEKF	87
5	Implementation and Experimental Results	89
5.1	Experimental Setup	89
5.1.1	Cameras and Optical Markers	89
5.1.2	Inertial Sensors	94
5.1.3	Sensor Unit	94
5.1.4	Data Acquisition	94
5.1.5	Algorithm Implementation	97
5.1.6	Tracking References	103
5.2	Experiments	106
5.2.1	Experiment 1: static case	106
5.2.2	Experiment 2: slow linear motion	106
5.2.3	Experiment 3: fast oscillating linear motion	106
5.2.4	Experiment 4: static orientations	106
5.3	Results	110
5.3.1	General Observations	110
5.3.2	Precision and Accuracy	118
5.3.3	High-Bandwidth Tracking	119
5.3.4	Influence of Sensor Unit Calibration	124
5.4	Calibration	128

5.4.1	Optical System	128
5.4.2	Accelerometers	129
5.4.3	Gyroscopes	130
5.4.4	Sensor Unit	130
6	Conclusion	135

Chapter 1

Introduction

Les systèmes de chirurgie assistée par ordinateur sont de plus en plus utilisés dans les salles opératoires. Pour la pose de prothèses de genou, par exemple, un tel système mesure des points anatomiques, calcule la position optimale de la prothèse et indique les lignes de coupe. Actuellement, les coupes sont exécutées à l'aide de guides de coupe mécaniques, mais une technique de coupe sans guide mécanique est demandée par les chirurgiens. Différentes méthodes ont été proposées, par exemple utilisant un feedback visuel pour le chirurgien ou des systèmes robotiques. Nous considérons un outil à main asservi qui utilise un système de tracking pour déterminer la position de l'outil par rapport au patient et aux plans de coupe désirés. Le tracking doit avoir une bande passante d'au moins 200Hz pour pouvoir suivre le mouvement rapide de l'outil.

Comme aucun système adapté aux conditions de la salle opératoire et au coût possible d'un système de chirurgie n'existe, nous proposons un nouveau système de tracking qui utilise des capteurs optiques et inertiels pour déterminer la position et l'orientation d'un outil à main asservi.

Le système a une grande bande passante grâce aux capteurs inertiels haute fréquence. Il a un temps de latence réduit par rapport à des systèmes similaires grâce à deux caractéristiques: nous proposons une approche directe utilisant les données brutes des capteurs sans faire des calculs complexes comme dans l'approche standard, et nous utilisons des algorithmes de fusion de données qui prennent en compte les symétries du système ce qui réduit le temps de calcul.

Nous présentons des résultats de simulation pour un modèle simple d'un outil à main asservi avec différents systèmes de tracking illustrant l'intérêt d'un tel système.

1.1 Navigated Surgery Systems

Computer-assisted or navigated surgery systems have become more and more common in operating rooms over the last 15 years [DiGioia et al., 2004].

In orthopedic surgery systems, e.g. for knee replacement, it is important to make accurate bone cuts and place the prosthesis correctly. The system acquires relevant patient anatomical landmarks and calculates appropriate prosthesis placement based on built up frame of reference. It then defines the desired cutting planes for the knee prosthesis [DiGioia et al., 2004]. Such a computer-assisted surgery system for knee replacement is described in detail in [Stulberg et al., 2004].

Here we consider so-called image-free or CT-free surgery systems. Image-guided systems [Jolesz et al., 1996] use image data from video, computer tomography (CT), magnetic-resonance imaging (MRI) or ultrasound (US) to obtain patient anatomical information before and during surgery. These imaging techniques demand for important processing and some, like CT scans, also expose the patient to radiation. In contrast to this, image-free systems use optical tracking to determine anatomical landmarks [Mösges and Lavallé, 1996, DiGioia et al., 2004]. They make use of cameras but unlike image-based systems they do not treat the whole image. Instead, the cameras observe optical markers which are attached to the patient's bones. The optical markers are detected in the images and only the information of their point coordinates are used in the subsequent analysis instead of the whole image. The first image-free system for knee replacement used in an operating room was presented in [Leitner et al., 1997]. This system was later commercialized as OrthoPilot (Aesculap AG, Tuttlingen, Germany [Aesculap AG, 2011]). Figure 1.1 shows the OrthoPilot.

It is important to note the difference between computer-assisted surgery systems and robotic surgery systems. In the latter, surgical procedures are executed autonomously by a robotic system [Taylor and Stoianovici, 2003]. Robotic systems can perform these procedures with high accuracy but rise questions about safety and liability [Davies, 1996]. Computer-assisted surgery systems on the other hand, let the surgeon keep the control over the whole surgical procedure.

We now return to the original problem of cutting bones for knee replacement. In current systems, the bone cuts are executed with the help of cutting guides (also called cutting jigs) which are fixed to the patient's bone in accordance with the desired cutting planes [Stulberg et al., 2004]. They guide the bone saw mechanically with good accuracy. A cutting guide is depicted in Figure 1.2. Cutting guides have two main drawbacks: Mounting and repositioning the guide takes time and the procedure is invasive because the guide is pinned to the bone with screws.

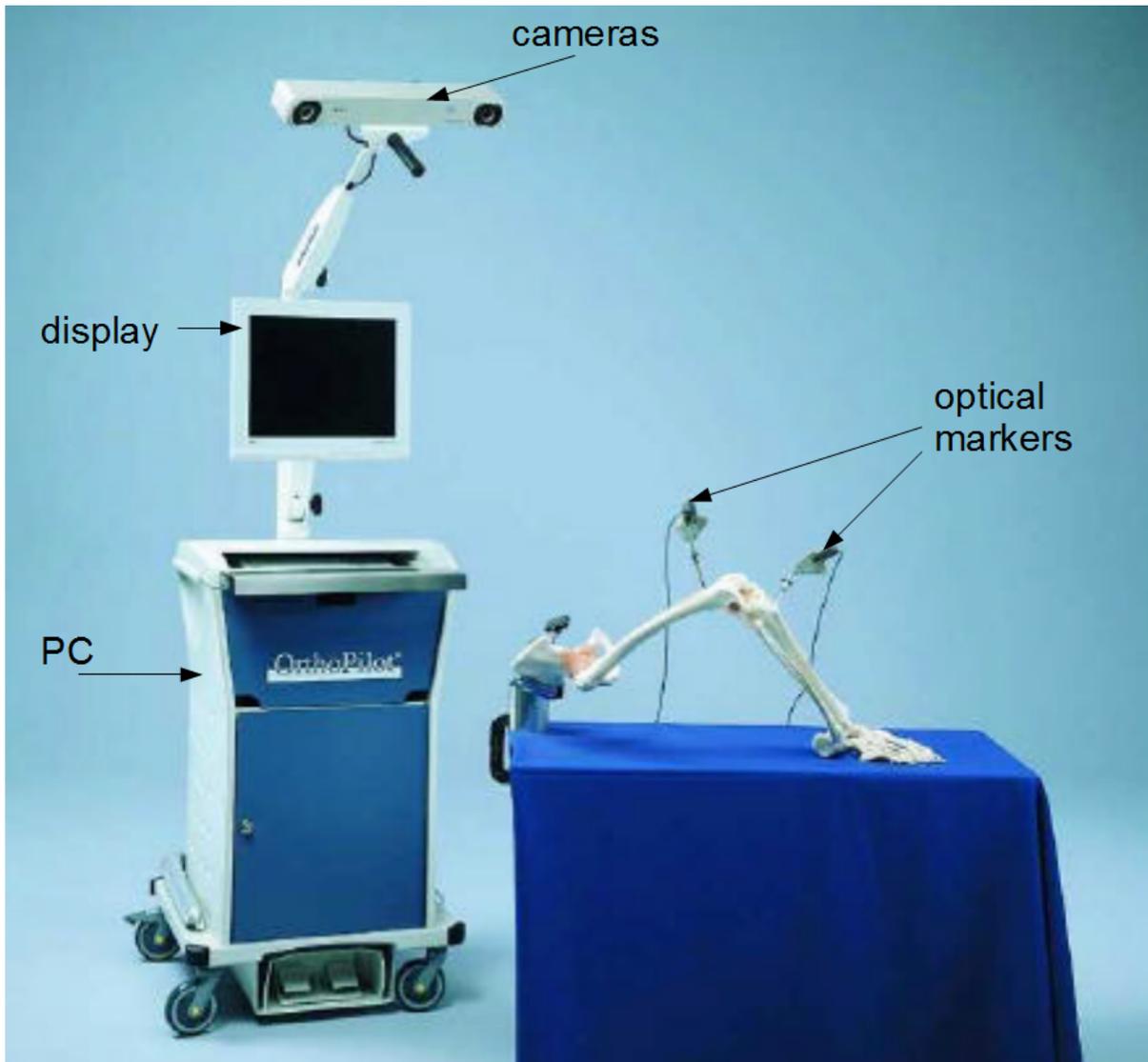


Figure 1.1: OrthoPilot® orthopaedic navigation system for knee replacement. Copyright Aesculap AG

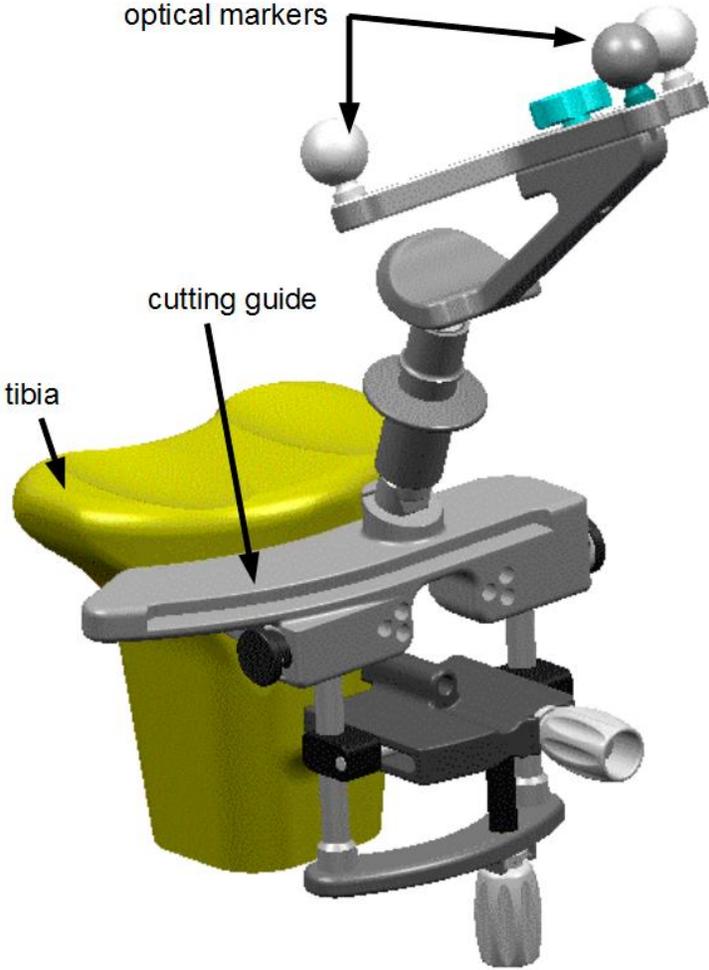


Figure 1.2: cutting guide. Copyright Aesculap AG

1.2 Handheld Tools for Navigated Surgery

Using a handheld saw without any cutting guides would have several advantages: the procedure would be less invasive, demand less surgical material and save time. Obviously, it would have to produce cuts with the same or even better accuracy to be a valuable improvement.

While a robotic system could achieve this task of cutting along a desired path, many surgeons wish to keep control over the cutting procedure. Therefore, an intelligent handheld tool should be used which combines the surgeon's skills with the accuracy, precision and speed of a computer-controlled system. Such a tool should be small and lightweight so as not to impede on the surgeon's work, compatible with existing computer-assisted surgery systems and relatively low-cost.

Controlling the tool position and keeping it along the desired cutting path necessitate the following steps:

1. define desired cutting plane relative to the patient,
2. track tool position and orientation relative to the patient and
3. compare desired and actual positions and correct the tool position accordingly.

The first step is done by the surgery system and the second by a tracking system. Step 3 can be executed either by the surgeon, by a robotic arm or directly by the handheld tool.

Several handheld tools have been developed in recent years, employing different strategies for the control of the tool position. In [Haider et al., 2007], the patient's bone and a handheld saw are tracked by an optical tracking system and the actual and desired cutting planes are shown on a screen. The surgeon corrects the position of the saw based on what he sees on the screen to make the actual and desired planes coincide. This approach is called "navigated freehand cutting" by the authors. A robotic arm is used in [Knappe et al., 2003] to maintain the tool orientation and correct deviations caused by slipping or inhomogeneous bone structure. The arm is tracked by an optical tracking system and internal encoders. The optical tracking also follows the patient's position. In [Maillet et al., 2005], a cutting guide is held by a robotic arm at the desired position, thus eliminating the problems linked to attaching cutting guides to the bone and repositioning. Several commercial systems with robotic arms are available, for example the Mako Rio (Mako Surgical, Fort Lauderdale, USA [Mako Surgical Corp., 2011]) and the Acrobot Sculptor (Acrobot LTD) [Jakopec et al., 2003].

Several "intelligent" handheld tools which are able to correct deviations from the desired cutting plane automatically by adapting the blade/drill position without a robotic arm have been developed. The systems presented in [Brisson et al., 2004] and [Kane et al., 2009] use an optical tracking system. [Schwarz et al., 2009] uses an

optical tracking system to determine the position of the patient and of the tool which also contains inertial sensors. The authors estimated the necessary tracking frequency to be of 100Hz to be able to compensate the surgeon's hand tremor which range is up to 12 Hz and the patient's respiratory motion.

The systems presented so far determine a desired cutting path based on absolute position measurements of tools and patients. In contrast, tools have been developed which are controlled relative to the patient only. [Ang et al., 2004] presents a handheld tool which uses accelerometers and magnetometers for tremor compensation. A handheld tool with actuators controlling the blade position is presented in [Follmann et al., 2010]. The goal is to cut the skull only up to a certain depth while the surgeon guides the tool along a path. While an optical tracking system is used in this work, an alternative version with a tracking system using optical and inertial sensors is presented in [Korff et al., 2010].

Three different approaches to bone cutting without cutting guides have been compared in [Cartiaux et al., 2010]: freehand, navigated freehand (surgeon gets visual feedback provided by a navigation system; similar to [Haider et al., 2007]) and with an industrial robot. The authors find that the industrial robot gives the best result and freehand cutting the worst.

The authors of [Haider et al., 2007] compared their navigated freehand cuts to those with conventional cutting jigs and found the cuts had rougher surfaces but better alignment.

1.3 Smart Handheld Tool

The handheld tool we consider here is supposed to be an extension for an image-free or image-based computer-assisted surgery system, hence it can make use of an optical tracking system but not of an active robotic arm. The tool is to be servo-controlled with motors in the tool joints which can change the blade position. We use a saw as an example but the same applies to drilling, pinning or burring tools. In the case of a saw for knee replacement surgery, the new smart handheld tool would eliminate the need for cutting guides.

The tracking system and particularly its bandwidth are a key for good performance of the servo-control. Firstly, the tracking system should be able to follow human motion and especially fast movements - these could be due to a sudden change of bone structure while cutting or to slipping of the surgeon's hand. These are the movements to be corrected. Secondly, it should be fast enough to let the servo-control make the necessary corrections. The faster the correction, the smaller the deviation will be. Finally, the servo-control should make the correction before the surgeon notices the error to avoid conflict between the control and the surgeon's reaction. We estimate the surgeon's perception time to be

of 10ms which corresponds to a frequency of 100Hz and therefore consider the necessary bandwidth for the tracking system to be at least 200Hz.

Commercially available optical tracking systems suitable for surgery have a bandwidth of only 60Hz, thus ruling out the use of this type of tracking for the smart handheld tool. Alternatively, there are electro-magnetic, mechanical and inertial tracking technologies. Electro-magnetic sensors are difficult to use in an operating room because of interferences of the electro-magnetic field with the surgical material. A robotic arm would provide mechanical tracking of the handheld tool but has been excluded because it would be too constrictive for the surgeon. Inertial sensors cannot be used on their own because of their drifts which cannot be compensated.

We propose to use a novel tracking system using both inertial and optical sensors which will be described in the next Section.

1.4 Tracking for the Smart Handheld Tool

In this thesis, we propose an optical-inertial tracking system which combines an optical tracking system with inertial sensors. These inertial sensors have a high bandwidth of typically 100-1000Hz [Groves, 2008, p. 111] and are suitable for use in an operating room. Since the inertial sensors we use are lightweight, of small size and relatively low-cost, they hardly change the size and weight of the handheld tool and add little cost to the existing optical systems. Even if high-frequency optical tracking became less expensive in the future, our optical-inertial setup with low frequency cameras would still be less expensive due to the low cost of inertial sensors. Also, inertial sensors bring further advantages as they can be used for disturbance rejection as will be shown in the following Section.

Our tracking system is to be used for servo-controlling a handheld tool. The proposed setup is shown in Figure 1.3 with a bone saw as an example for a tool. A servo-controlled handheld tool in combination with our proposed optical-inertial tracking system thus meets the requirements for an intelligent tool cited earlier. Such a tool would eliminate the need for cutting guides to perform bone cuts for total knee replacement and could also be used in other surgical applications.

In this thesis, we do not aim at developing such a tool but instead our tracking systems is intended to be used with an existing handheld tool.

An algorithm, called data fusion algorithm, is the heart of the proposed tracking system. It integrates inertial and optical sensor data to estimate the motion of the handheld tool and we study several algorithms with the goal of meeting the requirements for tracking for the handheld tool.

In contrast to other systems using optical and inertial sensors, we do not try to solve the line-of-sight problem. This is a problem occurring in optical tracking systems which can track an object only when there is a line-of-sight between cameras and markers.

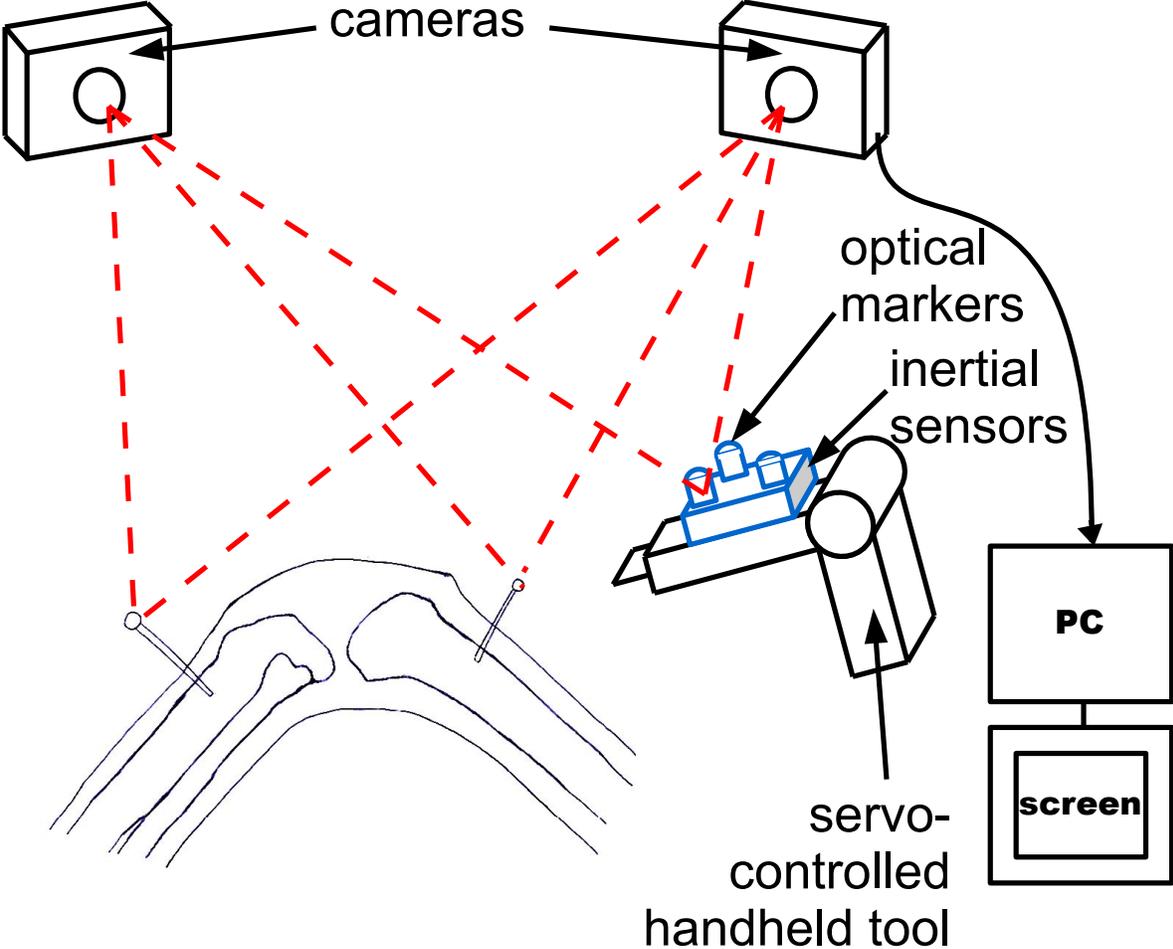


Figure 1.3: Optical-inertial tracking system with a servo-controlled handheld tool. Inertial sensors and optical markers are attached to the handheld tool.

In some optical-inertial systems presented in the literature, tracking depends on the inertial sensors only when there is no line-of-sight. Our goal is a tracking system with a high bandwidth and low latency, i.e. the tracking values should be available with as little delay as possible after a measurement has been made. This requires an algorithm which is particularly adapted to this problem and which reduces latency compared to similar systems using optical and inertial sensors as presented in [Tobergte et al., 2009] or [Hartmann et al., 2010] for example. High-bandwidth tracking is achieved by using inertial sensors with a sample rate of at least 200Hz and by executing the data fusion algorithm at this rate. To make tracking with low latencies possible, we propose an algorithm with a direct approach, i.e. it uses sensor data directly as inputs. This is opposed to the standard indirect approach which demands for computations on the measured values before using them in the data fusion algorithm. To further reduce latency, our algorithm takes into account the system geometry which should reduce computational complexity.

The following Section shows the positive effect of high-bandwidth optical-inertial tracking on servo-control of a handheld tool by means of a simulation with a simple model. In the rest of the thesis, we will concentrate on the setup of the optical-inertial tracking system and on the data fusion algorithm which integrates optical and inertial sensor data. We will discuss calibration of the hybrid setup and present an experimental setup and experimental results which show that the optical-inertial tracking system can track fast human motion. While we do not implement the servo-control we develop the necessary components for a high-bandwidth low-latency tracking system suitable to be used to servo-control a handheld tool in a computer-assisted surgery system.

1.5 Servo-Control for 1D model

In this Section, we are going to show the effect of a higher bandwidth in a Matlab/Simulink [Mathworks, 2011] simulation using a simple model of a handheld bone-cutting tool which is servo-controlled using different kind of measurements.

The tool in Figure 1.4 consists of a handle and a blade connected by a gearing mechanism which is actuated by a motor. The goal is to cut in y direction at a desired height z_r . The surgeon moves the tool in y direction at a speed v . A deviation from the desired z_r due to a change of bone structure is modeled by a disturbance D acting along z . In this simple model we assume that the disturbances can make the tool move in z direction only, which means that the tool's motion is constrained along z except for the cutting motion along y .

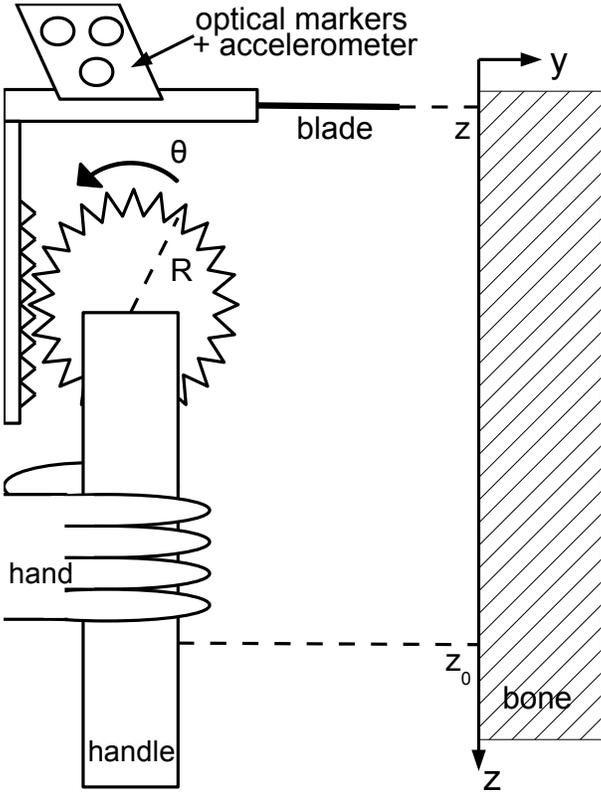


Figure 1.4: Handheld tool

The blade position z is determined by

$$z = R\theta + z_0 \quad (1.1)$$

$$\text{and } m\ddot{z} = F + D + mg \quad (1.2)$$

where R is the radius of the gear wheel, θ the wheel's angular position, z_0 the handle position, F the force applied by the gear, m the mass of the subsystem carrying the blade and g is gravity. The motor is governed by

$$J\ddot{\theta} = U - RF$$

where J is the motor and gear inertia and U the control input. Combining these equations gives

$$\ddot{z} = \underbrace{\frac{U}{mR + J/R}}_u + \underbrace{\frac{D}{m + J/R^2} + \frac{\ddot{z}_0 - g}{1 + mR^2/J}}_d + g. \quad (1.3)$$

This yields the simplified system

$$\dot{z} = v \quad (1.4)$$

$$\dot{v} = u + d + g. \quad (1.5)$$

The variable d includes the disturbance D due to bone structure as well as disturbances due to the surgeon motion (modeled by \ddot{z}_0).

An optical tracking system measures the position $z_m = z$ with a frequency of $1/T = 50Hz$ at discrete instants $z_{m,k} = z_m(kT)$, an inertial sensor (accelerometer) measures $a_m = u + d + a_b$ where a_b is the accelerometer constant bias.

The inertial measurements are considered continuous because their frequency is much higher than that of the optical ones.

We now present three systems using different types of measurements in a standard servo-control design. In all three cases, we do not take measurement noise into account and l , L , \tilde{L} , h and K are appropriately calculated constant gains. The disturbance d is modeled as a constant:

$$\dot{d} = 0.$$

System 1 uses only optical measurements $z_{m,k}$. An observer estimates the state $x = [z, v, d + g]^T$:

$$\text{prediction: } \hat{\hat{x}}^- = \begin{bmatrix} \hat{v}^- \\ u + \overline{(d + g)}^- \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \hat{\hat{x}}^- + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} u \quad (1.6)$$

$$\text{correction: } \hat{\hat{x}}_k = \hat{\hat{x}}_k^- + L(z_{m,k-1} - \hat{z}_{k-1}) \quad (1.7)$$

where $\hat{x}_k^- = \int_{kT-T}^{kT} \dot{\hat{x}}^-(\tau) d\tau$ with $\hat{x}^-(kT - T) = \hat{x}_{k-1}$. The state estimation is used in the controller which reads

$$u_k = -K\hat{x}_k + hz_r .$$

This system corresponds to the case where only an optical tracking system is used.

System 2 uses both optical and inertial data and represents the setup we propose in Chapter 2. A first observer with state $x = [z, v, a_b - g]^T$, measured input a_m and discrete optical measurements $z_{m,k}$ reads:

$$\text{prediction: } \dot{\hat{x}}^- = \begin{bmatrix} \hat{v} \\ a_m - \widehat{(a_b - g)}^- \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} \hat{x}^- + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} a_m \quad (1.8)$$

$$\text{correction: } \hat{x}_k = \hat{x}_k^- + L(z_{m,k-1} - \hat{z}_{k-1}) \quad (1.9)$$

where $\hat{x}_k^- = \int_{kT-T}^{kT} \dot{\hat{x}}^-(\tau) d\tau$ with $\hat{x}^-(kT - T) = \hat{x}_{k-1}$. This observer gives a continuous estimation $\hat{z}(t)$ which is used as a measurement $z_m(t)$ for a second observer with state $\tilde{x} = [\tilde{z}, \tilde{v}, \tilde{d} + g]^T$:

$$\dot{\tilde{x}} = \begin{bmatrix} \tilde{v} \\ \widehat{(g + \tilde{d})}_+ u \\ 0 \end{bmatrix} + \tilde{L}(z_m - \hat{z}) = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \tilde{x} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} u + \tilde{L}(z_m - \hat{z}) \quad (1.10)$$

This continuous state estimation of the second observer is used in the controller which reads

$$u = -K\tilde{x} + hz_r .$$

In system 3 which uses optical and inertial data we suppose that tracking and control are more tightly coupled than in system 2. A first observer is used to estimate the disturbance d with inertial measurements $a_m = u + d + a_b$:

$$\dot{\widehat{d + a_b}} = l(a_m - u - \widehat{(d + a_b)}) \quad (1.11)$$

This observer gives a continuous estimation $\widehat{d + a_b}(t)$ which is used as input for the second controller-observer. Its state is $x = [z, v, a_b - g]^T$ and it uses discrete optical measurements $z_{m,k}$:

$$\text{prediction: } \dot{\hat{x}}^- = \begin{bmatrix} \hat{v}^- \\ u + \widehat{(d + a_b)} - \widehat{(a_b - g)}^- \\ 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix} \hat{x}^- + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} (u + \widehat{(d + a_b)}) \quad (1.12)$$

$$\text{correction: } \hat{x}_k = \hat{x}_k^- + L(z_{m,k-1} - \hat{z}_{k-1}) \quad (1.13)$$

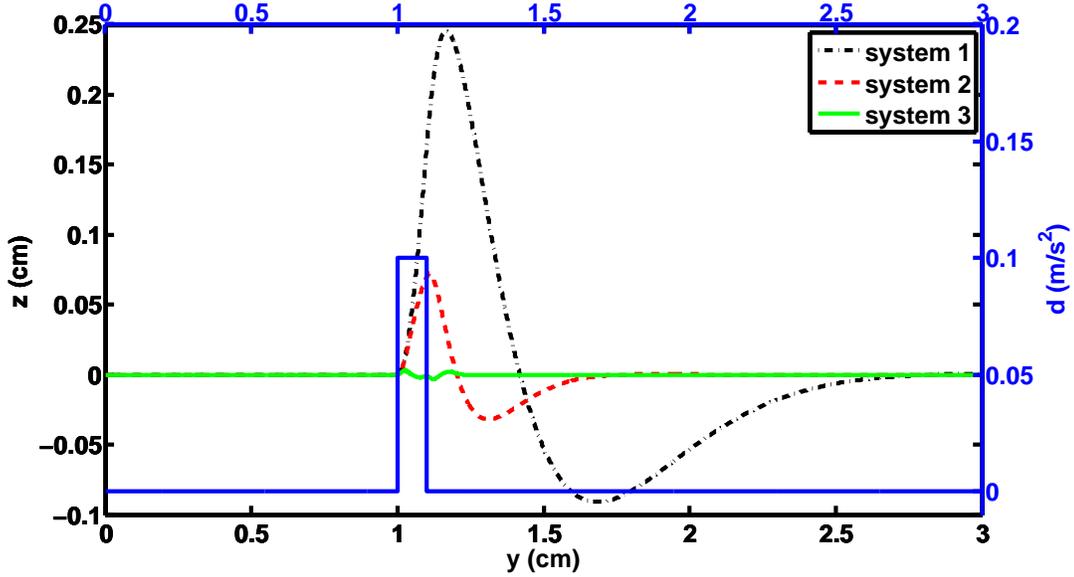


Figure 1.5: Simulation results for the 1D model of a handheld tool, showing the form of the cut for three different servo-control systems in response to a disturbance.

where $\hat{x}_k^- = \int_{kT-T}^{kT} \hat{\dot{x}}^-(\tau) d\tau$ with $\hat{x}^-(kT - T) = \hat{x}_{k-1}$. The control input is

$$u_k = -K\hat{x}_k + hz_r - (\widehat{d + a_b}) .$$

The handheld tool model and the three servo-control systems were implemented in a Simulink model. Figure 1.5 shows the simulated cuts for these three systems for a desired cutting position $z_r = 0\text{cm}$ and a disturbance d occurring from $t = 2.002\text{s}$ and $t = 2.202\text{s}$. At a speed of 0.5cm/s , this means the disturbance acts from $y = 1.001\text{cm}$ to $y = 1.101\text{cm}$. The disturbance causes the largest and longest deviation in the first system. In system 2, the position deviation can be corrected much faster and its amplitude is much smaller. Using the system 3 can correct the deviation even better. This simulation shows that using inertial sensors with a higher bandwidth allows the servo-control to correct a deviation caused by a disturbance much better than a system with a low bandwidth such as an optical tracking system.

It is important to note that the controller-observer for system 1 cannot be tuned to reject the disturbance faster; the choice of K and L is constrained by the frequency of the optical measurements.

System 1 corresponds to the case in which the existing optical tracking system in a computer-assisted surgery system would be used for a servo-controlled handheld saw without cutting guides. In the following chapters, we will look at an optical-inertial tracking system like the one in system 2. Here, the tracking is independent of the handheld

tool. In system 3, the tracking and servo-control are more tightly coupled. This allows for a even better disturbance rejection than in system 2 but the setup is more complex since the tool's model has to be known.

1.6 Outline

Optical-Inertial Tracking System The system setup and a mathematical model for the optical-inertial tracking system are described in Chapter 2.

State of the Art An overview of the state of the art of computer vision and optical tracking and of optical-inertial tracking systems is given in Chapter 3. The motivation of several choices for our optical-inertial tracking system as opposed to existing systems are exposed. Calibration methods for the individual sensors and for the hybrid optical-inertial system are discussed in Section 3.3.

Data Fusion Chapter 4 starts by presenting the Extended Kalman Filter (EKF) which is commonly used for data fusion of a nonlinear system and gives two of its variants: the Multiplicative EKF (MEKF) for estimating a quaternion which respects the geometry of the quaternion space and the continuous-discrete EKF for the case of a continuous system model and discrete measurements. Both of these variants are relevant to the considered problem since the Sensor Unit orientation is expressed by a quaternion and because the model for the Sensor Unit dynamics is continuous while the optical measurements are discrete.

The second part of the Chapter presents several data fusion algorithms for the optical-inertial tracking system, starting with an MEKF. Since our goal is to reduce the complexity of the algorithm, other variants are proposed, namely so-called invariant EKFs which exploit symmetries which are present in the considered system. We also discuss a fundamental difference of our data fusion algorithms to those presented in the literature which consists in using optical image data directly as a measurement instead of using triangulated 3D data. We call our approach a "direct" filter as opposed to "indirect" filters using 3D optical data.

The last part of the Chapter is dedicated to the analysis of the influence of parameter errors on one of the invariant EKFs for the optical-inertial system. Using this analysis, we propose a method for calibrating the optical-inertial setup using estimations from the invariant EKF.

Implementation Different aspects of the implementation of an optical-inertial tracking system are studied in Chapter 5. An experimental setup was developed consisting of low-cost cameras from the Wiimote and a Sensor Unit. A microcontroller is used for

synchronized data acquisition from optical and inertial sensors with sample rates of 50Hz and 250Hz respectively. The data fusion algorithm is implemented in Simulink and executed in real-time with an xPC Target application.

Several experiments have been conducted and analyzed to evaluate the performance of the tracking system. They show that the optical-inertial system can track fast motion and does so better than an optical tracking system.

This Chapter also discusses calibration of the individual sensors and of the hybrid setup. It gives the results of several calibration procedures and shows their influence on the tracking performance.

Conclusion To conclude, we give a summary of the work presented in this thesis and an outlook on future work.

1.7 Publications

Part of the work described in this thesis was published in articles at the following conferences:

- Claasen, G., Martin, P. and Picard, F.: Hybrid optical-inertial tracking system for a servo-controlled handheld tool. Presented at the 11th Annual Meeting of the International Society for Computer Assisted Orthopaedic Surgery (CAOS), London, UK, 2011
- Claasen, G. C., Martin, P. and Picard, F.: Tracking and Control for Handheld Surgery Tools. Presented at the IEEE Biomedical Circuits and Systems Conference (BioCAS), San Diego, USA, 2011
- Claasen, G. C., Martin, P. and Picard, F.: High-Bandwidth Low-Latency Tracking Using Optical and Inertial Sensors. Presented at the 5th International Conference on Automation, Robotics and Applications (ICARA), Wellington, New Zealand, 2011

The authors of the above articles have filed a patent application for the optical-inertial tracking with a direct data fusion approach. It has been assigned the Publication no. WO/2011/128766 and is currently under examination.

Chapter 2

Optical-Inertial Tracking System

Ce chapitre présente le système optique-inertiel que nous proposons. Nous expliquons les bases du fonctionnement du système optique et des capteurs inertiels et leurs erreurs et bruits respectifs. La deuxième partie traite du modèle mathématique du système en précisant les coordonnées utilisées, les équations de la dynamique et de sortie et les modèles des bruits.

2.1 System Setup

The goal of this tracking system is to estimate the position and orientation of a handheld object. A so-called Sensor Unit is attached to the object. The Sensor Unit consists of an IMU with triaxial accelerometers and triaxial gyroscopes and optical markers. The IMU and the markers are rigidly fixed to the Sensor Unit. A stationary stereo camera pair is placed in such a way that the optical markers are in its field of view. The setup is depicted in Figure 2.1.

The tracking system uses both inertial data from the IMU and optical data from the cameras and fuses these to estimate the Sensor Unit position and orientation.

This Section presents a few basic elements of the optical system and the inertial sensors which will serve as groundwork for the mathematical model of the system in Section 2.2.

2.1.1 Optical System

Vision systems treat images obtained from a camera, a CT scan or another imaging technique to detect objects, for example by their color or form. In contrast to this, an optical tracking system observes point-like markers which are attached to an object. Instead of treating and transmitting the whole image captured by the camera, they detect only the marker images and output their 2D coordinates.

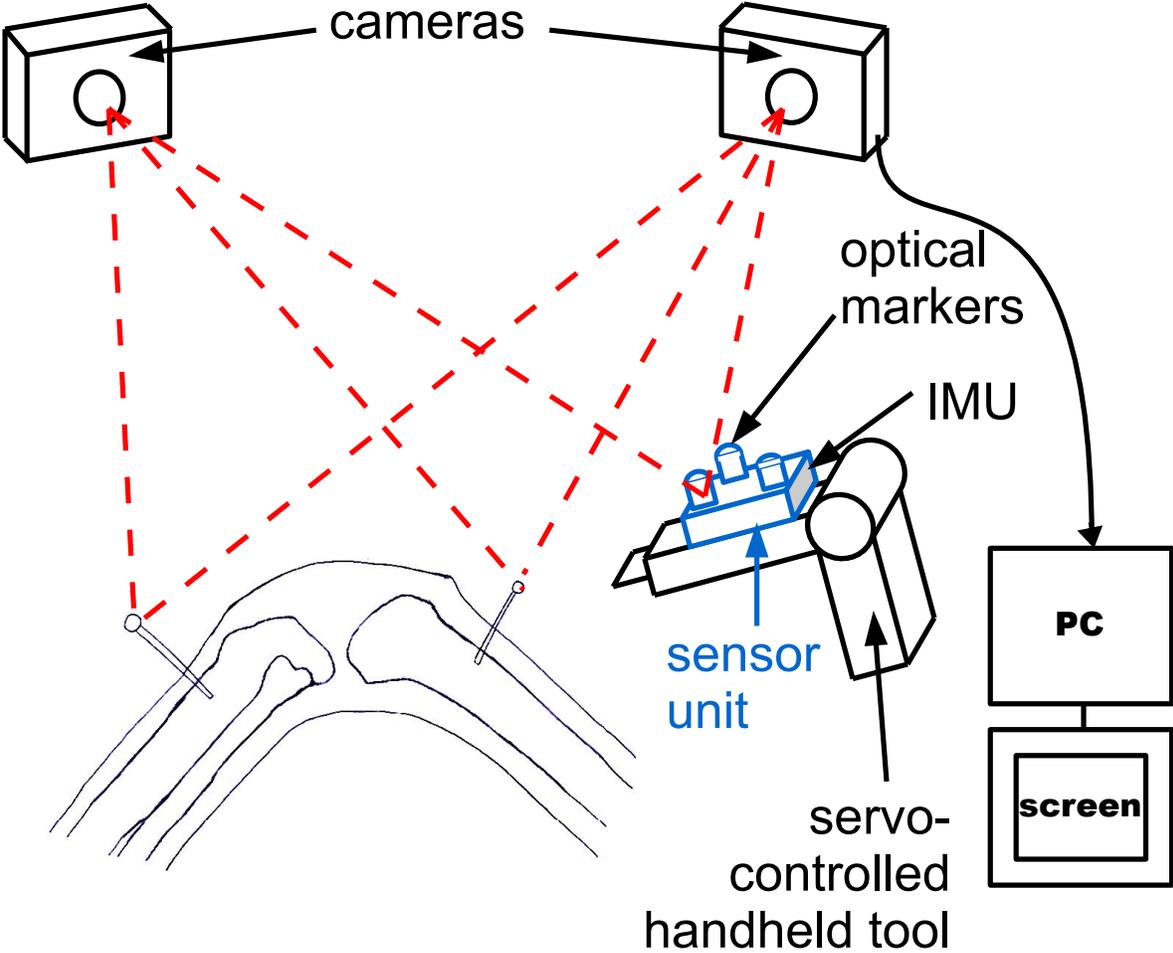


Figure 2.1: Optical-inertial tracking system with Sensor Unit and handheld tool

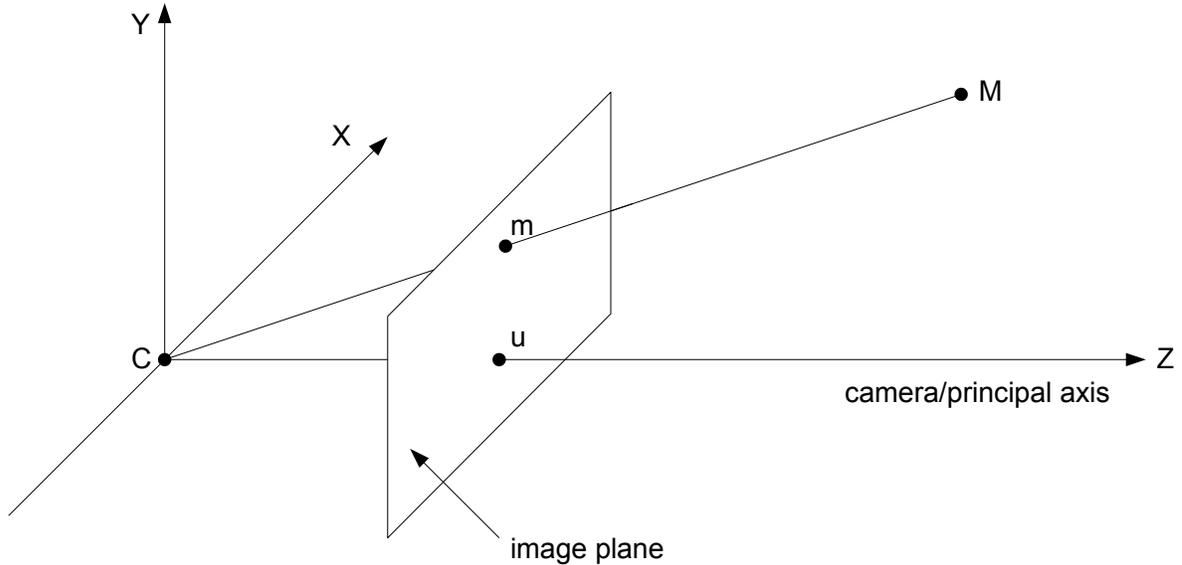


Figure 2.2: Pinhole camera model

The 2D marker coordinates can then be used to calculate information about the object position. For this, we need to model how a 3D point which represents an optical marker is projected to the camera screen. This Section describes the perspective projection we use and discusses problems and errors which can occur with optical tracking systems.

2.1.1.1 Pinhole Model

The pinhole model is the standard model for projecting object points to the camera image plane for CCD like sensors [Hartley and Zisserman, 2003, p. 153].

The model is depicted in Figure 2.2 [Hartley and Zisserman, 2003, p. 154]. C is the camera center. u is the principal point. A 3D point is denoted M and its projection in the image plane m . The distance between the camera center and the image plane is the focal distance f .

Figure 2.3 shows the concept of projection as given in [Hartley and Zisserman, 2003, p. 154]. The y coordinate of the image m can be calculated with the theorem of intercepting lines:

$$\frac{m_y}{Y} = \frac{f}{Z} \Leftrightarrow m_y = f \frac{Y}{Z}$$

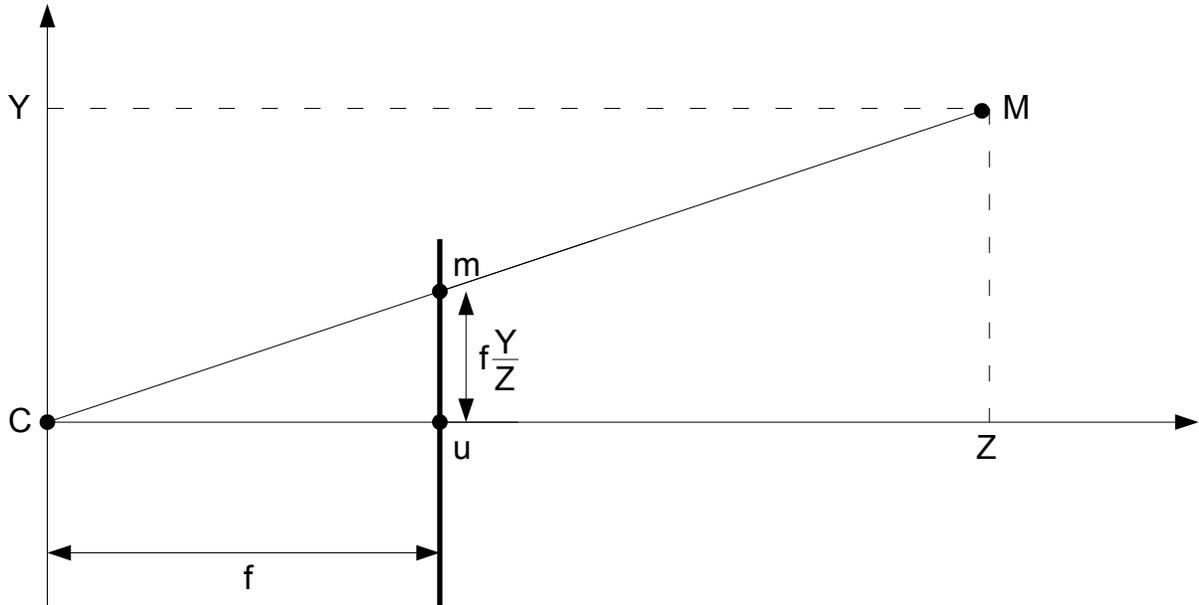


Figure 2.3: Projection

m_x can be calculated analogously. This gives for the image m :

$$m = \frac{f}{Z} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Often, images are expressed in the image plane, in coordinates attached to one corner of the image sensor. In this case, the principal point offset has to be taken into account:

$$m = \begin{bmatrix} f \frac{X}{Z} + u_0 \\ f \frac{Y}{Z} + v_0 \end{bmatrix} \quad (2.1)$$

where $u = [u_0, v_0]^T$ is the principal point, expressed in image plane coordinates.

Focal distance and principal point are camera parameters. They are called intrinsic values and have to be determined for each individual camera by a calibration procedure.

Projection in homogeneous coordinates Perspective projection for pinhole camera model is often expressed in homogeneous coordinates. In homogeneous coordinates, point M reads $\tilde{M} = [X, Y, Z, 1]^T$ and image m reads $\tilde{m} = [m_x, m_y, 1]^T$. Point \tilde{M} is then projected to \tilde{m} by:

$$\tilde{m} = P\tilde{M} \quad (2.2)$$

where P is the camera matrix. It is equal to

$$P = \underbrace{\begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_K [I_3 \ 0] \quad (2.3)$$

The projection (2.2) then writes

$$\tilde{m} = \underbrace{\begin{bmatrix} f & 0 & u_0 & 0 \\ 0 & f & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_P \tilde{M} \quad (2.4)$$

$$= \begin{bmatrix} fX + u_0Z \\ fY + v_0Z \\ Z \end{bmatrix} = Z \begin{bmatrix} fX/Z + u_0 \\ fY/Z + v_0 \\ 1 \end{bmatrix} = Z \begin{bmatrix} m \\ 1 \end{bmatrix} \quad (2.5)$$

The image m can then be retrieved from the homogeneous image \tilde{m} . The image is expressed in the image plane as in Equation (2.1).

The matrix K is called the intrinsic matrix. If f and $[u_0, v_0]$ are given in pixels, the projected image will be in pixels. If the pixels are not square, the aspect ration a which describes the relation between the width and height of a pixel has to be taken into account:

$$K = \begin{bmatrix} f & 0 & u_0 \\ 0 & af & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Alternatively, two focal lengths f_x and f_y for the sensor x and y dimensions can be used instead of the common focal length f [Szeliski, 2011, p. 47]. The intrinsic matrix K then writes

$$K = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

In this model, skew and lens distortion are not taken into account.

The values of the intrinsic matrix are usually determined for a individual camera by a calibration procedure as described in Section 3.3.1.

2.1.1.2 Noise/Error Sources

The image data measured by a camera can be corrupted by measurement noise and quantization noise where the latter depends on pixel size and resolution. When point images are considered, they may contain segmentation or blob detection errors.

For projection using the pinhole model, errors in the intrinsic values cause errors in the projected image.

2.1.2 Inertial Sensors

Inertial sensors designate accelerometers and gyroscopes which measure specific acceleration resp. angular rate without an external reference [Groves, 2008, p. 97].

An accelerometer measures specific acceleration along its sensitive axis. Most accelerometers have either a pendulous or a vibrating-beam design [Groves, 2008, p. 97].

A gyroscope measures angular rate about its sensitive axis. Three types of gyroscopes can be found which follow one of the three following principles: spinning mass, optical, or vibratory [Groves, 2008, p. 97].

Inertial sensors vary in size, mass, performance and cost. They can be grouped in five performance categories: marine-grade, aviation-grade, intermediate-grade, tactical-grade, automotive grade [Groves, 2008, p. 98].

The current development of inertial sensors is focused on micro-electro-mechanical systems (MEMS) technology. MEMS sensors are small and light, can be mass produced and exhibit much greater shock tolerance than conventional designs [Groves, 2008, p. 97].

An inertial measurement unit (IMU) consists of multiple inertial sensors, usually three orthogonal accelerometers and three orthogonal gyroscopes. The IMU supplies power to the sensors, transmits the outputs on a data bus and also compensates many sensor errors [Groves, 2008].

2.1.2.1 Noise/Error Sources

Several error sources are present in inertial sensors, depending on design and performance category. The most important ones are

- bias
- scale factor
- misalignment of sensor axes
- measurement noise

Each of the first three errors source has four components: a fixed term, a temperature-dependent variation, a run-to-run variation and an in-run variation [Groves, 2008]. The fixed and temperature-dependent terms can be corrected by the IMU. The run-to-run variation changes each time the IMU is switched but then stays constant while it is in use. The in-run variation term slowly changes over time.

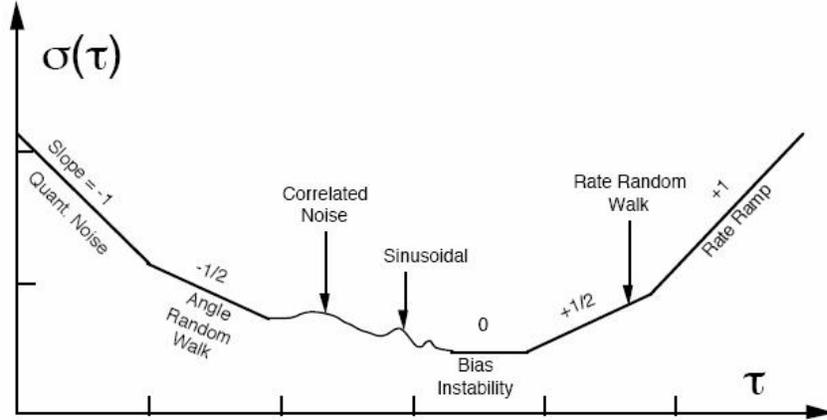


Figure 2.4: Sample plot of Allan variance analysis results from [IEEE, 1997]

Among the error sources listed above, bias and measurement noise are the dominant terms while scale factors and misalignment can be considered constant and can be corrected by calibration [Gebre-Egziabher, 2004, Gebre-Egziabher et al., 2004, Groves, 2008]. Bias and measurement noise for inertial sensor noise are often described using the Allan variance [Allan, 1966]. The Allan variance is calculated from sensor data collected over a certain length of time and gives power as a function of averaging time (it can be seen as a time domain equivalent to the power spectrum) [Gebre-Egziabher, 2004, Gebre-Egziabher et al., 2004]. For inertial sensors, different noise terms appear in different regions of the averaging time τ [IEEE, 1997] as shown in Figure 2.4 for a gyroscope. The main noise terms are the angle random walk, bias instability, rate random walk, rate ramp and quantization noise.

Since the Allan variance plots of the gyroscopes used in the implementation in Section 5 (see Section 5.1.5.3 for details on the noise parameters used in the implementation) can be approximated using only the angle random walk (for the measurement noise) and rate random walk (for the time-variation of the bias), we will use only these two terms in the gyroscope models presented in the following Section. The angle random walk corresponds to the line with slope $-1/2$ and its Allan variance is given [IEEE, 1997] as

$$\sigma^2(\tau) = \frac{N^2}{\tau}$$

where N is the angle random walk coefficient. The rate random walk is represented by the line with slope $+1/2$ and is given [IEEE, 1997] as

$$\sigma^2(\tau) = \frac{K^2\tau}{3}$$

where K is the rate random walk coefficient.

The accelerometer error characteristics can be described analogously to the gyroscope case, also using only two of the terms in the Allan variance plot. We will consider that possible misalignment and scale factors have been compensated for in the IMU or by a calibration procedure.

2.2 Mathematical Model

2.2.1 Coordinate systems

The motion of the Sensor Unit will be expressed in camera coordinates which are denoted by C and are fixed to the right camera center. ${}^C v$ is a velocity in camera coordinates, for example. The camera frame unit vectors are $E_1 = [1, 0, 0]^T$, $E_2 = [0, 1, 0]^T$ and $E_3 = [0, 0, 1]^T$. The camera's optical axis runs along E_1 . Image coordinates are expressed in the image sensor coordinate system S which is attached to one of the corners of the camera's image sensor.

The left camera coordinate system is denoted by CL and the image sensor coordinate system by SL . The left camera unit vectors are \tilde{E}_1 , \tilde{E}_2 and \tilde{E}_3 . Coordinates C and CL are related by a constant transformation.

The body coordinates, denoted by B , are fixed to the origin of the IMU frame and are moving relative to the camera frames. ${}^B a$ is an acceleration in body coordinates, for example.

The optical markers are defined in marker coordinates M . Marker and body coordinates are related by a constant transformation.

Finally, we also use an Earth-fixed world coordinate system, denoted by W .

The different frames are depicted in Figure 2.5. Figure 2.6 shows the image sensor and its frames in detail.

2.2.2 Quaternions

A quaternion q [Stevens and Lewis, 2003] consists of a scalar $q_0 \in \mathbb{R}$ and a vector $\tilde{q} \in \mathbb{R}^3$:

$$q = [q_0, \tilde{q}^T]^T .$$

The quaternion product of two quaternions s and q is defined as

$$s * q = \begin{bmatrix} s_0 q_0 - \tilde{s} \tilde{q} \\ s_0 \tilde{q} + q_0 \tilde{s} + \tilde{s} \times \tilde{q} \end{bmatrix} .$$

The cross product for quaternions reads:

$$s \times q = \frac{1}{2}(s * q + q * s) = \tilde{s} \times \tilde{q} .$$

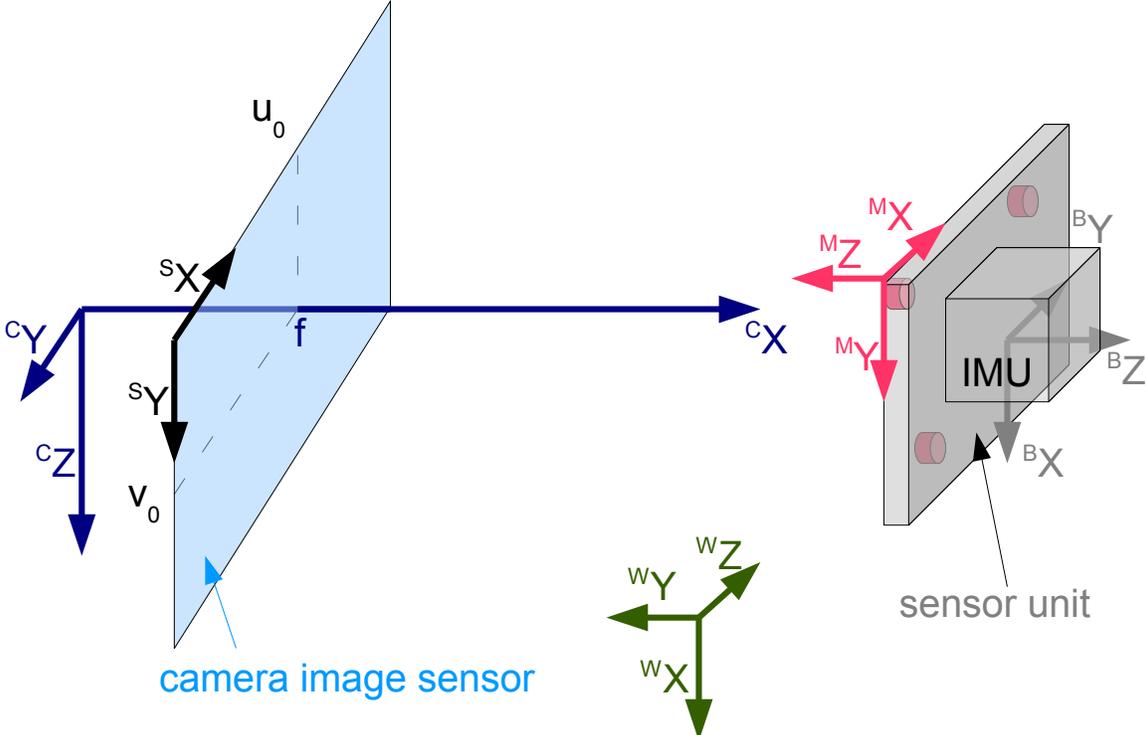


Figure 2.5: Camera and Sensor Unit

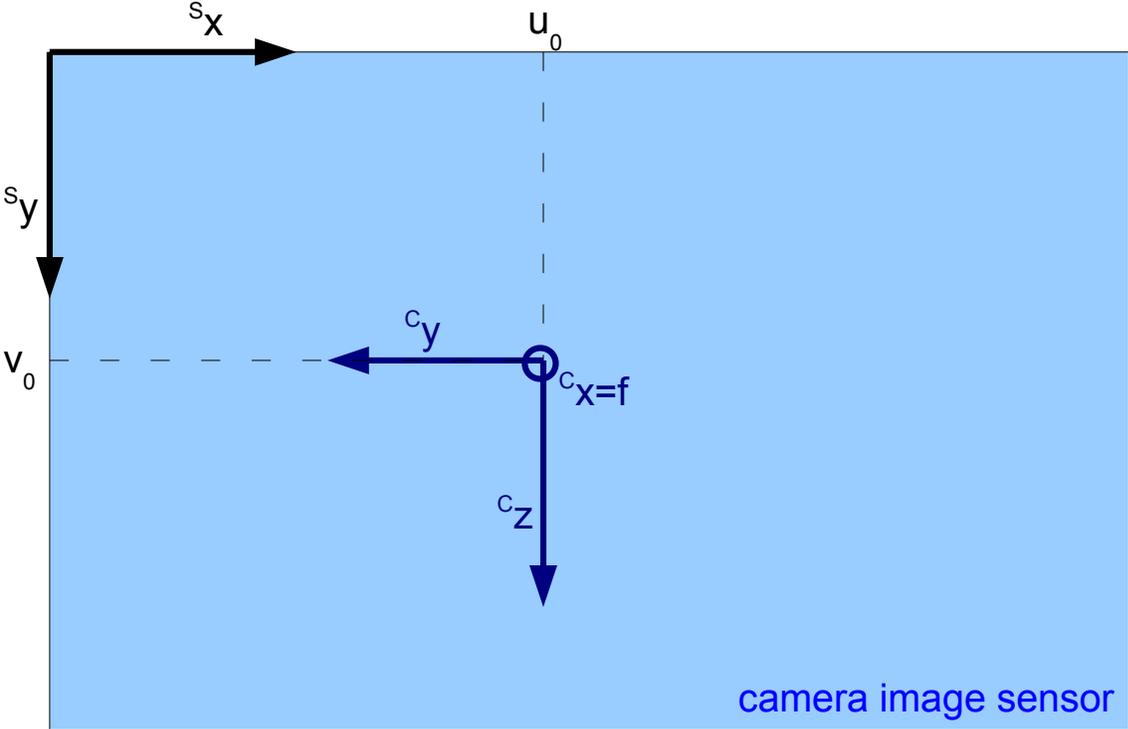


Figure 2.6: Camera image sensor with sensor and camera frames

A unit quaternion can be used to represent a rotation:

$$R(q)a = q * a * q^{-1}$$

where $a \in \mathbb{R}^3$ and $R(q)$ is the rotation matrix associated with quaternion q . If q depends on time, we have $\dot{q}^{-1} = -q^{-1} * \dot{q} * q^{-1}$. If

$$\dot{q} = q * a + b * q \quad (2.6)$$

with $a, b \in \mathbb{R}^3$ holds true, then $\|q(t)\| = \|q(0)\|$ for all t .

2.2.3 Dynamics Model

We have three equations representing the Sensor Unit motion dynamics:

$${}^C \dot{p} = {}^C v \quad (2.7)$$

$${}^C \dot{v} = {}^C G + {}^{BC} q * {}^B a * {}^{BC} q^{-1} \quad (2.8)$$

$${}^{BC} \dot{q} = \frac{1}{2} {}^{BC} q * {}^B \omega \quad (2.9)$$

where ${}^C G = {}^{WC} q * {}^W G * {}^{WC} q^{-1}$ is the gravity vector expressed in camera coordinates. ${}^W G = [0, 0, g]^T$ is the gravity vector in the world frame with $g = 9.81 \frac{m}{s^2}$ and ${}^{WC} q$ describes the (constant) rotation from world to camera coordinates. ${}^C p$ and ${}^C v$ are the Sensor Unit position and velocity, respectively. The orientation of the Sensor Unit with respect to camera coordinates is represented by the quaternion ${}^{BC} q$. ${}^B a$ and ${}^B \omega$ are the Sensor Unit accelerations and angular velocities.

2.2.4 Output Model

To project the markers to the camera we use the standard pinhole model from (2.1) but rewrite it using scalar products:

$${}^C y_i = \frac{f}{\langle {}^C m_i, {}^C E_1 \rangle} \begin{bmatrix} \langle {}^C m_i, {}^C E_2 \rangle \\ \langle {}^C m_i, {}^C E_3 \rangle \end{bmatrix} \quad (2.10)$$

with

$${}^C m_i = {}^C p + {}^{BC} q * {}^B m_i * {}^{BC} q^{-1}$$

where y_i is the 2D image of marker i with $i = 1, \dots, l$ (l is the number of markers). f is the camera's focal distance. To simplify notations, we use only one common focal length; if pixels are not square (see Section 2.1.1), we should write $\text{diag}(f, af)$ instead of f . ${}^C m_i$ and ${}^B m_i$ are the position of marker i in camera and body coordinates, respectively. $\langle a, b \rangle$ denotes the scalar product of vectors a and b .

The 2D image can be transformed from camera to sensor coordinates by a translation:

$${}^S y_i = {}^C y_i + {}^S u$$

where ${}^S u$ is the camera principal point.

The camera coordinate system in which the Sensor Unit pose is expressed is assumed to be attached to the right camera of the stereo camera pair. The transformation between the left and right camera coordinates is expressed by R_{St} and t_{St} :

$${}^{CL} p = R_{St} {}^C p + t_{St} \quad (2.11)$$

To project a marker to the left camera, replace ${}^C m_i$ in (2.10) by

$${}^{CL} m_i = R_{St} {}^C m_i + t_{St} \quad (2.12)$$

and use corresponding parameters and unit vectors:

$${}^{CL} y_i = \frac{f_L}{\langle {}^{CL} m_i, {}^{CL} \tilde{E}_1 \rangle} \begin{bmatrix} \langle {}^{CL} m_i, {}^{CL} \tilde{E}_2 \rangle \\ \langle {}^{CL} m_i, {}^{CL} \tilde{E}_3 \rangle \end{bmatrix}. \quad (2.13)$$

2.2.5 Noise Models

2.2.5.1 Inertial Sensor Noise Models

The noise models are motivated in Section 2.1.2.1 and will now be described mathematically as part of the dynamics model.

The gyroscope measurement ω_m is modeled as

$$\omega_m = {}^B \omega + \nu_\omega + {}^B \omega_b$$

where ω_m is the gyroscope measurement, ${}^B \omega$ is the true angular rate, ν_ω is the measurement noise and ${}^B \omega_b$ is the gyroscope bias. Note that the parameter ν_ω corresponds to the angle random walk coefficient N in the Allan variance in Section 2.1.2.1.

The gyroscope bias ${}^B \omega_b$ can be represented by a sum of two components: a constant one and a varying one. The bias derivative depends on the varying component which is modeled as a random walk as described in Section 2.1.2.1:

$${}^B \dot{\omega}_b = \nu_{\omega b}. \quad (2.14)$$

Note that the parameter $\nu_{\omega b}$ corresponds to the rate random walk coefficient K in the Allan variance in Section 2.1.2.1.

A more complex noise model is proposed in [Gebre-Egziabher et al., 2004, Gebre-Egziabher, 2004] which also models the variation of the time-varying part of the

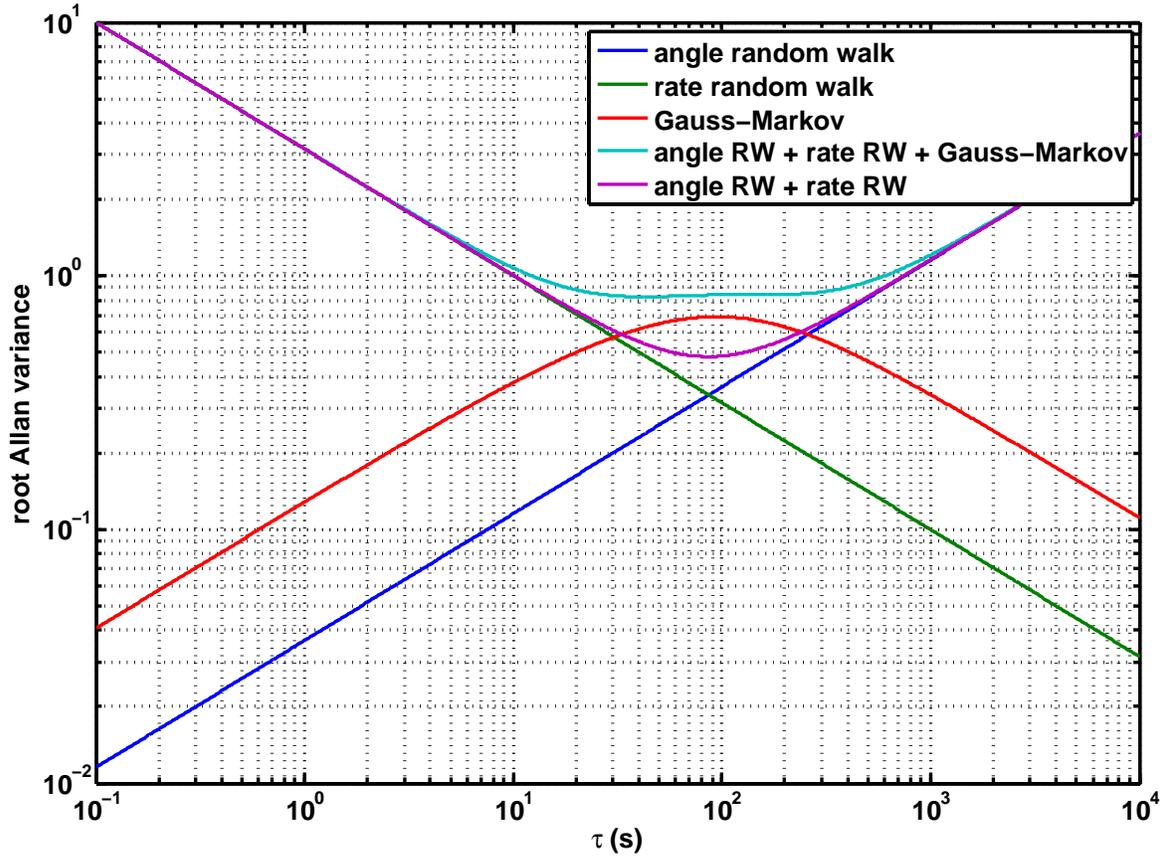


Figure 2.7: Allan variance for a hypothetical gyroscope

bias with an additional parameter. The bias ω_b is written as the sum of the constant part ω_{b0} and the time-varying part ω_{b1} :

$$\omega_b = \omega_{b0} + \omega_{b1} .$$

ω_{b1} is modeled as a Gauss-Markov process:

$$\dot{\omega}_{b1} = -\frac{\omega_{b1}}{\tau} + \nu_{\omega b1}$$

where τ is the time constant and $\nu_{\omega b1}$ the process driving noise. This model contains three parameters and the Allan variance is approximated by three terms as shown in Figure 2.7. We use a special case of this model with $\tau = \infty$ which simplifies the model but still gives a reasonable approximation in the interval of the Allan variance which we considered and which is given in the inertial sensor datasheet used in the implementation as described in Section 5.1.5.3. The approximated Allan variance with $\tau = \infty$ is illustrated in Figure 2.7.

The accelerometer model is of the same form as the gyroscope model:

$$a_m = {}^B a + \nu_a + {}^B a_b$$

where ${}^B a$ is the true specific acceleration, ν_a is the measurement noise and ${}^B a_b$ is the accelerometer bias. Note that the parameter ν_a corresponds to the velocity random walk coefficient N in the Allan variance in Section 2.1.2.1.

The accelerometer bias ${}^B a_b$ can be represented by a sum of two components: a constant one and a varying one. The bias derivative depends on the varying component which is modeled as a random walk as described in Section 2.1.2.1:

$${}^B \dot{a}_b = \nu_{ab} \tag{2.15}$$

Note that the parameter ν_{ab} corresponds to the velocity random walk coefficient K in the Allan variance in Section 2.1.2.1.

Since the IMU consists of a triad of identical accelerometers and a triad of identical gyroscopes and since we consider the inertial sensor noises to be independent white noises with zero mean, we can write for the auto-covariance

$$E(\nu_j(t)\nu_j^T(t+\tau)) = \xi_j^2 I_3 \delta(\tau) \tag{2.16}$$

for $j \in \{a, \omega, ab, \omega b\}$.

2.2.5.2 Camera Noise Model

The marker images are measured in the sensor frame and are corrupted by noise η_y :

$$y_m = \begin{bmatrix} S y \\ SL y \end{bmatrix} + \eta_y .$$

We use the simplest possible noise model which assumes the noise to be white. The camera noise is not actually white but the camera which will be used in the experimental setup in Section 5.1 does not have a datasheet. Thus we do not have any information about its noise characteristics which could permit us to model the noise realistically. Also, we consider that the camera noise model is not critical for the performance of the data fusion algorithms which will be presented in Section 4.3. However, it is important to use a correct noise model for the inertial sensors because these are used in the prediction steps of the data fusion algorithm.

2.2.6 Complete Model

The complete model with noises then reads:

$${}^C \dot{p} = {}^C v \quad (2.17)$$

$${}^C \dot{v} = {}^C G + {}^{BC} q * (a_m - \nu_a - {}^B a_b) * {}^{BC} q^{-1} \quad (2.18)$$

$${}^{BC} \dot{q} = \frac{1}{2} {}^{BC} q * (\omega_m - \nu_\omega - {}^B \omega_b) \quad (2.19)$$

$${}^B \dot{a}_b = \nu_{ab} \quad (2.20)$$

$${}^B \dot{\omega}_b = \nu_{\omega b} . \quad (2.21)$$

The outputs for the right and the left camera:

$${}^S y_i = \frac{f_R}{\langle {}^C m_i, {}^C E_1 \rangle} \left[\frac{\langle {}^C m_i, {}^C E_2 \rangle}{\langle {}^C m_i, {}^C E_3 \rangle} \right] + {}^S u_R \quad (2.22)$$

$${}^{SL} y_i = \frac{f_L}{\langle {}^{CL} m_i, {}^{CL} \tilde{E}_1 \rangle} \left[\frac{\langle {}^{CL} m_i, {}^{CL} \tilde{E}_2 \rangle}{\langle {}^{CL} m_i, {}^{CL} \tilde{E}_3 \rangle} \right] + {}^{SL} u_L \quad (2.23)$$

where $i = 1, \dots, l$. Indices R and L refer to right and left camera resp. (e.g. f_L is the focal distance of the left camera). The measured outputs are:

$$y_{im} = {}^S y_i + \eta_{yi} \quad (2.24)$$

$$y_{(i+l)m} = {}^{SL} y_i + \eta_{y(i+l)} . \quad (2.25)$$

The six accelerometer and gyroscope measurements

$$u = [a_m, \omega_m]$$

are considered as the inputs of our system and the marker images

$$y = \underbrace{[{}^S y_1, \dots, {}^S y_l]}_{S_y} \underbrace{[{}^{SL} y_1, \dots, {}^{SL} y_l]}_{SL_y}$$

as its outputs. y is a vector of length $2l + 2l = 4l$. The state vector which is to be estimated by the data fusion filter in Chapter 4 is of dimension 16 and has the form:

$$x = [{}^C p, {}^C v, {}^{BC} q, {}^B a_b, {}^B \omega_b] .$$

This system is observable with $l = 3$ or more markers. This is a condition for this model to be used in an observer/data fusion algorithm as the ones presented in Chapter 4. See Section 4.3.1 for an observability analysis.

Chapter 3

State of the Art

La présentation de l'état de l'art comporte trois volets: la vision par ordinateur et le tracking optique, le tracking optique-inertiel, et enfin le calibrage.

Dans le volet vision par ordinateur et tracking optique, nous abordons des méthodes pour déterminer la position et l'orientation d'un point ou d'un objet à partir d'images d'un système de caméra monoculaire ou stéréo. Ces méthodes seront utilisées pour le calibrage des caméras et comme référence pour le tracking optique-inertiel.

Des systèmes de tracking utilisant des capteurs optiques et inertiels présentés dans la littérature sont décrits dans la deuxième partie. Ici, nous donnons la motivation de notre approche pour le tracking optique-inertiel et indiquons les différences par rapport aux systèmes existants.

Finalement, nous traitons la question du calibrage des caméras, des capteurs inertiels et du système optique-inertiel.

3.1 Computer Vision and Optical Tracking

Optical tracking uses optical markers attached to an object and one or more cameras which look at these markers. With a camera model and several camera parameters, it is possible to calculate the object position and/or orientation from the marker images, depending on the setup used. The camera model used here is the pinhole model presented in Section 2.1.1.1.

The markers can be active or passive. Active markers send out light pulses. Passive markers are reflecting spheres which are illuminated by light flashes sent from the camera housing [Langlotz, 2004]. The markers are attached to a rigid body which is fixed to the object being tracked. Often, infrared (IR) light and infrared filters are used to simplify marker detection in the images.

Examples of commercially available optical tracking systems are: Optotrak, Polaris (both Northern Digital Inc., Waterloo, Canada [NDI, 2011]), Vicon (Vicon Motion Systems Limited, Oxford, United Kingdom [Vicon, 2011]) and ARTtrack (advanced realtime tracking GmbH, Weilheim, Germany [advanced realtime tracking GmbH, 2011]).

The field of computer vision provides algorithms for calculating an object's position and orientation from the marker images. Here we look into monocular and stereo tracking which use one and two cameras respectively. For monocular tracking, at least four coplanar markers have to be attached to the object to determine the object position and orientation. In stereo tracking, the 3D marker position can be determined from a single marker with a method called triangulation presented in Section 3.1.2.1. In order to calculate an object's orientation, at least 3 non-collinear markers are needed. Calculating the the object position and orientation from the three triangulated markers is called the "absolute orientation problem" and is discussed in Section 3.1.2.2. While the monocular setup is simpler and does not need synchronization between cameras, it only has a poor depth precision. Stereo tracking is the most common setup for optical tracking systems.

Although the optical-inertial tracking system we propose does not use any of these computer vision algorithms which can be computationally complex we study these algorithms here for several reasons. Monocular tracking is the basis of the camera calibration procedure implemented in the Camera Calibration Toolbox [Bouguet, 2010] and used for calibrating the cameras in the experimental setup in chapter 5 which led us to study this method. Wanting to show that the proposed optical-inertial tracking system is more suitable for following fast motion than an optical tracking system, we examined stereo tracking algorithms to calculate pose using only optical data from our experimental setup. This permits us to compare pose estimation from our optical-inertial system to that of a purely optical system in Section 5.3.

3.1.1 Monocular Tracking

With a single camera, it is possible to calculate position and orientation of a rigid body with four non-collinear points lying in a plane. Using less than four markers or four nonplanar markers does not yield a unique solution while five markers in general positions yield two solutions. To obtain a unique solution at least six markers in general positions have to be fixed to the object. The problem of determining an object position depending on the number and configuration of markers has been called the "Perspective- n -point problem" where n is the number of markers and the solution was given in [Fischler and Bolles, 1981].

We use the setup and frame notations as presented in Figure 2.5 and Section 2.2.1. The four marker points are known in marker coordinates and we define the object position as the origin of the marker frame. The position of marker i in camera coordinates then

reads

$${}^C m_i = {}^{MC} R {}^M m_i + {}^{MC} t \quad \text{for } i = 1, \dots, 4$$

where the rotation ${}^{MC} R$ and the translation ${}^{MC} t$ represent the object pose. ${}^M m_i$ is the position of marker i and the ${}^M m_i$ with $i = 1, \dots, 4$ are called the marker model. The goal is to determine the object pose from the camera images ${}^S y_i$.

Images and marker model are related by

$${}^S \tilde{y}_i = K {}^C m_i = K [{}^{MC} R {}^{MC} t]^M \tilde{m}_i$$

where we have used homogeneous coordinates (see Section 2.1.1.1). Since we have a planar rigid body, the marker coordinates read

$${}^M m_i = [{}^M X_i, {}^M Y_i, 0]^T$$

and we can write the homogeneous marker model as

$${}^M \tilde{m}_{i,2D} = [{}^M X_i, {}^M Y_i, 1]^T$$

This makes it possible to calculate the homography matrix H between markers and images:

$${}^S \tilde{y}_i = H {}^M \tilde{m}_{i,2D} . \quad (3.1)$$

The solution presented here first calculates the homography matrix H between markers and images and then determines ${}^{MC} t$ and ${}^{MC} R$ using H and the camera intrinsic matrix K .

Solving for homography H between model and image This calculation follows [Zhang, 2000] for the derivation of the equations and [Hartley and Zisserman, 2003, ch. 4.1] for the solution for H .

We note the homography matrix:

$$H = \begin{bmatrix} h_1^T \\ h_2^T \\ h_3^T \end{bmatrix} .$$

and the homogeneous image:

$${}^S \tilde{y}_i = [u, v, 1]^T .$$

Equation (3.1) gives us:

$${}^S \tilde{y}_i \times H \cdot {}^S \tilde{m}_{i,2D} = \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} \times \begin{bmatrix} h_1^T S \tilde{m}_{i,2D} \\ h_2^T S \tilde{m}_{i,2D} \\ h_3^T S \tilde{m}_{i,2D} \end{bmatrix} = \begin{bmatrix} v_i h_3^T S \tilde{m}_{i,2D} - h_2^T S \tilde{m}_{i,2D} \\ h_1^T S \tilde{m}_{i,2D} - u_i h_3^T S \tilde{m}_{i,2D} \\ u_i h_2^T S \tilde{m}_{i,2D} - v_i h_1^T S \tilde{m}_{i,2D} \end{bmatrix} = 0$$

Since only two of the three equations above are linearly independent, each marker i contributes two equations to the determination of H . We omit the third equation and use $h_i^T s \tilde{y}_{i,2D} = s \tilde{y}_{i,2D}^T h_i$ to rewrite the first two equations:

$$\underbrace{\begin{bmatrix} \tilde{M}_{i,2D}^T & 0 & -u_i \tilde{M}_{i,2D}^T \\ 0 & \tilde{M}_{i,2D}^T & -v_i \tilde{M}_{i,2D}^T \end{bmatrix}}_{A_i} \cdot \underbrace{\begin{bmatrix} h_1 \\ h_2 \\ h_3 \end{bmatrix}}_h = 0$$

Matrix A_i is of dimension 2×9 . The four matrices A_i are assembled into a single matrix A of dimension 8×9 .

The vector h is obtained by a singular value decomposition of A . The unit singular vector corresponding to the smallest singular value is the solution h which then gives matrix H . Note that H is only determined up to a scale.

The estimation of H can be improved by a nonlinear least squares which minimizes the error between the measured image points ${}^S y_i$ and the projection ${}^S \hat{y}_i$ of the marker coordinates using the homography H . The projection reads

$${}^S \hat{y}_i = \frac{1}{h_3^T M \tilde{m}_{i,2D}} \begin{bmatrix} h_1^T M \tilde{m}_{i,2D} \\ h_2^T M \tilde{m}_{i,2D} \end{bmatrix}$$

The cost function which is to be minimized with respect to H is

$$\sum_i \| {}^S y_i - {}^S \hat{y}_i \|^2 .$$

Closed-form solution for pose from homography and intrinsic matrix Having determined matrix H up to a scale, we can write

$$s {}^S \tilde{y}_i = H^M \tilde{m}_{i,2D} \tag{3.2}$$

where s is a scale. The homogeneous image can be expressed as

$${}^S \tilde{y}_i = K \begin{bmatrix} {}^{MC} R & {}^{MC} t \end{bmatrix}^M \tilde{m}_i = K \begin{bmatrix} r_1 & r_2 & r_3 & {}^{MC} t \end{bmatrix} \begin{bmatrix} {}^M X_i \\ {}^M Y_i \\ 0 \\ 1 \end{bmatrix} \tag{3.3}$$

$$= K \begin{bmatrix} r_1 & r_2 & {}^{MC} t \end{bmatrix} \underbrace{\begin{bmatrix} {}^M X_i \\ {}^M Y_i \\ 1 \end{bmatrix}}_{\tilde{m}_{i,2D}} \tag{3.4}$$

where K is the intrinsic matrix. With (3.2) and (3.4) we can then write

$$\lambda \underbrace{\begin{bmatrix} h_1^* & h_2^* & h_3^* \end{bmatrix}}_H = K \begin{bmatrix} r_1 & r_2 & {}^{MC}t \end{bmatrix}$$

where λ is an arbitrary scale and h_j^* are the columns of matrix H . This gives:

$$r_1 = \lambda K^{-1} h_1^* \tag{3.5}$$

$$r_2 = \lambda K^{-1} h_2^* \tag{3.6}$$

$$r_3 = r_1 \times r_2 \tag{3.7}$$

$${}^{MC}t = \lambda K^{-1} h_3^* \tag{3.8}$$

Theoretically, we should have

$$\lambda = \frac{1}{\|K^{-1}h_1^*\|} = \frac{1}{\|K^{-1}h_2^*\|}$$

but this is not true in presence of noise. [DeMenthon et al., 2001] proposes to use the geometric mean to calculate λ :

$$\lambda = \sqrt{\frac{1}{\|K^{-1}h_1^*\|} \cdot \frac{1}{\|K^{-1}h_2^*\|}} \tag{3.9}$$

Matrix ${}^{MC}R = \begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix}$ is calculated according to (3.5)–(3.7) and ${}^{MC}t$ according to (3.8), using the scale λ given in (3.9).

3.1.2 Stereo Tracking

3.1.2.1 Triangulation

Knowing the camera intrinsic parameters, a marker image point can be back-projected into 3D space. The marker lies on the line going through the marker image and the camera center but its position on this line cannot be determined. When two cameras are used, the marker position is determined by the intersection of the two back-projected lines. This concept is called triangulation.

However, these two lines only intersect in theory. In the presence of noise (and due to inaccurate intrinsic parameters), the two back-projected lines will not intersect. Several methods have been proposed to estimate the marker position for this case. The simplest one is the midpoint method [Hartley and Sturm, 1997] which calculates the midpoint of the common perpendicular of the two lines. In the comparison presented in [Hartley and Sturm, 1997] the midpoint method gives the poorest results. This is

explained by the fact that the method is not projective invariant and that the midpoint is not a projective concept. Alternative methods which will be presented in this chapter are linear triangulation and optimal triangulation which minimizes a geometric cost function.

In this section, we want to calculate the position ${}^C m_i$ of marker i in camera coordinates given left and right marker images ${}^{SR} y_i$ and ${}^{SL} y_i$ in right and left sensor coordinates, the camera intrinsic matrices K_R and K_L and the transformation (R_{St}, t_{St}) between left and right camera. The right and left camera matrices can be written as:

$$P_R = K_R \begin{bmatrix} I & 0 \end{bmatrix} = \begin{bmatrix} p_R^{1T} \\ p_R^{2T} \\ p_R^{3T} \end{bmatrix}$$

and

$$P_L = K_L \begin{bmatrix} R_{St} & t_{St} \end{bmatrix} = \begin{bmatrix} p_L^{1T} \\ p_L^{2T} \\ p_L^{3T} \end{bmatrix}.$$

The homogeneous images read:

$${}^{SR} \tilde{y}_i = P_R {}^C \tilde{m}_i = \begin{bmatrix} u_{iR} \\ v_{iR} \\ 1 \end{bmatrix} \quad \text{and} \quad {}^{SL} \tilde{y}_i = P_L {}^C \tilde{m}_i = \begin{bmatrix} u_{iL} \\ v_{iL} \\ 1 \end{bmatrix}.$$

Basic Linear Triangulation Here we describe the homogeneous method given in [Hartley and Zisserman, 2003, ch. 12.2]. The first step is to eliminate the homogeneous scale factor by using the fact that although a projected point and the image are not equal in homogeneous coordinates, they have the same direction and thus their cross product is zero:

$${}^{SR} \tilde{y}_i \times P_R {}^C \tilde{m}_i = 0 \Rightarrow \begin{bmatrix} v_{iR}(p_R^{3T} {}^C \tilde{m}_i) - p_R^{2T} {}^C \tilde{m}_i \\ p_R^{1T} {}^C \tilde{m}_i - u_{iR}(p_R^{3T} {}^C \tilde{m}_i) \\ u_{iR}(p_R^{2T} {}^C \tilde{m}_i) - v_{iR}(p_R^{1T} {}^C \tilde{m}_i) \end{bmatrix} = 0$$

Out of these three equations, only two are linearly independent. We use only the first two equations and write them for both right and left cameras:

$$\underbrace{\begin{bmatrix} u_{iR} p_R^{3T} - p_R^{1T} \\ v_{iR} p_R^{3T} - p_R^{2T} \\ u_{iL} p_L^{3T} - p_L^{1T} \\ v_{iL} p_L^{3T} - p_L^{2T} \end{bmatrix}}_A {}^C \tilde{m}_i = 0$$

We then solve for ${}^C \tilde{m}_i$ by singular value decomposition of A . The unit singular vector corresponding to the smallest singular value is the solution ${}^C \tilde{m}_i$. Since ${}^C \tilde{m}_i$ is a homogeneous vector, bring it into the form $[{}^C m_i^T, 1]^T$ to calculate ${}^C m_i$.

Optimal Triangulation [Hartley and Zisserman, 2003, ch. 12.5] presents a triangulation method which is optimal if the noise is Gaussian. It minimizes the sum of squared distances between image and projected point subject to the epipolar constraint

$${}^{SL}y_i^T F {}^{SR}y_i = 0$$

where F is the fundamental matrix which can be computed from intrinsic matrices K_R and K_L according to [Hartley and Zisserman, 2003, p.246]:

$$F = [K_L t_{Stereo}]_{\times} K_L R_{Stereo} K_R^{-1} / \det(K_L)$$

Two measured stereo image points usually do not satisfy the epipolar constraint due to the presence of noise. The optimal triangulation method corrects the measured points in such a way that they satisfy the epipolar constraint and then triangulates the corrected points using the basic linear method presented above.

Planar Triangulation A special triangulation method [Chum et al., 2005] has been developed for the case when the markers are coplanar. It makes use of the homography matrix between the two cameras. However, this matrix has to be calculated with noise-free image data which is impossible in practice. In consequence, planar triangulation is not treated here.

Comparison of different methods using synthetic data To compare basic and optimal triangulation using synthetic data we proceeded the following way:

1. Generation of a trajectory defined by translation ${}^{MC}t$ and quaternion ${}^{MC}q$ with $n = 214$ frames as shown in Figure 3.3.
2. Computation of 3D marker positions Cm_i from ${}^{MC}t$ and ${}^{MC}q$ using marker coordinates Mm_i for $i = 1 \dots 4$.
3. Projection of 3D marker positions to left and right cameras as in (2.4) using intrinsic matrices K_L and K_R to obtain image coordinates.
 - Test 1: Addition of noise to image coordinates with $\sigma = 1$ pixel
 - Test 2: Addition of error to intrinsic matrices:

$$\tilde{K}_R = \underbrace{\begin{bmatrix} 491 & 1344 & 0 \\ 396 & 0 & 1337 \\ 1 & 0 & 0 \end{bmatrix}}_{K_R} + \begin{bmatrix} 10 & -25 & 0 \\ -5 & 0 & 10 \\ 0 & 0 & 0 \end{bmatrix}$$

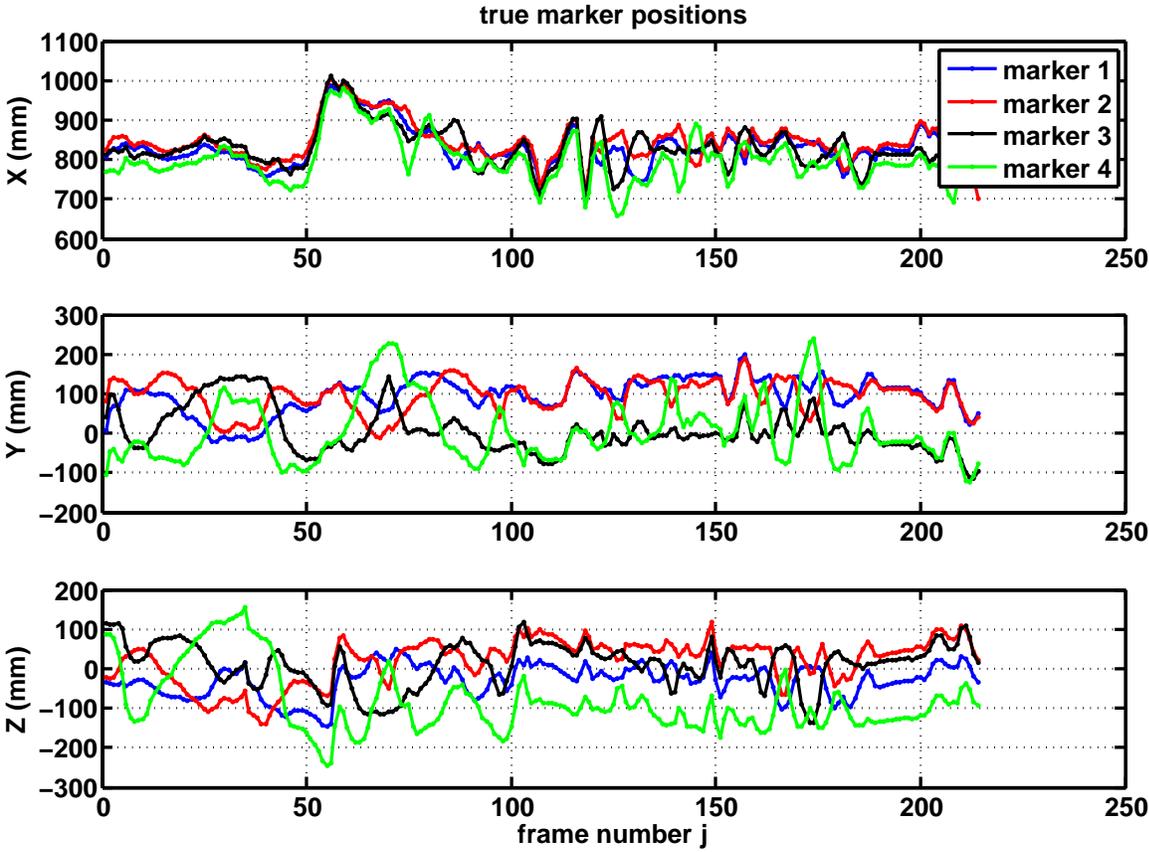


Figure 3.1: synthetic trajectory

	basic	optimal
${}^C m_{1rms}(mm)$	0.8523	0.8522
${}^C m_{2rms}(mm)$	0.8644	0.8644
${}^C m_{3rms}(mm)$	0.8697	0.8697
${}^C m_{4rms}(mm)$	0.8832	0.8829

Table 3.1: RMS for triangulation error with noisy data and correct intrinsic values (test 1)

	basic	optimal
${}^C m_{1rms}(mm)$	9.9640	10.1745
${}^C m_{2rms}(mm)$	9.8547	10.0385
${}^C m_{3rms}(mm)$	10.9958	11.1846
${}^C m_{4rms}(mm)$	11.3301	11.5958

Table 3.2: RMS for triangulation error for noise-free data and incorrect intrinsic values (test 2)

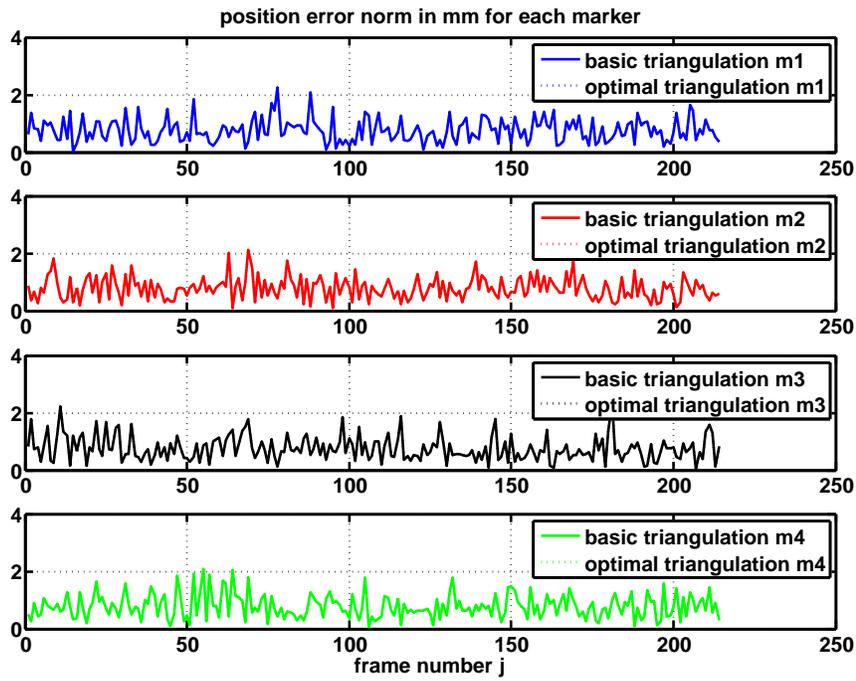
$$\tilde{K}_L = \underbrace{\begin{bmatrix} 491 & 1344 & 0 \\ 396 & 0 & 1337 \\ 1 & 0 & 0 \end{bmatrix}}_{K_L} + \begin{bmatrix} 0 & 30 & 0 \\ 10 & 0 & -20 \\ 0 & 0 & 0 \end{bmatrix}$$

4. Basic and optimal triangulation to obtain estimated marker positions ${}^C \hat{m}_i$.
5. Calculation of RMS (root mean square) for error $x_j = {}^C \hat{m}_{ij} - {}^C m_{ij}$ for each marker i :

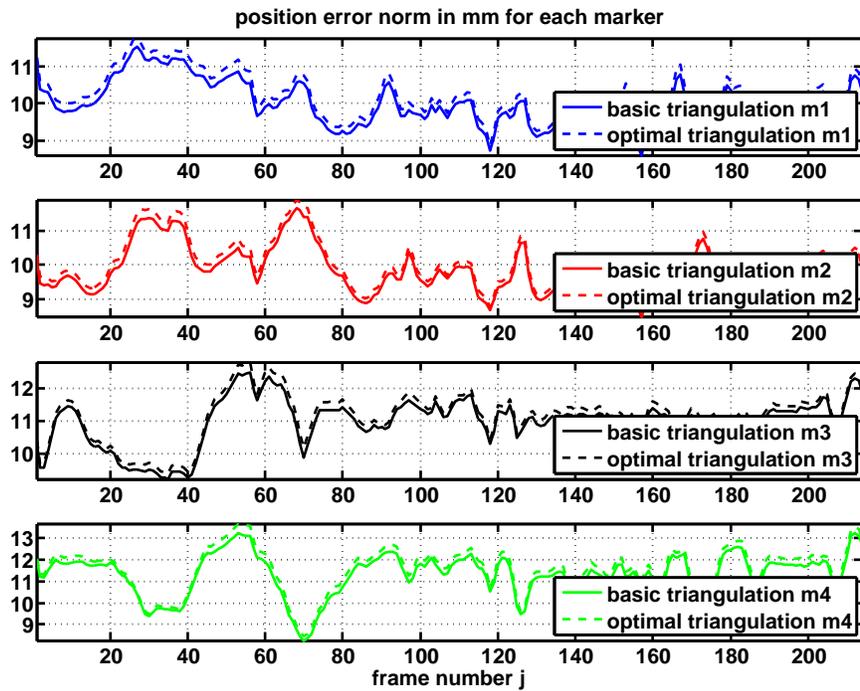
$$x_{rms} = \sqrt{\frac{\sum_{j=1}^n \|x_j\|^2}{n}}$$

Figure 3.2(a) shows the position error norms for both methods for test 1 and Figure 3.2(b) shows those for test 2. The RMS results for test 1 and 2 are listed in Tables 3.1 and 3.2 resp. Note that we use Marker and Camera frames as described in Section 2.2.1. For test 1 with noisy data we see that optimal triangulation only brings a very small improvement over basic triangulation. When using incorrect intrinsic values - which could happen in practice when the cameras have not been calibrated exactly - optimal triangulation even increases the position error. This is probably due to the fact that this method uses the intrinsic values to correct the measured images which obviously demands for correct intrinsic values.

In the light of the results of this simulation of basic and optimal triangulation using synthetic data we decide to use the basic method in the following because the small



(a) test 1



(b) test 2

Figure 3.2: Position error norms for tests 1 and 2

improvement of accuracy it brings does not justify the additional computational load and because it might even deteriorate the estimation when using incorrect intrinsic values.

3.1.2.2 Absolute Orientation Problem

Triangulating marker images gives information about their position only, not about their orientation relative to the camera frame. To determine the orientation of an object, at least three rigidly fixed non-collinear markers are needed. The problem of computing the transformation between two sets of corresponded feature measurements (here markers and their triangulated 3D positions) is called the "absolute orientation problem" [Eggert et al., 1997].

We define the position and orientation of a rigid body to be the position and orientation of the origin of the marker frame in camera coordinates. We can express each marker i in camera coordinates:

$${}^C m_i = {}^{MC}t + {}^{MC}R^M m_i$$

where ${}^{MC}t$ is the translation and ${}^{MC}R$ the rotation between marker and camera frame. These two variables represent the position and orientation of the rigid body and are to be determined by the solution to the absolute orientation problem.

Several methods have been proposed to compute a closed-form solution to calculate the transformation (rotation and translation) between two 3D point sets. In our case, one 3D point set is the marker model ${}^M m_i$ (coordinates of each marker in marker coordinates) and the other 3D point set is the set of 3D marker positions in camera coordinates ${}^C m_i$ (calculated from 2D marker images by e.g. triangulation, see Section 3.1.2.1). The solution then is the position and orientation of the origin of the marker model in camera coordinates, the desired variables ${}^{MC}t$ and ${}^{MC}R$.

In this section, we present three different methods to solve the absolute orientation problem.

Horn's method using unit quaternions This method was presented in [Horn, 1987].

step 1: calculate centroids and normalize vector centroids:

$${}^C \bar{m} = \frac{1}{l} \sum_{i=1}^l {}^C m_i, \quad {}^M \bar{m} = \frac{1}{l} \sum_{i=1}^l {}^M m_i$$

normalize:

$${}^C m'_i = {}^C m_i - {}^C \bar{m}, \quad {}^M m'_i = {}^M m_i - {}^M \bar{m}$$

step 2: calculate matrix S

$$S = \begin{bmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{yx} & S_{yy} & S_{yz} \\ S_{zx} & S_{zy} & S_{zz} \end{bmatrix}$$

with

$$S_{uv} = \sum_{i=1}^l ({}^M m'_i)_u ({}^C m'_i)_v$$

where $({}^M m'_i)_u$ is the u th element of vector ${}^M m'_i$ and $({}^C m'_i)_v$ the v th element of vector ${}^C m'_i$.

step 3: calculate matrix N

$$N = \begin{bmatrix} S_{xx} + S_{yy} + S_{zz} & S_{yz} - S_{zy} & S_{zx} - S_{xz} & S_{xy} - S_{yx} \\ S_{yz} - S_{zy} & S_{xx} - S_{yy} - S_{zz} & S_{xy} + S_{yx} & S_{zx} + S_{xz} \\ S_{zx} - S_{xz} & S_{xy} + S_{yx} & -S_{xx} + S_{yy} - S_{zz} & S_{yz} + S_{zy} \\ S_{xy} - S_{yx} & S_{zx} + S_{xz} & S_{yz} + S_{zy} & -S_{xx} - S_{yy} + S_{zz} \end{bmatrix}$$

step 4: calculate quaternion q

Calculate the eigenvalues and eigenvectors of N . The quaternion q is the eigenvector corresponding to the most positive eigenvalue of N . Then ${}^{MC}q = q$ and ${}^{MC}R = R({}^{MC}q)$.

step 5: calculate scale $s = \frac{\sum_{i=1}^l {}^C m'_i ({}^{MC}R {}^M m'_i)}{\sum_{i=1}^l \|{}^M m'_i\|^2}$

step 6: calculate translation ${}^{MC}t = {}^C \bar{m} - s {}^{MC}R {}^M \bar{m}$

Arun's method by singular value decomposition and Umeyama's strict solution

step 1: calculate centroids and normalize vectors like in step 1 of Horn's method

step 2: calculate matrix H

$$H = \sum_{i=1}^l {}^M m'_i {}^C m'_i{}^T$$

step 3: calculate SVD of $H = U\Lambda V^T$, then $X = VU^T$.

step 4: solution for ${}^{MC}R$

Arun's solution: The result depends on the value of $d = \det(X)$:

$$d = \begin{cases} 1 & \text{then } {}^{MC}R = X \\ -1 & \text{then the solution depends on the singular values of } H. \end{cases}$$

If $\det(X) = -1$ and the k th singular value of H is zero (this corresponds to the case of coplanar point sets), then

$${}^{MC}R = \tilde{X} = \tilde{V}U^T$$

where \tilde{V} is obtained from V by multiplying the k th column by -1 . If more or less than one singular value of H is zero, the algorithm fails.

Umeyama's strict solution:

$${}^{MC}R = VSU^T$$

where

$$S = \begin{cases} I & \text{if } \det(U)\det(V) = 1 \\ \text{diag}(1, 1, \dots, 1, -1) & \text{if } \det(U)\det(V) = -1 \end{cases}$$

which can be expressed [Eggert et al., 1997] as

$$S = \begin{bmatrix} 1 & & \\ & 1 & \\ & & \det(VU^T) \end{bmatrix}$$

step 5: calculate translation ${}^{MC}t = {}^C\bar{m} - {}^{MC}R{}^M\bar{m}$

Walker's method using dual number quaternions Another method for solving the absolute orientation problem is proposed in [Walker et al., 1991]. In contrast to the last two algorithms, this approach can incorporate both positional and directional information and uses weighting factors reflecting data reliability. Here, we present a form of the algorithm which uses only positional data and no weighting factors [Eggert et al., 1997] and is thus comparable to the other two.

A second difference to the other two algorithms is that rotation and translation are estimated in the same step and not separately as in the other two algorithms.

step 1: Calculate position quaternions

A position quaternion is defined in [Walker et al., 1991] as $\tilde{m} = \frac{1}{2} \begin{bmatrix} m \\ 0 \end{bmatrix}$ where m is a 3D position vector. Calculate position quaternions ${}^C\tilde{m}_i$ and ${}^M\tilde{m}_i$.

step 2: Calculate rotation

Calculate matrices

$$C_1 = -2 \sum_{i=1}^l Q({}^C \tilde{m}_i)^T W({}^M \tilde{m}_i)$$

$$C_2 = m * I_m$$

$$C_3 = 2 \sum_{i=1}^l (W({}^M \tilde{m}_i) - Q({}^C \tilde{m}_i))$$

where

$$K(v) = \begin{bmatrix} 0 & -v_2 & v_1 \\ v_2 & 0 & -v_0 \\ -v_1 & v_0 & 0 \end{bmatrix}$$

$$Q(v) = Q \left(\begin{bmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \end{bmatrix} \right) = \begin{bmatrix} v_3 * I_3 + K(v_{0..2}) & v_{0..2} \\ -v_{0..2}^T & v_3 \end{bmatrix}$$

$$W(v) = \begin{bmatrix} v_3 * I_3 - K(v_{0..2}) & v_{0..2} \\ -v_{0..2}^T & v_3 \end{bmatrix}.$$

A vector r is calculated as the eigenvector corresponding to the largest positive eigenvalue of the matrix

$$A = \frac{1}{2} (C_3^T (C_2 + C_2^T)^{-1} C_3 - C_1 - C_1^T).$$

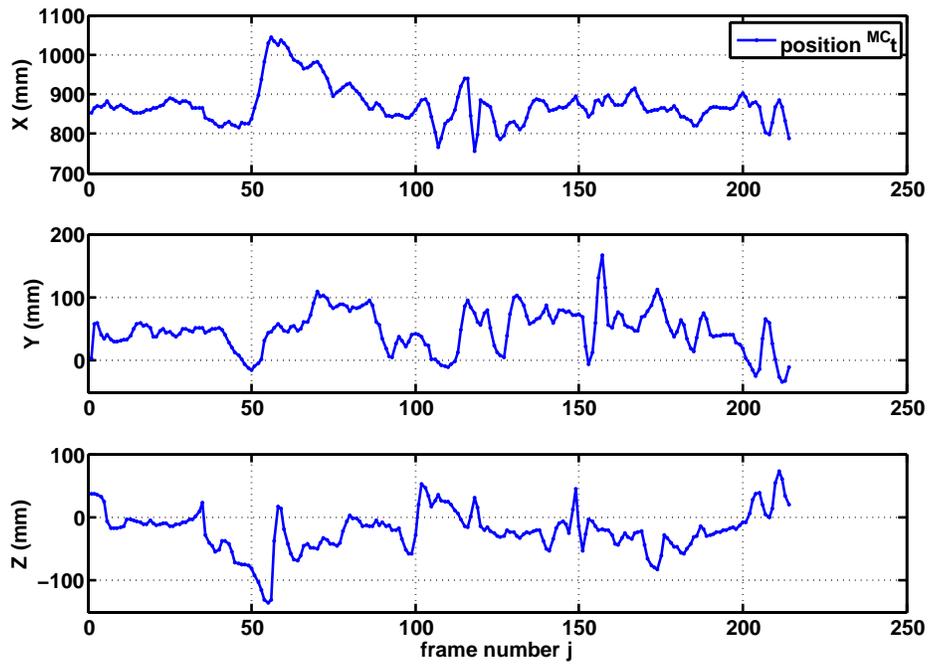
The rotation matrix ${}^{MC}R$ is determined from this vector $r = [r_0, r_1, r_2, r_3]^T$:

$${}^{MC}R = (r_3^2 - r_{0..2}^T r_{0..2}) I_3 + 2r_{0..2} r_{0..2}^T + 2r_3 K(r_{0..2})$$

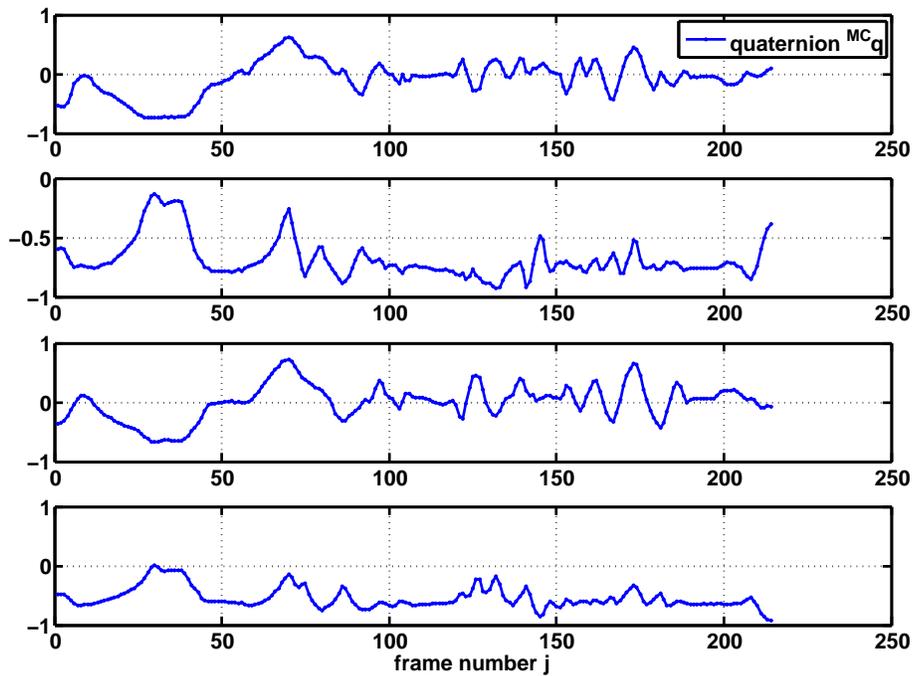
step 3: Calculate translation

Calculate $s = -(C_2 + C_2^T)^{-1} C_3 r$ and then the position quaternion of the translation ${}^{MC}\tilde{t} = W(r)^T s$. Finally, convert to give the translation ${}^{MC}t$.

Comparison of different methods using synthetic data We use the generated trajectory and projected image data from the comparison of triangulation methods in Section 3.1.2.1 and consider the same two cases: in test 1, the image data are corrupted by noise and in test 2, the triangulation uses erroneous intrinsic values. We apply three absolute orientation algorithms (Horn, Umeyama and Walker) and compare the results.



(a) position ^{MC}t



(b) quaternion ^{MC}q

Figure 3.3: synthetic trajectory

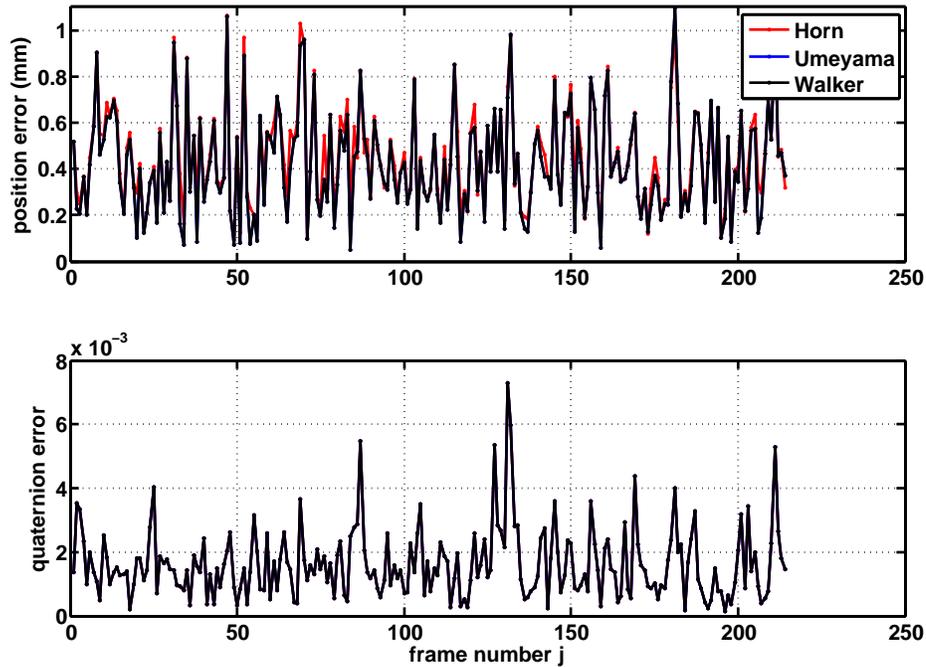


Figure 3.4: Position and quaternion error norm for test 1

	Horn	Umeyama	Walker
${}^{MC}t_{rms}(mm)$	0.4949	0.4770	0.4770
${}^{MC}q_{rms}$	0.0020	0.0020	0.0020

Table 3.3: RMS errors for test 1

The true trajectory defined by translation ${}^{MC}t$ and quaternion ${}^{MC}q$ with n frames is shown in Figure 3.3.

To analyze the results, we calculate the position error $x = {}^{MC}\hat{t} - {}^{MC}t$ and quaternion error $x = 1 - {}^{MC}\hat{q} * {}^{MC}q^{-1}$ for each frame and also the RMS (root mean square) of these errors over all frames:

$$x_{rms} = \sqrt{\frac{\sum_{j=1}^n \|x\|^2}{n}}$$

Figure 3.4 shows the position and quaternion error norm for test 1 and Figure 3.5 displays those for test 2. The RMS errors are listed in Tables 3.3 and 3.4.

These results show that the methods proposed by Umeyama and by Walker give the same results. Horn's method gives a slightly larger position error in both tests but the same quaternion.

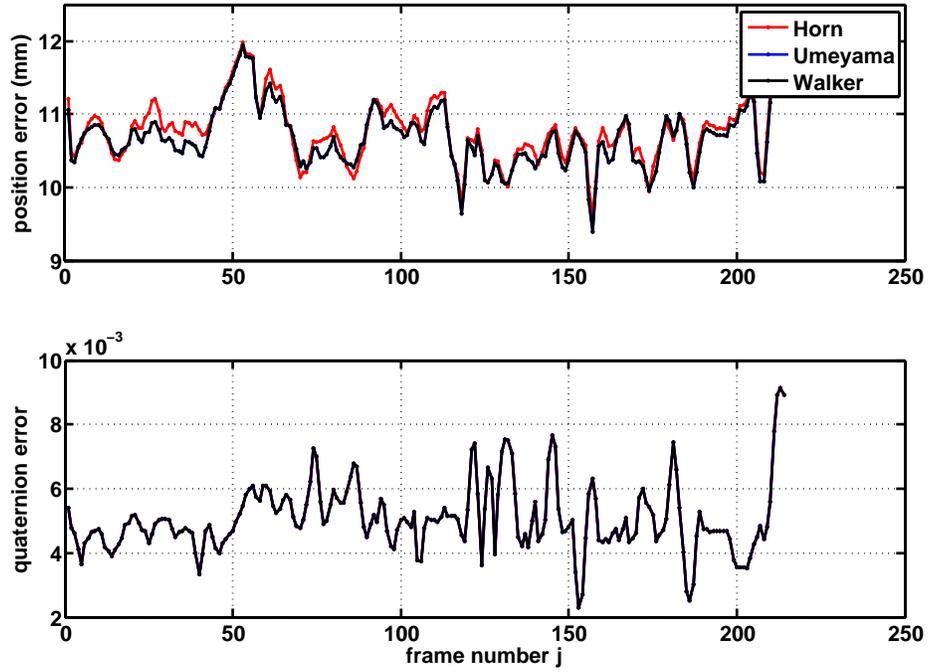


Figure 3.5: Position and quaternion error norm for test 2

	Horn	Umeyama	Walker
${}^{MC}t_{rms}(mm)$	10.7835	10.6980	10.6980
${}^{MC}q_{rms}$	0.0052	0.0052	0.0052

Table 3.4: RMS errors for test 2

In the light of these results, we choose to implement Walker’s method because it gives a smaller position error than Horn’s method and because it is more elegant than Umeyama’s method since it calculates position and quaternion in the same step.

3.1.3 Problems/Disadvantages

Optical tracking suffers from the following drawbacks. The most important one with regard to the surgical tool tracking application considered here is its low bandwidth. Typical optical tracking system used in computer-assisted surgery have a maximum bandwidth of 60Hz which might be lower when multiple objects have to be tracked. This bandwidth is too low to capture rapid human motion. Systems with higher bandwidths exist, for example the Vicon (Vicon Motion Systems Limited, Oxford, United Kingdom [Vicon, 2011]) tracking system which runs at 120Hz, but cannot be used in surgery due to their high cost. Additionally, computer vision algorithms demand complex computations which cause latencies.

Another problem regarding optical tracking is that the object has to always be in the cameras’ field of view. A person or an object blocking the line-of-sight or one marker occluding another leads to complete or partial loss of tracking information.

The tracking performance is affected by noise in the cameras, camera model errors, inaccurate camera calibration, segmentation errors and motion blur which depends on shutter speed [Szeliski, 2011, p.66].

3.2 Optical-Inertial Tracking Systems

3.2.1 Systems Presented in the Literature

In [Parnian and Golnaraghi, 2008] and [Parnian and Golnaraghi, 2010], the system tracks a pen-like tool to which is attached an IMU and whose tip is detected by four cameras placed on a curved line. The cameras are placed in a way as to prevent line-of-sight problems.

[Roetenberg, 2006] presents a setup consisting of a Vicon optical tracking system with 6 cameras and an optical marker attached to an IMU which is fixed to the object being tracked. The system is to be used for offline human motion analysis. Optical data is used to calculate the marker position and inertial data to calculate position, velocity, acceleration and orientation. A Kalman Filter estimates the inertial state errors using the difference between optical and inertial positions. Since the data are analyzed offline, a Kalman filter smoothing algorithm is applied. The inertial sample rate is 100Hz and the optical sample rate is varied from 100Hz to 10Hz. The author finds that in the hybrid system, the optical rate can be lowered to 10Hz and the pose estimation is as good as

the Vicon's at 100Hz. In the case of occlusions, the hybrid estimation is better than the interpolation of optical data with a spline function.

The authors of [Tobergte et al., 2009] combine an optical tracking system with two cameras and three markers with a sample rate of about 55Hz with an IMU being sampled at 500Hz which is attached to the optical markers. They take into account the latency of the optical measurements and find that pose estimation is possible even during short marker occlusions, as long as at least one marker is visible to the cameras.

In [Hartmann et al., 2010], only one camera is used which tracks the position (and the yaw angle when the object is moving) of a fiducial marker pattern which is attached to an IMU being sampled at 400Hz. The camera sample rate is 5 to 10Hz. An EKF is used to fuse optical and inertial data. In simulation, the hybrid system can overcome short-time marker occlusions. Following experiments with a marker pattern and IMU fixed to a test person's wrist, the authors assume that this system can track a moving object and compensate short-time marker occlusions although they do not have proof for this due to a lack of ground truth.

[Korff et al., 2010] presents a tracking system for a handheld surgery tool (which is also presented in [Follmann et al., 2010]) with inertial sensors and an optical tracking system (OTS). Two Unscented Kalman Filters are employed: one for position, using position and orientation as given by an OTS, updated at the corresponding frequency, and one for orientation, using inertial data, updated at inertial frequency.

3.2.1.1 System Setups

The systems presented in the literature differ in the number of markers and the number of cameras. This choice determines the number of degrees of freedom which can be estimated.

The system considered here and the systems cited in Section 3.2.1 all consist of optical markers and inertial sensor rigidly fixed to the moving object and of stationary cameras. Thus they belong to the group of "outside-in" setups. A lot of the early research on optical-inertial data fusion has been carried out on "inside-out" setups in which the inertial sensors and cameras are rigidly attached to the moving object (see for example [You et al., 1999, Rehlinger and Ghosh, 2003, Lobo and Dias, 2003, Bleser and Stricker, 2008]). Most of these systems use only one camera which provides poor precision. Also, the systems in the listed references use visual tracking and detect objects in the surrounding scene which obviously changes both requirements and the mathematical model. Consequently we do not consider inside-out system here.

3.2.1.2 Motivations

Line-of-sight Problem A problem experienced in purely optical tracking systems is the occurrence of occlusions when there is no line-of-sight between camera and markers

which makes pose estimation impossible during the time of the occlusion. Several optical-inertial systems have been presented in the literature to solve this problem by using only inertial data to estimate pose when markers are out of the cameras' line-of-sight for several frames [Parnian and Golnaraghi, 2010, Hartmann et al., 2010]. Roetenberg compares optical-inertial tracking to optical tracking and shows that in the case of short-time marker occlusions, the optical-inertial motion estimate is better than the interpolation of optical data with a spline function [Roetenberg, 2006].

Reduction of Optical Sample Rate In some applications, inertial sensors are added to an optical tracking system with the goal of lowering the optical sample rate, thus saving processing time compared to optical tracking [Parnian and Golnaraghi, 2008, Roetenberg, 2006].

The authors of [Parnian and Golnaraghi, 2008] present a system which samples the IMU at 100Hz and the cameras at 20Hz and they find that with this setup it is possible to lower the camera sample rate to 2Hz. This saves processing time compared to purely optical tracking systems.

In [Roetenberg, 2006], an optical sample rate of 10Hz and an IMU sample rate of 100Hz are used. With this configuration, the system accuracy is found to be as good as the Vicon's at 100Hz.

Augmentation of Bandwidth Inversely to the previous approach of reducing the optical sample rate, optical-inertial tracking is used in several cases to augment the bandwidth of optical tracking by adding inertial sensors and fusing the different sensor data. Thanks to the high sample rates of inertial sensors, the system bandwidth can be increased. This is the main motivation for the present work as well as for the systems in [Tobergte et al., 2009, Hartmann et al., 2010].

3.2.1.3 Applications

Optical-inertial tracking systems are commonly used for human motion tracking. Applications include gait analysis [Roetenberg, 2006], biomechanical research, virtual reality and motion capture for animation.

In augmented reality, inside-out systems are used for head-mounted displays (HMD) [Bleser and Stricker, 2008].

Object tracking with optical and inertial sensors is done for handheld tools, for example of surgery tools [Tobergte et al., 2009], and for industrial tools [Parnian and Golnaraghi, 2010].

[Hartmann et al., 2010] proposes a general indoor tracking system and studies the example of hand tracking in the presented experiment.

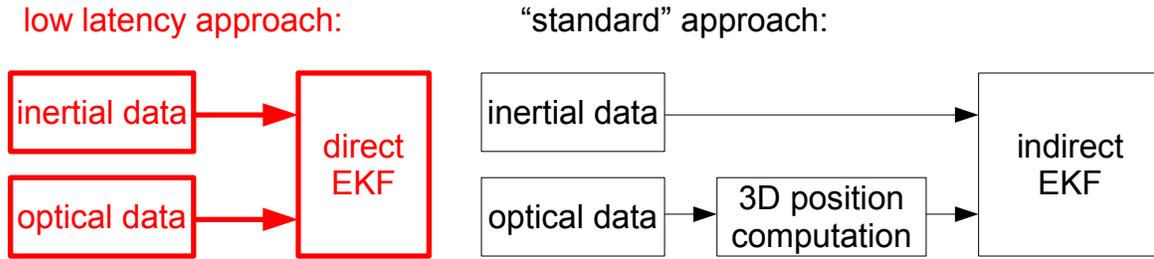


Figure 3.6: Diagram for direct and indirect approach

3.2.2 Motivation of Our Approach

The intended use of our optical-inertial tracking system for servo-controlling a handheld tool in a computer-assisted surgery system imposes several choices to meet the requirements stated in Section 1.2. In order to be compatible with existing optical tracking systems, we use stereo cameras. The use of three optical markers and an three-axis IMU is necessary to obtain the pose with six degrees of freedom. The optical bandwidth is augmented through the use of an IMU with a high bandwidth (at least 200Hz) and of a data fusion algorithm running at the inertial frequency.

The most important difference to existing approaches is the use of optical data directly as inputs to the data fusion algorithm instead of triangulated 3D data. This should reduce latency because it omits the computationally complex computer vision algorithms necessary for obtaining 3D data. It also prevents the optical marker position estimation errors from being propagated in the data fusion algorithm. We call our approach "direct" as opposed to the "indirect" one using 3D data.

Both approaches are shown in Figure 3.6. Among the systems presented in Section 3.2.1, [Hartmann et al., 2010, Korff et al., 2010] and [Tobergte et al., 2009] are similar to our setup. They all use an indirect data fusion approach.

To further reduce the data fusion algorithm's complexity, we will develop algorithms in Chapter 4 which take into account system symmetries.

Our optical-inertial tracking system will have a high bandwidth (due to the inertial sensors) and a small latency (due to the direct approach and suitable algorithm which reduces computational complexity) and will thus be suitable for servo-controlling a handheld tool, in contrast to existing tracking systems which run with a low bandwidth and/or an important latency or do not meet the requirements to be used in an operating room.

3.3 Calibration

Mathematical system models generally contain a number of parameters which describe, for example, scale factors, coordinate transformations and biases. The parameters can either be measured, obtained from calibration or from a sensor datasheet.

Using sensors also demands knowledge of several parameters. First of all, the sensor readings have to be converted into physical measurements. Secondly, when employing several sensors their relative positions and orientations and possible interactions have to be known.

For the system presented in Chapter 2, we have to determine the camera intrinsic values and misalignment and scale factors for the triaxial inertial sensors. Calibration procedures for these cases are described in Sections 3.3.1 and 3.3.2. Since we are using two different kind of sensors, optical and inertial ones, we also have to calibrate this hybrid setup. Calibration of optical-inertial systems is studied in Section 3.3.3.

Calibrating the sensors and other parameters in the system model is important in this work because the sensor measurements and the system model will be used in the data fusion algorithms for the optical-inertial tracking system. If the parameters used in the model differ from their true values, this will have an impact on the data fusion algorithm. This explains why special care has to be taken to calibrate the optical-inertial tracking system.

3.3.1 Camera Calibration

Camera calibration methods determine camera intrinsic parameters which are used in the pinhole model as described in Section 2.1.1.1. These parameters are the focal length f , the aspect ratio a (relation between pixel length and height) and the principal point u . It is also possible to estimate distortion but this parameter will not be treated here.

A widely used method is the one proposed by Zhang [Zhang, 2000] using a planar calibration pattern which can be a chessboard pattern printed out with an ordinary printer. It uses the fact that with a camera and a planar target with at least four non-collinear points it is possible to calculate the pattern position and orientation (see Section 3.1.1). Since this method uses only a very simple pattern it is much easier to use than other methods using 3D calibration objects. Zhang's method is implemented in the Camera Calibration Toolbox for Matlab [Bouguet, 2010]. The cameras used in the implementation in Chapter 5 are calibrated using this toolbox.

3.3.1.1 Stereo Calibration

When using a stereo camera rig, the transformation (rotation R_{St} and translation t_{St}) between the rigidly fixed cameras

$${}^C p = R_{St} {}^L p + t_{St}$$

has to be known. It can be calculated using a stereo calibration method, for example the one implemented in the Camera Calibration Toolbox for Matlab [Bouguet, 2010] which is the one used in Chapter 5 for the experimental setup.

3.3.2 IMU Calibration

We consider the calibration of a triad of accelerometers and of a triad of gyroscopes. These calibration procedures usually estimate the following parameters: biases and scale factors (also called sensitivities or gains) for accelerometers and gyroscopes and additionally axis misalignments for accelerometers.

For gyroscopes, it is usually sufficient to know and compensate only the biases among the above mentioned parameters. This is the case for our optical-inertial tracking system which is why we will not study gyroscope calibration methods here. The biases can be determined by averaging the gyroscope measurements over several second for a static position. Since in this case the angular velocities are zero, the measured values correspond to the gyroscope biases.

Traditionally, IMUs have been calibrated using mechanical platforms, for example a turntable, which turns the IMU with known velocity into known orientations. This approach does not seem appropriate for low-cost MEMS IMUs because in this case the calibration would be more expensive than the sensor itself. Instead, other accelerometer calibration procedures have been developed which do not need a turntable. All the accelerometer calibration procedures presented here use the fact that the sensed specific acceleration in a static position should be equal to gravity. Neither of these methods need any additional material. We will compare them here with regard to the data collection process.

One group of methods demands the IMU to be placed in several different orientations and then kept stationary during data collection. In [Skog and Händel, 2006] the IMU is placed by hand in 18 different positions (the 6 sides and 12 edges of the IMU) and 100 samples are collected for each orientation. The cost function is the sum over all samples of the norm of the difference of the measured acceleration and $1g$. The estimated parameters are scale factors, biases and misalignments. The calibration in [Fong et al., 2008] is similar to the one in [Skog and Händel, 2006], but additionally estimates cross-axis sensitivities. The authors propose two different procedures. For the first one, the IMU is placed in 18 different positions (IMU resting on its 6 flat surfaces and 12 edges). A quasi-static

detector chooses data parts for which the IMU was static. The static signals are averaged over 1s to reduce noise. In the second procedures, the IMU is placed on the head or hand of a non-technical user and is moved and paused at least 24 times. A quasi-static detector indicates parts where the sensors were static for 0.25s. In both procedures, the data are used to compute a cost function which measures the deviation from $1g$ for each position which is then minimized to calculate the calibration parameters. The procedure in [Panahandeh et al., 2010] estimates the same parameters as in [Skog and Händel, 2006] but also two Euler angles for each of the stationary positions. The cost function is again equal to the sum over all samples of the norm of the difference of the measured acceleration and $1g$. It is first minimized w.r.t. the Euler angles and then w.r.t. to the parameters. A simulation for 9 positions with 25 samples each showed that the square-root of the empirical mean square error converges to the square-root of the Cramer-Rao bound which is the lower limit for the estimation error [Panahandeh et al., 2010] after a few measurements.

A different method is proposed in [Lötters et al., 1998] for which data can be collected while the sensors are in use. A quasi-static moments detector discards all samples for which the variance of the norm of the measured acceleration vector is above a certain threshold. The sensor model which is a nonlinear function of the parameters is linearized around the previous parameter estimation. The cost function for the estimation of the bias and scale factor parameters is again the difference of the norm to $1g$, using the linearized sensor model. This method has been developed for the case of movement monitoring of a patient wearing an IMU attached to his lower back. Simulations and experimental data show that the best results are obtained when the IMU is placed in such a way that gravity is equally distributed over the three axes. The authors also observe that accuracy is increased the more different orientations are used.

With regard to our optical-inertial tracking system, the first group of methods might be difficult to apply when the Sensor Unit is attached to a handheld tool. It also seems improbable to have the scenario necessary for the in-use method in [Lötters et al., 1998] in our application.

A method with less constraints on the data collection process is presented in [Dorveaux et al., 2009]. It only demands the IMU to be moved slowly into different orientations, ideally covering the whole sphere of possible orientations. N accelerometer measurements y_i with $i = 1 \dots N$ are collected. The actual value of the acceleration is expressed as

$$Y_i = Ay_i + B \tag{3.10}$$

where the matrix A represents scale factors and misalignment terms and B is the bias vector. Since the IMU moves very slowly, we can consider that the acceleration norm is always equal to $1g$, i.e. $\|Y_i\| = g$. The authors propose to estimate the parameters by iterations of least square problems. The following cost function is formulated for the k th

iteration:

$$h(A, B, k) = \sum_{i=1}^N \left\| Ay_i + B - g \frac{y_{i,k}}{\|y_{i,k}\|} \right\|^2 \quad (3.11)$$

The least-squares minimization of this cost function with respect to A and B gives (A_{k+1}, B_{k+1}) which are used to update the data vectors:

$$y_{i,k+1} = A_{k+1}y_{i,k} + B_{k+1} \quad (3.12)$$

After K iterations, the so-called "calibrated" data vectors are obtained:

$$y_{i,K} = \tilde{A}_K y_{i,k} + \tilde{B}_K \quad (3.13)$$

where \tilde{A}_K and \tilde{B}_K contain the estimated parameters; they have been determined recursively by

$$\tilde{A}_k = A_k \tilde{A}_{k-1} \quad (3.14)$$

$$\tilde{B}_k = A_k \tilde{B}_{k-1} + B_k \quad (3.15)$$

This method is used in the implementation in Chapter 5 because of its simple data collection process.

3.3.3 Optical-Inertial Calibration

When different kind of sensors are used in a setup, the transformations between the sensor frames has to be known. In many cases, the transformation cannot be measured directly but has to be determined by some kind of calibration procedure. For optical-inertial systems, this transformation is between IMU and camera for inside-out systems and between IMU and markers for outside-in systems.

3.3.3.1 IMU-Camera for inside-out systems

In an inside-out system, the transformation between the rigidly connected IMU and camera has to be determined. All the methods presented here consider monocular systems.

The most demanding procedure in regard to the data collection and hardware requirements was proposed in [Lobo and Dias, 2007]. The translation is estimated using a turntable by which the system can be turned about the IMU origin into several static poses and then applying a variant of the hand-eye calibration [Tsai and Lenz, 1989]. The rotation is determined from several measurements when the camera is looking at a vertical chessboard pattern in different poses. The problem then corresponds to finding the rotation between two sets of vectors as in [Horn, 1987].

Two methods which do not need additional hardware except for a chessboard pattern are presented in [Hol et al., 2008] and [Mirzaei and Roumeliotis, 2008]. The latter estimates the translation and rotation as additional states in the Kalman Filter developed for tracking the sensor position, velocity and orientation. The former first estimates the same parameters in a Kalman Filter and uses these values as a predictor for a least squares minimization. Neither of these methods require specific positions or trajectories during data collection but a motion which sufficiently excites the different sensors.

A joint calibration of IMU-camera transformation and of IMU parameters (scale factors, biases and misalignments) with a recursive Sigma-Point Kalman Filter was proposed in [Zachariah and Jansson, 2010]. This method was only tested in simulation with a generated trajectory with continuous accelerations.

3.3.3.2 IMU-Marker for outside-in systems

[Kim and Golnaraghi, 2004] presents a calibration procedure which estimates several IMU parameters (biases, scale factors, misalignments) and the rotation between the IMU and several optical markers which are rigidly connected. An optical tracking system provides position and orientation measurements of the optical markers while the system is rotated and accelerated manually. The optical measurements are derived twice to calculate the IMU's angular velocities and specific accelerations. The difference between these values and the velocities and accelerations measured by the IMU equals the cost function which is minimized with a least squares approach. The IMU-marker translation was not estimated in the calibration because the authors consider it is "not sensitive to millimeter-level accuracy" and thus use the design specifications from manufacturing.

As mentioned earlier, calibration of our optical-inertial system is important to ensure good performance, but few calibration procedures exist for this setup. It might be possible to adapt methods for inside-out systems can be adapted to the outside-in case, but the method in [Lobo and Dias, 2007] demands for a precise data collection which is not possible for our setup. The method in [Mirzaei and Roumeliotis, 2008] is implemented in Section 5.4.4 but does not converge either with synthetic or real data. Instead we propose a novel method Section 4.4.2 which uses estimations from the optical-inertial data fusion algorithm to determine the rotation between the optical marker frame and the body frame of the IMU. After having calibrated this rotation, we apply a method similar to the one in [Kim and Golnaraghi, 2004] using optical data as reference to determine the translation between marker and body coordinates. The Sensor Unit calibration procedure for the experimental optical-inertial setup is described in Section 5.4.4.

Chapter 4

Data Fusion

Le coeur de notre système optique-inertiel est un algorithme de fusion de données qui intègre les données des différents capteurs pour déterminer la position et l'orientation d'un objet. Il utilise les informations complémentaires des capteurs pour donner une estimation avec une grande bande passante. L'algorithme prend la forme d'un observateur dont nous rappelons le fonctionnement. Un type d'observateur pour les systèmes nonlinéaires est le filtre de Kalman étendu. Ce filtre est présenté dans la deuxième partie du chapitre.

La troisième partie présente des filtres pour notre système optique-inertiel. Nous commençons avec une première modification du filtre de Kalman étendu standard qui respecte la géométrie des quaternions qui sont utilisés pour représenter l'orientation. Ensuite, nous proposons deux autres modifications: le filtre de Kalman étendu invariant à droite et celui invariant à gauche. Ces deux filtres prennent en compte les symétries du système. Le système est invariant par rapport à une rotation et il est logique d'utiliser un filtre qui soit aussi invariant par rapport à cette rotation.

Le calibrage est un point important pour la performance de notre système de tracking. Dans la dernière partie de ce chapitre, nous étudions d'abord l'influence de différents erreurs paramétriques sur l'estimation du filtre de Kalman invariant à droite et proposons ensuite une méthode de calibrage de la rotation entre les repères marqueurs et body qui utilise les estimations du filtre.

4.1 Motivation

The optical-inertial system presented in Chapter 2 measures data from two types of sensors. The optical sensors output image data from which the object's absolute position and orientation can be determined. The bandwidth is usually too low to follow fast human motion. Inertial sensors measure angular velocities and specific accelerations at a high bandwidth which is suitable for tracking fast human motion. However, when using inertial

data to calculate the object position and orientation, these are always relative to an initial position and will also drift with time due to sensor errors which cannot be compensated for. The fact that optical and inertial sensors give complementary information leads to the idea of using both in a hybrid optical-inertial tracking system.

When using data from different types of sensors, these data are integrated in a so called data fusion algorithm to calculate the desired variables. These kind of algorithms demand for a mathematical model of the system dynamics and output:

$$\begin{aligned}\dot{x} &= f(x, u) \\ y &= h(x, u)\end{aligned}$$

where x is the state, u the input and y the output. The algorithm usually takes the form of an observer. An observer is a filter which estimates the system state x using the known input u and the known output y [Besancon, 2007] and which makes the estimated state \hat{x} converge to the real state x . In its most common form, an observer consists of a copy of the system dynamics plus a correction term depending on the measured output:

$$\dot{\hat{x}} = f(\hat{x}, u) - K(y - h(\hat{x}, u))$$

where \hat{x} is the estimated state and y is the measured output. K is the matrix observer gain which has to be determined and which can be constant or depend on y , \hat{x} , u and/or time.

To design an observer, the considered system must be observable, i.e. it must be possible to recover information on the system state from the measured output and known input [Besancon, 2007]. Conditions for observability are given in [Besancon, 2007].

In the mathematical model for the optical-inertial tracking system as described in Section 2.2.6, the system dynamics describe the Sensor Unit motion. Inertial measurements are considered to be the system inputs and the optical measurements its outputs.

A common algorithm for data fusion is the Kalman Filter for linear systems and the Extended Kalman Filter (EKF) for nonlinear systems. It has the form of the observer given above and calculates the filter gain as well as the estimation uncertainty at each time step. Introductions to Kalman Filtering can be found in [Simon, 2006, Grewal et al., 2001, Farrel and Barth, 1998, Groves, 2008], for example.

We have chosen an Extended Kalman Filter as a data fusion algorithm for our optical-inertial tracking system. To adapt the filter to our system and to improve its performance, we make several modifications which take into account, and make use of, the system geometry.

The general equations for the continuous EKF are presented in Section 4.2.1. The next Section presents the first modification we have to make in the EKF when one of the states to be estimated is an quaternion: this gives the Multiplicative EKF which respects the

quaternion geometry. Section 4.2.3 presents a continuous-discrete form of the EKF which takes into account that the system dynamics are continuous but the output measurements discrete.

Section 4.3 treats data fusion for our particular application of an optical-inertial tracking system. It begins with an observability analysis to show that the system state is indeed observable. We then explain the choice of the system output which distinguishes our system from other optical-inertial tracking systems. Our approach is a direct one which saves processing time compared to the standard indirect approach. We then continue by giving three different versions of the EKF: an MEKF, a Right-Invariant EKF and a Left-Invariant EKF. The invariant EKFs take into account system symmetries which are studied in Section 4.3.4.2. For each of these filters, we give the filter equation and then analyse an error system to calculate the Kalman Filter gain and propagate estimation uncertainties. Finally, we give the continuous-discrete multi-rate version of the Right-Invariant EKF which takes into account the different sample rates of optical and inertial sensors.

The system model we use in the data fusion algorithms for the optical-inertial tracking system contains several parameters which have been measured, obtained from calibration or from a datasheet, such as the camera's intrinsic values, the marker model and the rotation between camera and world frames. The parameters used in the model might differ from their true values which will have an impact on the data fusion algorithm. In Section 4.4 we study the influence of parameter errors on the estimated state and propose a calibration procedure which uses RIEKF estimates to determine the rotation between body and marker frames.

4.2 Extended Kalman Filter

4.2.1 Continuous EKF

We consider a system

$$\dot{x} = f(x, u) + M\nu \tag{4.1}$$

$$y = h(x, u) + N\eta \tag{4.2}$$

where ν and η are independent white noises. M is of dimension $n \times r$ and N of dimension $s \times s$ where n is the number of states, r is the number of input noises and s is the number of outputs. The input u and the output y are known. x is the state which is to be estimated.

The EKF calculates an estimate $\hat{x}(t)$ of the state $x(t)$ according to

$$\dot{\hat{x}} = f(\hat{x}, u) - K(y - h(\hat{x}, u)) \quad (4.3)$$

$$K = PC^T R^{-1} \quad (4.4)$$

$$\dot{P} = AP + PA^T + Q - PC^T R^{-1} CP \quad (4.5)$$

where $A = \partial_1 f(\hat{x}, u)$ and $C = \partial_1 h(\hat{x}, u)$. In this notation, ∂_i is the partial derivative with respect to the i th argument. P is the estimate error covariance. The matrices Q (of dimension $n \times n$) and R (of dimension $s \times s$) depend on \tilde{Q} and \tilde{R} through

$$Q = M\tilde{Q}M^T \quad (4.6)$$

$$R = N\tilde{R}N^T \quad (4.7)$$

where \tilde{Q} contains the covariance parameters of ν and \tilde{R} those of η .

The estimation error $\Delta x = \hat{x} - x$ satisfies up to higher-order terms the linear equation

$$\Delta \dot{\hat{x}} = (A - KC)\Delta x - M\nu + KN\eta. \quad (4.8)$$

Recall that when A and C are constant, e.g. around a steady state point, the EKF converges when (A, M) is stabilisable and (A, C) detectable [Goodwin et al., 2001, pp. 704], [Wonham, 1968].

4.2.2 Multiplicative EKF (MEKF)

When a quaternion has to be estimated, care has to be taken to preserve its unit norm. This is not the case with the standard correction term $K(y - h(\hat{x}, u))$ because it does not respect Equation (2.6). The differential equation for the quaternion reads

$$\dot{q} = \frac{1}{2}q * \omega$$

where ω is the angular rate. With the standard correction time, we would have the following equation for the estimated quaternion:

$$\dot{\hat{q}} = \frac{1}{2}\hat{q} * \omega - K(y - h(\hat{x}, u))$$

which does not respect (2.6) for unit quaternion differential equations.

The so-called Multiplicative EKF (MEKF) [Lefferts et al., 1982, Crassidis et al., 2007] respects the geometry of the quaternion space and preserves the quaternion unit norm. The first possible correction term for the quaternion differential equation is $\hat{q} * K_q(y - h(\hat{x}, u))$. The quaternion differential equation then reads

$$\dot{\hat{q}} = \frac{1}{2}\hat{q} * \omega - \hat{q} * K_q(y - h(\hat{x}, u)) = \hat{q} * \left(\frac{1}{2}\omega - K_q(y - h(\hat{x}, u)) \right).$$

The second possible correction term is $K_q(y - h(\hat{x}, u)) * \hat{q}$ which yields for the quaternion differential equation:

$$\dot{\hat{q}} = \frac{1}{2} \hat{q} * \omega - K_q(y - h(\hat{x}, u)) * \hat{q} .$$

Both equations respect (2.6) and thus preserve the quaternion unit norm.

Also, the standard linear error $\hat{q} - q$ in the error system analysis does not make sense for quaternions. A multiplicative error like $\hat{q}^{-1} * q$ or $\hat{q} * q^{-1}$ has to be used instead.

4.2.3 Continuous-Discrete EKF

The Extended Kalman Filter exists in a continuous - as presented in Section 4.2.1 - and in a discrete form. However, in many cases the model is continuous while the output measurements are discrete. In this case, a continuous-discrete form of the EKF can be used. This can also be applied to systems with several different sample rates; if one sample rate is much higher than the others it can be considered to be continuous.

In the continuous-discrete EKF, the filter equations are split up into two parts: one which uses the continuous model and continuous input measurements and one which uses the discrete output measurements. The continuous step is called prediction and the discrete one correction or measurement update. Both steps output an estimated state: the prediction calculates an "a priori" state and the correction an "a posteriori" state.

In the prediction step, the continuous a priori state and matrix P^- are calculated according to

$$\dot{\hat{x}}^-(t) = f(\hat{x}^-(t), u(t)) \tag{4.9}$$

$$\dot{P}^-(t) = AP(t) + P(t)A^T + Q \tag{4.10}$$

where $A = \partial_1 f(\hat{x}^-(t), u(t))$ and $Q = M\tilde{Q}M^T$.

For the correction step at a discrete time t_k , sampled variables $P_k^- = P^-(t_k)$, $\hat{x}_k^- = \hat{x}^-(t_k)$ and $y_k = y(t_k)$ are used. The estimated a priori state and matrix P are corrected to give an a posteriori state according to

$$\hat{x}_k = \hat{x}_k^- - K(y_k - h(\hat{x}_k^-, u_k)) \tag{4.11}$$

$$P_k = (I - KC_k)P_k^- . \tag{4.12}$$

with $C_k = \partial_1 h(\hat{x}_k^-, u_k)$ and $K = P_k^- C_k^T (C_k P_k^- C_k^T + R_d)^{-1}$. Note that for discrete measurements we have set

$$R_d = N\tilde{R}_d N^T . \tag{4.13}$$

The discrete \tilde{R}_d is related to the continuous \tilde{R} by $\tilde{R}_d = \tilde{R}/\Delta t$ where Δt is the time between two discrete measurements [Simon, 2006, p. 232].

Figure 4.1 shows a diagram for the continuous-discrete EKF.

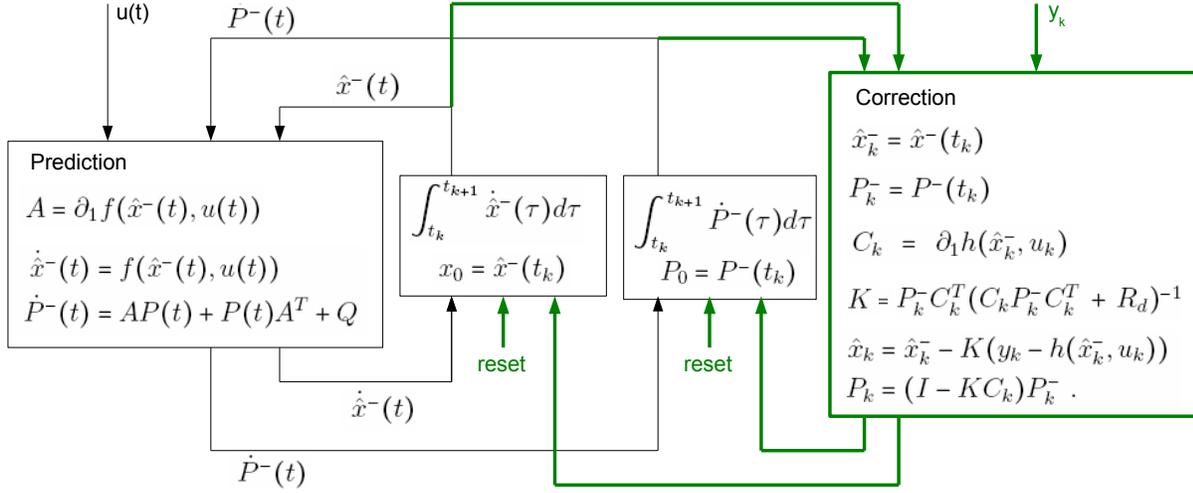


Figure 4.1: continuous-discrete EKF diagram

4.3 Data Fusion for Optical-Inertial Tracking

4.3.1 System Observability

To design a data fusion algorithm which takes the form of an observer as presented in Section 4.1, the system has to be observable [Besancon, 2007]. Otherwise it would not be possible to estimate the entire state.

The system is observable if the system states can be expressed as a function of the inputs, outputs and their derivatives. We will now analyze the observability of the considered optical-inertial system and show that the system is indeed observable and especially that the inertial sensor biases have been placed in a way such that they are observable. Note that we neglect the noise terms in this analysis.

The ${}^C m_i$ can be determined as a function of the output y via triangulation [Hartley and Sturm, 1997]. A solution to the absolute orientation problem, e.g. [Walker et al., 1991], calculates position ${}^C p$ and quaternion ${}^{BC} q$ using the ${}^C m_i$. Hence, ${}^C p$ and ${}^{BC} q$ can be expressed as a function of the output:

$$[{}^C p^T, {}^{BC} q^T] = \zeta_1(y) . \quad (4.14)$$

According to (2.17), ${}^C v = {}^C \dot{p}$ and since ${}^C p$ is a function of y ,

$${}^C v = \zeta_2(y, \dot{y}) . \quad (4.15)$$

Equation (2.18) gives

$${}^B a_b = a_m - {}^{BC} q^{-1} * ({}^C \dot{v} - {}^C G) * {}^{BC} q$$

which can be written as

$${}^B a_b = \zeta_3(u, y, \dot{y}, \ddot{y})$$

thanks to (4.14) and (4.15). Finally, (2.19) can be transformed to

$${}^B \omega_b = \omega_m - 2{}^{BC} q^{-1} \star {}^{BC} \dot{q}$$

which yields, thanks to (4.14):

$${}^B \omega_b = \zeta_4(u, y, \dot{y}) .$$

All the states can be expressed as a function of the inputs, outputs and their derivatives and hence the system proposed in Section 2.2.6 is observable.

4.3.2 Direct and Indirect Approaches

The direct approach uses inertial and optical (2D marker images) data directly in the EKF. The output model is the one used here as given in Section 2.2.4:

$${}^C y_i = \frac{f}{\langle {}^C m_i, {}^C E_1 \rangle} \begin{bmatrix} \langle {}^C m_i, {}^C E_2 \rangle \\ \langle {}^C m_i, {}^C E_3 \rangle \end{bmatrix}$$

with

$${}^C m_i = {}^C p + {}^{BC} q \star {}^B m_i \star {}^{BC} q^{-1} .$$

This is the output which will be used for all of the data fusion algorithms in the following Sections.

This direct approach which uses 2D optical data as measurements in the EKF is a novel one. It is in contrast the indirect approach which employs 3D marker positions as optical measurements which have been previously calculated from the 2D marker images. The output model for this case reads

$${}^C y_i = {}^C m_i = {}^C p + {}^{BC} q \star {}^B m_i \star {}^{BC} q^{-1} .$$

The indirect approach seems to be the standard in the literature on optical-inertial tracking as presented in Section 3.2.1. Our direct data fusion algorithm saves processing time otherwise necessary for 3D marker computations and should thus reduce latency. It also prevents the optical marker position estimation errors from being propagated in the estimation. See also Section 3.2.2 for a motivation of the direct approach.

4.3.3 MEKF

The first data fusion algorithm we propose for the system presented in Chapter 2 is a multiplicative Extended Kalman Filter (MEKF). It uses the system model presented in Section 2.2.6 and a multiplicative error term for the quaternion as discussed in Section 4.2.2. All other error terms are the standard linear ones of the Extended Kalman Filter presented in Section 4.2.1.

Filter Equations We propose the following MEKF for the system presented in Section 2.2.6.

$${}^C \dot{\hat{p}} = {}^C \hat{v} + K_p e_y \quad (4.16)$$

$${}^C \dot{\hat{v}} = {}^C G + {}^{BC} \hat{q} * (a_m - {}^B \hat{a}_b) * {}^{BC} \hat{q}^{-1} + K_v e_y \quad (4.17)$$

$${}^{BC} \dot{\hat{q}} = \frac{1}{2} {}^{BC} \hat{q} * (\omega_m - {}^B \hat{\omega}_b) + {}^{BC} \hat{q} * K_q e_y \quad (4.18)$$

$${}^B \dot{\hat{a}}_b = K_a e_y \quad (4.19)$$

$${}^B \dot{\hat{\omega}}_b = K_\omega e_y \quad (4.20)$$

with output error $e_y = y_m - \hat{y}$. The gain

$$K = (-K_p, -K_v, K_q, -K_a, -K_\omega)^T$$

is calculated according to

$$K = PC^T R^{-1}$$

where P satisfies

$$\dot{P} = AP + PA^T + Q - PC^T R^{-1} CP .$$

The matrices A and C are calculated using a linearized error system as shown in the following section. The choice of the covariance parameters and the computation of Q and R will be explained in Section 4.3.5.

Error System We consider the state error

$$e = \begin{bmatrix} e_p \\ e_v \\ e_q \\ e_a \\ e_\omega \end{bmatrix} = \begin{bmatrix} {}^C \hat{p} - {}^C p \\ {}^C \hat{v} - {}^C v \\ {}^{BC} \hat{q}^{-1} * {}^{BC} q \\ {}^B \hat{a}_b - {}^B a_b \\ {}^B \hat{\omega}_b - {}^B \omega_b \end{bmatrix} .$$

The error system reads

$$\begin{aligned} \dot{e}_p &= e_v + K_p e_y \\ \dot{e}_v &= {}^{BC} \hat{q} * (a_m - {}^B \hat{a}_b) * {}^{BC} \hat{q}^{-1} + \tilde{\nu}_a \\ &\quad + {}^{BC} \hat{q} * e_q * (a_m - {}^B \hat{a}_b + e_a) * e_q^{-1} * {}^{BC} \hat{q}^{-1} + K_v e_y \\ \dot{e}_q &= -(\omega_m - {}^B \hat{\omega}_b) \times e_q + \frac{1}{2} e_q * e_\omega - \frac{1}{2} e_q * \nu_\omega - K_q e_y e_q \\ \dot{e}_a &= -\nu_{ab} + K_a e_y \\ \dot{e}_\omega &= -\nu_{\omega b} + K_\omega e_y \end{aligned}$$

where we have set $\tilde{\nu}_a = {}^{BC}q * \nu_a * {}^{BC}q^{-1}$. Notice $\tilde{\nu}_a$ also is a white noise with the same characteristics as ν_a . Indeed,

$$\begin{aligned} E(\tilde{\nu}_a(t)\tilde{\nu}_a^T(t+\tau)) &= E((R_t\nu_a(t))(R_{t+\tau}\nu_a(t+\tau))^T) \\ &= R_t E(\nu_a(t)\nu_a^T(t+\tau))R_{t+\tau}^T \\ &\stackrel{(2.16)}{=} \xi_a^2 R_t R_{t+\tau}^T \delta(\tau) \\ &= \xi_a^2 I_3 \delta(\tau) \end{aligned}$$

where $R_t = R({}^{BC}q(t))$.

We linearize the corresponding error system around $\bar{e} = (0, 0, 1, 0, 0)$ to bring it into the form (4.8):

$$\begin{bmatrix} \delta\dot{e}_p \\ \delta\dot{e}_v \\ \delta\dot{e}_q \\ \delta\dot{e}_a \\ \delta\dot{e}_\omega \end{bmatrix} = (A - KC) \begin{bmatrix} \delta e_p \\ \delta e_v \\ \delta e_q \\ \delta e_a \\ \delta e_\omega \end{bmatrix} - M \begin{bmatrix} \tilde{\nu}_a \\ \nu_\omega \\ \nu_{ab} \\ \nu_{\omega b} \end{bmatrix} + KN\eta_y$$

with matrices

$$A = \begin{bmatrix} 0 & I & 0 & 0 & 0 \\ 0 & 0 & 2R({}^{BC}\hat{q})[a_m - \hat{a}_b]_\times & -R({}^{BC}\hat{q}) & 0 \\ 0 & 0 & -[\omega_m - \hat{\omega}_b]_\times & 0 & \frac{1}{2}I \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.21)$$

$$C_i = \frac{1}{\alpha^2} \begin{bmatrix} \beta & -\alpha & 0 \\ \gamma & 0 & -\alpha \end{bmatrix} [I \ 0 \ 2R({}^{BC}\hat{q})[{}^B m_i]_\times \ 0 \ 0] \quad (4.22)$$

$$C_{i+l} = \frac{1}{\chi^2} \begin{bmatrix} \mu & -\chi & 0 \\ \kappa & 0 & -\chi \end{bmatrix} R_{St} [I \ 0 \ 2R({}^{BC}\hat{q})[{}^B m_i]_\times \ 0 \ 0] \quad (4.23)$$

$$M = \begin{bmatrix} 0 \\ \text{diag}(-I_3, \frac{1}{2}I_3, I_3, I_3) \end{bmatrix} \quad (4.24)$$

$$N = -I_{4l} \quad (4.25)$$

where ${}^C \hat{m}_i = {}^C \hat{p} + {}^{BC}\hat{q} * {}^B m_i * {}^{BC}\hat{q}^{-1} = [\alpha, \beta, \gamma]^T$ and ${}^{CL}\hat{m}_i = R_{St} {}^C \hat{m}_i + t_{St} = [\chi, \mu, \kappa]^T$. C_i and C_{i+l} each correspond to two lines of the matrix C , associated with marker i . For the cross product, we use the notation $[a]_\times b = a \times b$.

4.3.4 Right- and Left-Invariant EKF

We now present two more filters for the system presented in Chapter 2, called Right-Invariant Extended Kalman Filter (RIEKF) and Left-Invariant Extended Kalman

Filter (LIEKF) [Bonnabel et al., 2009]. The system and the filter are right- resp. left-invariant w.r.t. a transformation group, i.e. applying this transformation to the system equations and the correction terms leaves them invariant. In other words, these filters takes into account system symmetries.

We begin by giving a motivation for invariant filters and go on to explaining the concept of system symmetries and invariance and giving two transformation groups for the system studied here. We then give the filter equations for a Right-Invariant and a Left-Invariant EKF and show that they are indeed invariant with respect to the respective transformation group.

4.3.4.1 Motivation for Invariant Kalman Filters

Many physical systems possess symmetries which means that they are left unchanged by one or more transformations, for example by rotation or translation. It seems logical to design a filter which will also have these symmetries.

A symmetry-preserving observer are proposed in [Bonnabel et al., 2008]. It gives a constructive method for designing observers for invariant systems. [Bonnabel et al., 2009] introduces the notion of Invariant Kalman Filters. Here, geometrically adapted correction terms depending on invariant output errors are used instead of the standard linear correction and error terms in an Extended Kalman Filter. Also, the gain matrix is calculated using an invariant state error instead of the standard linear state error. The matrices A and C in Equation (4.8) are constant on a larger set of trajectories than the EKF which is only constant for equilibrium points [Salaün, 2009].

The Invariant EKF is in fact a generalization of the Multiplicative EKF presented in Section 4.2.2. The MEKF is a modification of the general EKF taking into account the geometry of the quaternion space and using a multiplicative quaternion error because a linear error does not make sens for quaternions. Indeed, applying the Invariant EKF concept to a system which estimates a quaternion leads to the same correction term for the quaternion differential equation as the MEKF.

The system presented in Chapter 2 is not completely invariant and thus we cannot apply the method for symmetry-preserving observers in [Bonnabel et al., 2008]. Instead we propose two Invariant EKFs which take into account the fact that the system is invariant with respect to a rotation. Since the Sensor Unit works the same way when rotated into a different orientation, it is indeed logical that the filter should also behave the same way no matter how the Sensor Unit is oriented in space.

4.3.4.2 System Symmetries

In this Section we study the symmetries of the system presented in Section 2.2.6, that is we study the system's invariance with respect to a transformation group. We

start by defining transformation groups and invariance as in [Bonnabel et al., 2008, Martin and Salaün, 2010].

Definition 1. Let G be a Lie Group with identity e and Σ an open set (or more generally a manifold). A *transformation group* $(\phi_g)_{g \in G}$ on Σ is a smooth map

$$(g, \xi) \in G \times \Sigma \mapsto \phi_g(\xi) \in \Sigma$$

such that:

- $\phi_e(\xi) = \xi$ for all ξ
- $\phi_{g_2} \circ \phi_{g_1}(\xi) = \phi_{g_2 g_1}(\xi)$.

By construction ϕ_g is a diffeomorphism on Σ for all g .

We now consider the smooth output system

$$\dot{x} = f(x, u) \tag{4.26}$$

$$y = h(x, u) \tag{4.27}$$

where (x, u, y) belongs to an open subset $\mathcal{X} \subset \mathbb{R}^n \times \mathcal{U} \subset \mathbb{R}^m \times \mathcal{Y} \subset \mathbb{R}^p$ (or more generally a manifold). The signals $u(t)$, $y(t)$ are assumed known (y is measured, u is measured or a known control input). Consider also the local group of transformations on $\mathcal{X} \times \mathcal{U} \times \mathcal{Y}$ defined by

$$(X, U, Y) := (\varphi_g(x), \psi_g(u), \rho_g(y))$$

where φ_g, ψ_g and ρ_g are local diffeomorphisms.

Definition 2. The system (4.26)-(4.27) is *invariant* with *equivariant* output if for all g , x , u

$$\begin{aligned} f(\varphi_g(x), \psi_g(u)) &= D\varphi_g(x) \cdot f(x, u) \\ h(\varphi_g(x), \psi_g(u)) &= \rho_g(h(x, u)) \end{aligned}$$

This property also reads $\dot{X} = f(X, U)$ and $Y = h(X, U)$, i.e. the system is left unchanged by the transformation.

Next we define invariance for an observer:

Definition 3. The observer $\dot{\hat{x}} = F(\hat{x}, u, y)$ is *invariant* if $F(\varphi_g(\hat{x}), \psi_g(u)) = D\varphi_g(\hat{x}) \cdot F(\hat{x}, u, y)$ for all g , \hat{x} , u , y .

The property also reads $\dot{\hat{X}} = F(\hat{X}, U, Y)$.

After having given the definitions for a transformation group and for invariance of a system and of an observer, the rest of this Section will deal with two transformation groups for the system presented in Section 2.2.6. These transformation represent the system symmetries. These transformation groups will be used in the following two Sections 4.3.4.3 and 4.3.4.4 where two observers will be presented which are invariant with respect to these transformation groups.

The first transformation group considered here is

$$\varphi_{q_0}^R \begin{pmatrix} p \\ v \\ q \\ a_b \\ \omega_b \end{pmatrix} = \begin{pmatrix} p \\ v \\ q * q_0 \\ q_0^{-1} * a_b * q_0 \\ q_0^{-1} * \omega_b * q_0 \end{pmatrix} =: \begin{pmatrix} p^\circ \\ v^\circ \\ q^\circ \\ a_b^\circ \\ \omega_b^\circ \end{pmatrix}. \quad (4.28)$$

φ^R is indeed a transformation group since we have $\varphi_{q_1}^R(\varphi_{q_0}^R(x)) = \varphi_{q_0 * q_1}^R(x)$. The system (2.17)–(2.21) is invariant w.r.t. φ^R ; for example, we have for (2.19):

$$\begin{aligned} {}^{BC} \dot{q}^\circ &= \overbrace{{}^{BC} q * q_0}^{\dot{}} = {}^{BC} \dot{q} * q_0 \\ &= \frac{1}{2} ({}^{BC} q * q_0) * q_0^{-1} * (\omega_m - \nu_\omega - {}^B \omega_b) * q_0 \\ &= \frac{1}{2} {}^{BC} q^\circ * (\omega_m^\circ - \nu_\omega^\circ - {}^B \omega_b^\circ) \end{aligned}$$

Here we have used the transformation group ψ for the input:

$$\psi_{q_0}^R \begin{pmatrix} a_m \\ \omega_m \\ {}^B m_i \end{pmatrix} = \begin{pmatrix} q_0^{-1} * a_m * q_0 \\ q_0^{-1} * \omega_m * q_0 \\ q_0^{-1} * {}^B m_i * q_0 \end{pmatrix} =: \begin{pmatrix} a_m^\circ \\ \omega_m^\circ \\ {}^B m_i^\circ \end{pmatrix}$$

with $i \in \{1, 2, \dots, l\}$. The output

$$y_{im} = \frac{f_R}{\langle {}^C m_i, {}^C E_1 \rangle} \begin{bmatrix} \langle {}^C m_i, {}^C E_2 \rangle \\ \langle {}^C m_i, {}^C E_3 \rangle \end{bmatrix} + s u_R + \eta_{yi} = \begin{bmatrix} S y_{i,u} \\ S y_{i,v} \end{bmatrix}$$

respects the transformation φ^R which we will show for the component ${}^S y_{i,u}$ of the output:

$$\begin{aligned}
 {}^S y_{im,u} &= f_R \frac{\langle {}^C m_i, {}^C E_2 \rangle}{\langle {}^C m_i, {}^C E_1 \rangle} + S u_{Ru} + \eta_{yi,u} \\
 &= f_R \frac{\langle {}^C p + {}^{BC} q * {}^B m_i * {}^{BC} q^{-1}, {}^C E_2 \rangle}{\langle {}^C p + {}^{BC} q * {}^B m_i * {}^{BC} q^{-1}, {}^C E_1 \rangle} + S u_{Ru} + \eta_{yi,u} \\
 &= f_R \frac{\langle {}^C p + ({}^{BC} q * q_0) * (q_0^{-1} * {}^B m_i * q_0) * (q_0^{-1} * {}^{BC} q^{-1}), {}^C E_2 \rangle}{\langle {}^C p + ({}^{BC} q * q_0) * (q_0^{-1} * {}^B m_i * q_0) * (q_0^{-1} * {}^{BC} q^{-1}), {}^C E_1 \rangle} + S u_{Ru} + \eta_{yi,u} \\
 &= f_R \frac{\langle {}^C p^\circ + {}^{BC} q^\circ * {}^B m_i^\circ * {}^{BC} q^{\circ-1}, {}^C E_2 \rangle}{\langle {}^C p^\circ + {}^{BC} q^\circ * {}^B m_i^\circ * {}^{BC} q^{\circ-1}, {}^C E_1 \rangle} + S u_{Ru} + \eta_{yi,u}
 \end{aligned}$$

Thus we have

$$y_m = h(x, u) = h(\varphi^R(x), \psi^R(u))$$

The transformation group φ^R corresponds to a quaternion multiplication by the right and thus the system is called "right-invariant".

The second transformation group for the system considered here is

$$\varphi_{q_0}^L \begin{pmatrix} p \\ v \\ q \\ a_b \\ \omega_b \end{pmatrix} = \begin{pmatrix} q_0 * p * q_0^{-1} \\ q_0 * v * q_0^{-1} \\ q_0 * q \\ a_b \\ \omega_b \end{pmatrix} =: \begin{pmatrix} p^\circ \\ v^\circ \\ q^\circ \\ a_b^\circ \\ \omega_b^\circ \end{pmatrix}. \quad (4.29)$$

φ^L is indeed a transformation group since we have $\varphi_{q_1}^L(\varphi_{q_0}^L(x)) = \varphi_{q_1 * q_0}^L(x)$. The system (2.17)–(2.21) is invariant w.r.t. φ^L ; for example, we have for (2.19):

$$\begin{aligned}
 {}^{BC} \dot{\tilde{q}} &= \overbrace{{}^{BC} \dot{q}} \\
 &= \frac{1}{2} q_0 * {}^{BC} \dot{q} * (\omega_m - \nu_\omega - {}^B \omega_b) \\
 &= \frac{1}{2} {}^{BC} \tilde{q} * (\omega_m - \nu_\omega - {}^B \omega_b)
 \end{aligned}$$

Here we have used the transformation group ψ^L for the input:

$$\psi_{q_0}^L \begin{pmatrix} a_m \\ \omega_m \\ E_j \\ \tilde{E}_j \end{pmatrix} = \begin{pmatrix} a_m \\ \omega_m \\ q_0 * E_j * q_0^{-1} \\ q_0 * \tilde{E}_j * q_0^{-1} \end{pmatrix} =: \begin{pmatrix} a_m^\circ \\ \omega_m^\circ \\ E_j^\circ \\ \tilde{E}_j^\circ \end{pmatrix}$$

with $j \in \{1, 2, 3\}$. The output

$$y_{im} = \frac{f_R}{\langle {}^C m_i, {}^C E_1 \rangle} \begin{bmatrix} \langle {}^C m_i, {}^C E_2 \rangle \\ \langle {}^C m_i, {}^C E_3 \rangle \end{bmatrix} + S u_R + \eta_{yi} = \begin{bmatrix} {}^S y_{i,u} \\ {}^S y_{i,v} \end{bmatrix}$$

respects the transformation φ^L which we will show for the component $S_{y_i,u}$ of the output:

$$\begin{aligned}
 S_{y_{im,u}} &= f_R \frac{\langle {}^C m_i, {}^C E_2 \rangle}{\langle {}^C m_i, {}^C E_1 \rangle} + S_{u_{Ru}} + \eta_{y_i,u} \\
 &= f_R \frac{\langle {}^C p + {}^{BC} q * {}^B m_i * {}^{BC} q^{-1}, {}^C E_2 \rangle}{\langle {}^C p + {}^{BC} q * {}^B m_i * {}^{BC} q^{-1}, {}^C E_1 \rangle} + S_{u_{Ru}} + \eta_{y_i,u} \\
 &= f_R \frac{\langle q_0 * ({}^C p + {}^{BC} q * {}^B m_i * {}^{BC} q^{-1}) * q_0^{-1}, q_0 * {}^C E_2 * q_0^{-1} \rangle}{\langle q_0 * ({}^C p + {}^{BC} q * {}^B m_i * {}^{BC} q^{-1}) * q_0^{-1}, q_0 * {}^C E_1 * q_0^{-1} \rangle} + S_{u_{Ru}} + \eta_{y_i,u} \\
 &= f_R \frac{\langle {}^C p^\circ + (q_0 * {}^{BC} q) * {}^B m_i * ({}^{BC} q^{-1} * q_0^{-1}), {}^C E_2^\circ \rangle}{\langle {}^C p^\circ + (q_0 * {}^{BC} q) * {}^B m_i * ({}^{BC} q^{-1} * q_0^{-1}), {}^C E_1^\circ \rangle} + S_{u_{Ru}} + \eta_{y_i,u} \\
 &= f_R \frac{\langle {}^C p^\circ + {}^{BC} q^\circ * {}^B m_i * {}^{BC} q^{\circ-1}, {}^C E_2^\circ \rangle}{\langle {}^C p^\circ + {}^{BC} q^\circ * {}^B m_i * {}^{BC} q^{\circ-1}, {}^C E_1^\circ \rangle} + S_{u_{Ru}} + \eta_{y_i,u}
 \end{aligned}$$

Thus we have

$$y_m = h(x, u) = h(\varphi^L(x), \psi^L(u))$$

The transformation group φ^L represents a quaternion multiplication by the left. This is why the system is called "left-invariant".

4.3.4.3 Right-Invariant EKF

Filter Equations We propose the following right-invariant EKF for the system presented in Section 2.2.6.

$${}^C \dot{\hat{p}} = {}^C \hat{v} + K_p e_y \quad (4.30)$$

$${}^C \dot{\hat{v}} = {}^C G + {}^{BC} \hat{q} * (a_m - {}^B \hat{a}_b) * {}^{BC} \hat{q}^{-1} + K_v e_y \quad (4.31)$$

$${}^{BC} \dot{\hat{q}} = \frac{1}{2} {}^{BC} \hat{q} * (\omega_m - {}^B \hat{\omega}_b) + K_q e_y * {}^{BC} \hat{q} \quad (4.32)$$

$${}^B \dot{\hat{a}}_b = {}^{BC} \hat{q}^{-1} * K_a e_y * {}^{BC} \hat{q} \quad (4.33)$$

$${}^B \dot{\hat{\omega}}_b = {}^{BC} \hat{q}^{-1} * K_\omega e_y * {}^{BC} \hat{q} \quad (4.34)$$

with $e_y = y_m - \hat{y}$. The gain

$$K = -[K_p, K_v, K_q, K_a, K_\omega]^T$$

is calculated according to

$$K = P C^T R^{-1}$$

where P satisfies

$$\dot{P} = A P + P A^T + Q - P C^T R^{-1} C P$$

with A and C as given in the next section.

The transformation group considered here is φ^R as defined in (4.28). The filter (4.30)–(4.34) is invariant w.r.t. φ^R as shown for (4.34):

$$\begin{aligned} {}^B\dot{\hat{\omega}}_b &= \overbrace{q_0^{-1} * {}^B\hat{\omega}_b * q_0}^{\cdot} = q_0^{-1} * {}^B\dot{\hat{\omega}}_b * q_0 \\ &= q_0^{-1} * {}^{BC}\hat{q}^{-1} * K_\omega e_y * {}^{BC}\hat{q} * q_0 \\ &= {}^{BC}\tilde{q}^{-1} * K_\omega e_y * {}^{BC}\tilde{q} \end{aligned}$$

Since the transformation group φ^R represents a quaternion multiplication by the right, this type of EKF is called "right-invariant EKF" (RIEKF).

Error System For the proposed system and filter we consider the state error

$$e = \begin{bmatrix} e_p \\ e_v \\ e_q \\ e_a \\ e_\omega \end{bmatrix} = \begin{bmatrix} C\hat{p} - Cp \\ C\hat{v} - Cv \\ {}^{BC}\hat{q} * {}^{BC}q^{-1} \\ {}^{BC}q * ({}^B\hat{a}_b - {}^B a_b) * {}^{BC}q^{-1} \\ {}^{BC}q * ({}^B\hat{\omega}_b - {}^B\omega_b) * {}^{BC}q^{-1} \end{bmatrix}.$$

These errors are all expressed in camera coordinates. The error system reads:

$$\dot{e}_p = e_v + K_p e_y \quad (4.35)$$

$$\dot{e}_v = -e_q^{-1} * {}^{BC}\hat{q} * (a_m - {}^B\hat{a}_b) * {}^{BC}\hat{q}^{-1} * e_q + \tilde{\nu}_a \quad (4.36)$$

$$+ {}^{BC}\hat{q} * (a_m - {}^B\hat{a}_b) * {}^{BC}\hat{q}^{-1} - e_a + K_v e_y \quad (4.37)$$

$$\dot{e}_q = -\frac{1}{2}e_q * e_\omega + \frac{1}{2}\tilde{\nu}_\omega + K_q e_y e_q \quad (4.38)$$

$$\dot{e}_a = [e_q^{-1} * {}^{BC}\hat{q} * (\omega_m - {}^B\hat{\omega}_b) * {}^{BC}\hat{q}^{-1} * e_q + e_\omega] \times e_a \quad (4.39)$$

$$- \tilde{\nu}_\omega \times e_a - \tilde{\nu}_{ab} + e_q^{-1} * K_a e_y * e_q \quad (4.40)$$

$$\dot{e}_\omega = [e_q^{-1} * {}^{BC}\hat{q} * (\omega_m - {}^B\hat{\omega}_b) * {}^{BC}\hat{q}^{-1} * e_q] \times e_\omega \quad (4.41)$$

$$- \tilde{\nu}_\omega \times e_\omega - \tilde{\nu}_{\omega b} + e_q^{-1} * K_\omega e_y * e_q \quad (4.42)$$

where we have set $\tilde{\nu}_j = {}^{BC}q * \nu_j * {}^{BC}q^{-1}$ for $j \in \{a, \omega, ab, \omega b\}$. Notice $\tilde{\nu}_j$ also is a white noise with the same characteristics as ν_j . Indeed,

$$\begin{aligned} E(\tilde{\nu}_j(t)\tilde{\nu}_j^T(t+\tau)) &= E((R_t \nu_j(t))(R_{t+\tau} \nu_j(t+\tau))^T) \\ &= R_t E(\nu_j(t)\nu_j^T(t+\tau)) R_{t+\tau}^T \\ &\stackrel{(2.16)}{=} \xi_j^2 R_t R_{t+\tau}^T \delta(\tau) \\ &= \xi_j^2 I_3 \delta(\tau) \end{aligned}$$

where $R_t = R(^{BC}q(t))$.

The output error is:

$$e_y = y_m - \hat{y} = \begin{bmatrix} C y \\ CL y \end{bmatrix} + \eta_y - \begin{bmatrix} C \hat{y} \\ CL \hat{y} \end{bmatrix}.$$

We linearize the error system around $(\bar{e}_p, \bar{e}_v, \bar{e}_q, \bar{e}_a, \bar{e}_\omega) = (0, 0, 1, 0, 0)$, neglecting higher-order terms, to bring it into the form (4.8):

$$\begin{bmatrix} \delta \dot{e}_p \\ \delta \dot{e}_v \\ \delta \dot{e}_q \\ \delta \dot{e}_a \\ \delta \dot{e}_\omega \end{bmatrix} = (A - KC) \begin{bmatrix} \delta e_p \\ \delta e_v \\ \delta e_q \\ \delta e_a \\ \delta e_\omega \end{bmatrix} - M \begin{bmatrix} \tilde{v}_a \\ \tilde{v}_\omega \\ \tilde{v}_{ab} \\ \tilde{v}_{\omega b} \end{bmatrix} + KN\eta_y$$

with

$$A = \begin{bmatrix} 0 & I_3 & 0 & 0 & 0 \\ 0 & 0 & -2[R(^{BC}\hat{q})(a_m - \hat{a}_b)]_\times & -I_3 & 0 \\ 0 & 0 & 0 & 0 & -0.5I_3 \\ 0 & 0 & 0 & A_1 & 0 \\ 0 & 0 & 0 & 0 & A_1 \end{bmatrix} \quad (4.43)$$

$$A_1 = [R(^{BC}\hat{q})(\omega_m - \hat{\omega}_b)]_\times \quad (4.44)$$

$$C_i = f_R \frac{1}{\alpha^2} \begin{bmatrix} \beta & -\alpha & 0 \\ \gamma & 0 & -\alpha \end{bmatrix} [I_3 \ 0 \ -2[R(^{BC}\hat{q})^{B}m_i]_\times \ 0 \ 0] \quad (4.45)$$

$$C_{i+l} = f_L \frac{1}{\chi^2} \begin{bmatrix} \mu & -\chi & 0 \\ \kappa & 0 & -\chi \end{bmatrix} R_{St} [I_3 \ 0 \ -2[R(^{BC}\hat{q})^{B}m_i]_\times \ 0 \ 0] \quad (4.46)$$

$$M = \begin{bmatrix} 0 \\ \text{diag}([-I_3, -0.5I_3, I_3, I_3]) \end{bmatrix} \quad (4.47)$$

$$N = I_{4l} \quad (4.48)$$

where ${}^C\hat{m}_i = {}^C\hat{p} + {}^{BC}\hat{q} * {}^Bm_i * {}^{BC}\hat{q}^{-1} = [\alpha, \beta, \gamma]^T$ and ${}^{CL}\hat{m}_i = R_{St} {}^C\hat{m}_i + t_{St} = [\chi, \mu, \kappa]^T$. C_i and C_{i+l} each correspond to two lines of the matrix C , associated with marker i . For the cross product, we use the notation $[a]_\times b = a \times b$.

4.3.4.4 Left-Invariant EKF

Filter Equations We propose the following left-invariant EKF for the system presented in Section 2.2.6.

$${}^C \dot{\hat{p}} = {}^C \hat{v} + {}^{BC} \hat{q} * K_p e_y * {}^{BC} \hat{q}^{-1} \quad (4.49)$$

$${}^C \dot{\hat{v}} = {}^C G + {}^{BC} \hat{q} * (a_m - {}^B \hat{a}_b) * {}^{BC} \hat{q}^{-1} + {}^{BC} \hat{q} * K_v e_y * {}^{BC} \hat{q}^{-1} \quad (4.50)$$

$${}^{BC} \dot{\hat{q}} = \frac{1}{2} {}^{BC} \hat{q} * (\omega_m - {}^B \hat{\omega}_b) + {}^{BC} \hat{q} * K_q e_y \quad (4.51)$$

$${}^B \dot{\hat{a}}_b = K_a e_y \quad (4.52)$$

$${}^B \dot{\hat{\omega}}_b = K_\omega e_y \quad (4.53)$$

with $e_y = y_m - \hat{y}$. The gain

$$K = [-K_p, -K_v, K_q, -K_a, -K_\omega]^T$$

is calculated according to

$$K = PC^T R^{-1}$$

where P satisfies

$$\dot{P} = AP + PA^T + Q - PC^T R^{-1} CP$$

with A and C as given in the next section.

The transformation group considered here is φ^L as defined in (4.29). The filter (4.49)–(4.53) is invariant w.r.t. φ^L as shown for (4.49):

$$\begin{aligned} {}^C \dot{\tilde{p}}_b &= \overbrace{q_0 * {}^C \hat{p} * q_0^{-1}}^{\dot{\tilde{p}}_b} = q_0 * {}^C \hat{v} * q_0^{-1} + q_0 * {}^{BC} \hat{q} * K_p e_y * {}^{BC} \hat{q}^{-1} * q_0^{-1} \\ &= {}^C \tilde{v} + {}^{BC} \tilde{q} * K_p e_y * {}^{BC} \tilde{q}^{-1} \end{aligned}$$

Since the transformation group φ^L represents a quaternion multiplication by the left, this type of EKF is called "left-invariant EKF" (LIEKF).

Error System For the proposed system we consider the state error

$$e = \begin{bmatrix} e_p \\ e_v \\ e_q \\ e_a \\ e_\omega \end{bmatrix} = \begin{bmatrix} {}^{BC} q^{-1} * ({}^C \hat{p} - {}^C p) * {}^{BC} q \\ {}^{BC} q^{-1} * ({}^C \hat{v} - {}^C v) * {}^{BC} q \\ {}^{BC} \hat{q}^{-1} * {}^{BC} q \\ {}^B \hat{a}_b - {}^B a_b \\ {}^B \hat{\omega}_b - {}^B \omega_b \end{bmatrix}.$$

These errors are all expressed in body coordinates. The error system reads:

$$\begin{aligned}
 \dot{e}_p &= -(\omega_m - {}^B\hat{\omega}_b - \nu_\omega) \times e_p + e_v + e_q^{-1} * K_p e_y * e_q \\
 \dot{e}_v &= -(\omega_m - {}^B\hat{\omega}_b - \nu_\omega) \times e_v - e_q^{-1} * (a_m - {}^B\hat{a}_b) * e_q + (a_m - {}^B\hat{a}_b) - e_a \\
 &\quad + \nu_a + e_q^{-1} * K_v e_y * e_q \\
 \dot{e}_q &= -(\omega_m - {}^B\hat{\omega}_b) \times e_q + \frac{1}{2} e_q * e_\omega - \frac{1}{2} e_q * \nu_\omega - K_q e_y * e_q \\
 \dot{e}_a &= -\nu_{ab} + K_a e_y \\
 \dot{e}_\omega &= -\nu_{\omega b} + K_\omega e_y .
 \end{aligned}$$

The output error is:

$$e_y = y_m - \hat{y} = \begin{bmatrix} C y \\ CL y \end{bmatrix} + \eta_y - \begin{bmatrix} C \hat{y} \\ CL \hat{y} \end{bmatrix} .$$

We linearize the error system around $(\bar{e}) = (0, 0, 1, 0, 0)$, neglecting higher-order terms, to bring it into the form (4.8):

$$\begin{bmatrix} \delta \dot{e}_p \\ \delta \dot{e}_v \\ \delta \dot{e}_q \\ \delta \dot{e}_a \\ \delta \dot{e}_\omega \end{bmatrix} = (A - KC) \begin{bmatrix} \delta e_p \\ \delta e_v \\ \delta e_q \\ \delta e_a \\ \delta e_\omega \end{bmatrix} - M \begin{bmatrix} \nu_a \\ \nu_\omega \\ \nu_{ab} \\ \nu_{\omega b} \end{bmatrix} + KN \eta_y$$

with

$$A = \begin{bmatrix} -[\omega_m - \hat{\omega}_b]_\times & I_3 & 0 & 0 & 0 \\ 0 & -[\omega_m - \hat{\omega}_b]_\times & 2[a_m - \hat{a}_b]_\times & -I_3 & 0 \\ 0 & 0 & -[\omega_m - \hat{\omega}_b]_\times & 0 & 0.5I_3 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.54)$$

$$C_i = f_R \frac{1}{\alpha^2} \begin{bmatrix} \beta & -\alpha & 0 \\ \gamma & 0 & -\alpha \end{bmatrix} [R({}^{BC}\hat{q}) \quad 0 \quad 2R({}^{BC}\hat{q})[{}^B m_i]_\times \quad 0 \quad 0] \quad (4.55)$$

$$C_{i+l} = f_L \frac{1}{\chi^2} \begin{bmatrix} \mu & -\chi & 0 \\ \kappa & 0 & -\chi \end{bmatrix} R_{St} [R({}^{BC}\hat{q}) \quad 0 \quad 2R({}^{BC}\hat{q})[{}^B m_i]_\times \quad 0 \quad 0] \quad (4.56)$$

$$M = \begin{bmatrix} 0 \\ \text{diag}([-I_3, 0.5I_3, I_3, I_3]) \end{bmatrix} \quad (4.57)$$

$$N = -I_{4l} \quad (4.58)$$

where ${}^C \hat{m}_i = {}^C \hat{p} + {}^{BC} \hat{q} * {}^B m_i * {}^{BC} \hat{q}^{-1} = [\alpha, \beta, \gamma]^T$ and ${}^{CL} \hat{m}_i = R_{St} {}^C \hat{m}_i + t_{St} = [\chi, \mu, \kappa]^T$. C_i and C_{i+l} each correspond to two lines of the matrix C , associated with marker i . For the cross product, we use the notation $[a]_\times b = a \times b$.

4.3.5 Covariance Parameters

Matrices Q and R are calculated according to (4.6) and (4.7) with M , N of the filter being considered and with noise covariance parameters for noises ν and η_y resp. which are

$$\begin{aligned}\tilde{Q} &= \text{diag}(\xi_a^2 I_3, \xi_\omega^2 I_3, \xi_{ab}^2 I_3, \xi_{\omega b}^2 I_3) \\ \tilde{R} &= \xi_y^2 I_{4l}\end{aligned}$$

with

- ξ_a : accelerometer velocity random walk or output noise density in $m/s^2/\sqrt{Hz}$
- ξ_ω : gyroscope angle random walk or output noise density in $^\circ/s/\sqrt{Hz}$
- ξ_{ab} : accelerometer rate random walk in $m/s^3/\sqrt{Hz}$
- $\xi_{\omega b}$: gyroscope rate random walk in $^\circ/s^2/\sqrt{Hz}$
- ξ_y : output noise density in pixels/ \sqrt{Hz} .

ξ_a and ξ_ω are given in the IMU data sheet. ξ_{ab} and $\xi_{\omega b}$ are tuning parameters; their values are given in the IMU data sheet, but it is reasonable to increase these values to take into account uncompensated bias temperature drift. ξ_y is a tuning parameter as well, including measurement noise and calibration errors.

4.3.6 Continuous-Discrete and Multi-rate

The system model considered here and presented in Section 2.2.6 is continuous. Since the inertial sensor sample rate is 5 to 15 times higher than the optical sample rate, inertial sensor readings are considered continuous and optical sensor readings discrete.

We take this into account for the implementation of the data fusion algorithms by choosing a continuous-discrete multi-rate form.

In the continuous-discrete EKF [Simon, 2006] as it is presented in Section 4.2.3 for the general case, the prediction step is continuous and the correction step discrete. In the following Section, the equations for the continuous-discrete RIEKF will be given as an example for the continuous-discrete case. Continuous-discrete versions of other EKFs follow the same pattern.

Since the two sensors have different sample rates, the filter has to be multi-rate, i.e. predictions and corrections are executed at different rates. In our case, a prediction step is executed at each new IMU reading. Whenever a camera reading is available, a correction step is done. Since the IMU sample rate is an integer multiple of the optical sample rate,

a correction is always preceded by a prediction. Both steps calculate an estimation of the system state: the prediction outputs the so-called a priori estimate $\hat{x}(t)^-$ and the correction the a posteriori estimate \hat{x}_k .

In an ideal case where perfect mathematical models and parameters are used in the filter, the predictions would estimate the state correctly and no corrections would be needed. In practice, models and parameters will always deviate from their real values and thus both predictions and corrections are necessary to estimate the system state.

4.3.6.1 Continuous-discrete RIEKF

The right-invariant EKF (RIEKF) in its continuous-discrete version has the following form: In the prediction step, we have

$$\begin{aligned} {}^C \dot{\hat{p}}^-(t) &= {}^C \hat{v}(t) \\ {}^C \dot{\hat{v}}^-(t) &= {}^C G + {}^{BC} \hat{q}^-(t) * ({}^B a(t) - {}^B \hat{a}_b(t)) ({}^{BC} \hat{q}^-(t))^{-1} \\ {}^{BC} \dot{\hat{q}}^-(t) &= \frac{1}{2} {}^{BC} \hat{q}^-(t) * ({}^B \omega(t) - {}^B \hat{\omega}_b(t)) + \lambda(1 - \|{}^{BC} \hat{q}^-(t)\|^2) {}^{BC} \hat{q}^-(t) \\ {}^B \dot{\hat{a}}_b^-(t) &= 0 \\ {}^B \dot{\hat{\omega}}_b^-(t) &= 0 \end{aligned}$$

and

$$\dot{P}^-(t)^- = AP(t) + P(t)A^T + Q$$

where A is calculated as in (4.43) with $\hat{x} = \hat{x}^-$ and Q as in (4.6).

The term $\lambda(1 - \|{}^{BC} \hat{q}^-(t)\|^2) * {}^{BC} \hat{q}^-(t)$ is added to keep the quaternion norm at 1, since the norm might deviate due to numerical errors. λ is a positive scalar. We now show how this term keeps the norm at one. To simplify notations, we note $\hat{q} := {}^{BC} \hat{q}^-(t)$. We write the derivative

$$\begin{aligned} \overbrace{\|\hat{q}\|^2}^{\cdot} &= 2\hat{q}\dot{\hat{q}}^T + 2\dot{\hat{q}}\hat{q}^T \\ &= 2\hat{q}\left(\frac{1}{2}(\omega_m - {}^B \hat{\omega}_b)^T \hat{q}^T + \lambda(1 - \|\hat{q}\|^2)\hat{q}^T\right) + 2\left(\frac{1}{2}\hat{q}(\omega_m - {}^B \hat{\omega}_b) + \lambda(1 - \|\hat{q}\|^2)\hat{q}\right)\hat{q}^T \\ &= \underbrace{\hat{q}\left((\omega_m - {}^B \hat{\omega}_b)^T + (\omega_m - {}^B \hat{\omega}_b)\right)\hat{q}^T}_{=0} + 4\lambda(1 - \|\hat{q}\|^2) \underbrace{\hat{q}\hat{q}^T}_{\|\hat{q}\|^2} \\ &= 4\lambda(1 - \|\hat{q}\|^2) \|\hat{q}\|^2 \end{aligned}$$

If $0 \leq \|\hat{q}\| < 1$, then the derivative is positive and $\|\hat{q}\|$ will increase and tend to 1. If $1 > \|\hat{q}\|$, then the derivative is negative and $\|\hat{q}\|$ will decrease and tend to 1. Thus the norm will always tend to 1 with the help of this term.

For the correction step at a discrete time t_k , we use $P_k^- = P^-(t_k)$ and $\hat{x}_k^- = \hat{x}^-(t_k)$. The estimated a priori state is corrected according to

$$\begin{aligned} {}^C\hat{p}_k &= {}^C\hat{p}_k^- + K_p e_{y,k} \\ {}^C\hat{v}_k &= {}^C\hat{v}_k^- + K_v e_{y,k} \\ {}^{BC}\hat{q}_k &= {}^{BC}\hat{q}_k^- + K_q e_{y,k} * {}^{BC}\hat{q}_k^- \\ {}^B\hat{a}_{b,k} &= {}^B\hat{a}_{b,k}^- + ({}^{BC}\hat{q}_k^-)^{-1} * K_a e_{y,k} * {}^{BC}\hat{q}_k^- \\ {}^B\hat{\omega}_{b,k} &= {}^B\hat{\omega}_{b,k}^- + ({}^{BC}\hat{q}_k^-)^{-1} * K_\omega e_{y,k} * {}^{BC}\hat{q}_k^- \end{aligned}$$

to give the a posteriori estimate \hat{x}_k . The discrete output error $e_{y,k}$ at time k is $e_{y,k} = y_{mk} - \hat{y}_k$ where y_{mk} are the measured outputs at time k , and \hat{y}_k are the estimated marker images calculated with Equations (2.22) and (2.23) using \hat{x}_k^- . The gain $K = -[K_p, K_v, K_q, K_a, K_\omega]^T$ is determined according to

$$K = P_k^- C_k^T (C_k P_k^- C_k^T + R_d)^{-1} .$$

where the matrix C_k is calculated using equations (4.45) and (4.46) for the continuous case with $\hat{x} = \hat{x}_k^-$. Since the quaternion norm might deviate from 1 due to numerical problems, we normalize ${}^{BC}\hat{q}_k^-$ after the correction step.

Finally, the matrix P is updated according to

$$P_k = (I - K_k C_k) P_k^- .$$

Note that for discrete measurements we have set

$$R_d = N \tilde{R}_d N^T . \quad (4.59)$$

The discrete \tilde{R}_d is related to the continuous \tilde{R} by $\tilde{R}_d = \tilde{R}/\Delta t$ where Δt is the time between two optical measurements [Simon, 2006, p. 232].

4.4 RIEKF for Calibration

The system model we use in the data fusion algorithms for the optical-inertial tracking system contains several parameters which have been either measured, obtained from calibration or from a datasheet. The parameters used in the model might differ from their true values which will have an impact on the data fusion algorithm. Most importantly, these errors will be absorbed by the estimated biases. Incorrect biases lead to errors in the predicted state. However, it is especially important in our optical-inertial tracking system to have correct prediction steps because these are the steps which give high-bandwidth information. This shows the importance of a correct calibration of the model parameters

for the performance of the tracking system. Several calibration procedures are presented in Section 3.3 and Section 5.4 treats the calibration of the experimental setup.

The parameters in question are the marker model, the camera intrinsic values, the transformation between stereo cameras and the rotation between camera and world frames. In Section 4.4.1 we study the influence of errors in these parameters on the estimated state. It turns out that the errors in the Marker-Body transformation - that is errors in the translation and rotation - lead to very simple expressions of their influence on the estimated state. In fact, a translation error causes an additive error in the position only. A rotational error leads to additive errors in the position, quaternion and accelerometer bias.

The influence of the other parameters cannot be described by simple terms which explains why we go on to analyze only the effect of the camera intrinsic values as an example and do not cover the remaining parameters.

Having determined the influence of parameter errors on the estimated state, the next logical step is to try to determine the parameter errors from the estimated state. However, we do not know the true position, velocity or quaternion and thus cannot determine estimation errors. The only true values of the state we can obtain are accelerometer and gyroscope biases calculated by a calibration procedure as described in Section 5.4. Since an error in the Marker-Body rotation causes an error on the accelerometer biases and we know the true bias, we propose a calibration method for the Marker-Body rotation using RIEKF estimates in Section 4.4.2.

4.4.1 Influence of calibration errors on the RIEKF

4.4.1.1 Error in Marker-Body transformation

True position of marker i :

$${}^C m_i = {}^C p + {}^{BC} q * \underbrace{({}^{MB} R^M m_i + {}^B d)}_{B m_i} * {}^{BC} q^{-1}$$

In the estimated position of each marker i , we make an error Δd in the Marker-Body translation and an error q_E in the Marker-Body rotation:

$${}^C \hat{m}_i = {}^C \hat{p} + {}^{BC} \hat{q} * ({}^{MB} \hat{q} * {}^M m_i * {}^{MB} \hat{q}^{-1} + {}^B \hat{d}) * {}^{BC} \hat{q}^{-1} \quad (4.60)$$

$$= {}^C \hat{p} + {}^{BC} \hat{q} * (q_E * {}^{MB} q * {}^M m_i * {}^{MB} q^{-1} * q_E^{-1} + {}^B d + \Delta d) * {}^{BC} \hat{q}^{-1} \quad (4.61)$$

To calculate the influence of the Marker-Body transformation error on the estimated state in the direct RIEKF, we consider the error system as for the RIEKF in Section 4.3.4.3.

The errors are

$$\begin{bmatrix} e_p \\ e_v \\ e_q \\ e_a \\ e_\omega \end{bmatrix} = \begin{bmatrix} {}^C\hat{p} - {}^Cp \\ {}^C\hat{v} - {}^Cv \\ {}^{BC}\hat{q} * {}^{BC}q^{-1} \\ {}^{BC}q * ({}^B\hat{a}_b - {}^Ba_b) * {}^{BC}q^{-1} \\ {}^{BC}q * ({}^B\hat{\omega}_b - {}^B\omega_b) * {}^{BC}q^{-1} \end{bmatrix}. \quad (4.62)$$

with additional errors $\Delta d = {}^B\hat{d} - {}^Bd$ and $q_E = {}^{MB}\hat{q} * {}^{MB}q^{-1}$. The error system is the same as for the RIEKF (4.35)-(4.42), except for the output error which now writes for marker i :

$$e_{y,i} = \frac{1}{\langle {}^Cm_i, E_1 \rangle} \begin{bmatrix} \langle {}^Cm_i, E_2 \rangle \\ \langle {}^Cm_i, E_3 \rangle \end{bmatrix} - \frac{1}{\langle {}^C\hat{m}_i, E_1 \rangle} \begin{bmatrix} \langle {}^C\hat{m}_i, E_2 \rangle \\ \langle {}^C\hat{m}_i, E_3 \rangle \end{bmatrix}$$

with

$$\begin{aligned} {}^Cm_i &= {}^C\hat{p} - e_p + e_q^{-1} * {}^{BC}\hat{q} * ({}^B\hat{d} - \Delta d) * {}^{BC}\hat{q}^{-1} * e_q \\ &\quad + e_q^{-1} * {}^{BC}\hat{q} * (q_E^{-1} * {}^{MB}\hat{q} * {}^Mm_i * {}^{MB}\hat{q}^{-1} * q_E) * {}^{BC}\hat{q}^{-1} * e_q. \end{aligned}$$

Note that we have omitted the focal distances and principal point parameters to simplify notations.

The solution of the error system in steady state when the filter has converged would give us the steady-state error \bar{e} as a function of \hat{x} , $\overline{\Delta d}$ and \bar{q}_E . Since we are only interested in small errors, we linearize the error system around

$$(\bar{e}_p, \bar{e}_v, \bar{e}_q, \bar{e}_a, \bar{e}_\omega, \overline{\Delta d}_d, \bar{q}_E) = (0, 0, 1, 0, 0, 0, 1), \quad \bar{a}_m = \bar{a} + a_b = {}^BG + a_b, \quad \bar{\omega}_m = \omega_b$$

When we neglect the noise terms, this gives for steady-state:

$$0 = \delta \dot{e}_p = \delta e_v + \sum_i K_{pi} \delta e_{y,i} \quad (4.63)$$

$$\begin{aligned} 0 = \delta \dot{e}_v &= -2 [{}^{BC}\hat{q} * (\bar{a}_m - {}^Ba_b) * {}^{BC}\hat{q}^{-1}] \times \delta e_q - \delta e_a + \sum_i K_{vi} \delta e_{y,i} \\ &= -2 [{}^{BC}\hat{q} * \bar{a} * {}^{BC}\hat{q}^{-1}] \times \delta e_q - \delta e_a + \sum_i K_{vi} \delta e_{y,i} \end{aligned} \quad (4.64)$$

$$0 = \delta \dot{e}_q = -\frac{1}{2} \delta e_\omega + \sum_i K_{qi} \delta e_{y,i} \quad (4.65)$$

$$0 = \delta \dot{e}_a = [{}^{BC}\hat{q} * (\bar{\omega}_m - {}^B\omega_b) * {}^{BC}\hat{q}^{-1}] \times \delta e_a + \sum_i K_{ai} \delta e_{y,i} = \sum_i K_{ai} \delta e_{y,i} \quad (4.66)$$

$$0 = \delta \dot{e}_\omega = [{}^{BC}\hat{q} * (\bar{\omega}_m - {}^B\omega_b) * {}^{BC}\hat{q}^{-1}] \times \delta e_\omega + \sum_i K_{\omega i} \delta e_{y,i} = \sum_i K_{\omega i} \delta e_{y,i} \quad (4.67)$$

Here we have expressed the correction terms as sums of the output errors associated with each marker, for example for the position correction term:

$$K_p E = [K_{p1} K_{p2} \dots K_{pl}] [\delta e_{y,1} \delta e_{y,2} \dots \delta e_{y,l}]^T = \sum_i K_{pi} \delta e_{y,i}$$

where each K_{pi} is a 3x2 matrix. Equations (4.63)-(4.67) give

$$\delta e_v = - \sum_i K_{pi} \delta e_{y,i} \quad (4.68)$$

$$\delta e_a = -2[R(BC\hat{q})\bar{a}]_x \delta e_q + \sum_i K_{vi} \delta e_{y,i} \quad (4.69)$$

$$\delta e_\omega = 2 \sum_i K_{qi} \delta e_{y,i} \quad (4.70)$$

$$0 = \sum_i K_{ai} \delta e_{y,i} \quad (4.71)$$

$$0 = \sum_i K_{\omega i} \delta e_{y,i} \quad (4.72)$$

The linearized output error $e_{y,i} = y_i - \hat{y}_i$ for marker i :

$$\begin{aligned} \delta e_{y,i} &= \frac{1}{\langle F_i, E_1 \rangle} \left[\begin{array}{l} \langle -\delta e_p + 2(BC\hat{q} * Bm_i * BC\hat{q}^{-1}) \times \delta e_q, E_2 \rangle \\ \langle -\delta e_p + 2(BC\hat{q} * Bm_i * BC\hat{q}^{-1}) \times \delta e_q, E_3 \rangle \end{array} \right] \\ &\quad - \frac{\langle -\delta e_p + 2(BC\hat{q} * Bm_i * BC\hat{q}^{-1}) \times \delta e_q, E_1 \rangle}{\langle F_i, E_1 \rangle^2} \left[\begin{array}{l} \langle C\hat{p} + BC\hat{q} * Bm_i * BC\hat{q}^{-1}, E_2 \rangle \\ \langle C\hat{p} + BC\hat{q} * Bm_i * BC\hat{q}^{-1}, E_3 \rangle \end{array} \right] \\ &\quad + \frac{1}{\langle F_i, E_1 \rangle} \left[\begin{array}{l} \langle BC\hat{q} * (2[MBq * Mm_i * MBq^{-1}] \times \delta q_E - \delta d) * BC\hat{q}^{-1}, E_2 \rangle \\ \langle BC\hat{q} * (2[MBq * Mm_i * MBq^{-1}] \times \delta q_E - \delta d) * BC\hat{q}^{-1}, E_3 \rangle \end{array} \right] \\ &\quad - \frac{\langle BC\hat{q} * (2[MBq * Mm_i * MBq^{-1}] \times \delta q_E - \delta d) * BC\hat{q}^{-1}, E_1 \rangle}{\langle F_i, E_1 \rangle^2} \left[\begin{array}{l} \langle F_i, E_2 \rangle \\ \langle F_i, E_3 \rangle \end{array} \right] \\ &= \frac{1}{\langle F_i, E_1 \rangle^2} \underbrace{\begin{bmatrix} \langle F_i, E_2 \rangle & -\langle F_i, E_1 \rangle & 0 \\ \langle F_i, E_3 \rangle & 0 & -\langle F_i, E_1 \rangle \end{bmatrix}}_{G_i} \\ &\quad \cdot (\delta e_p - 2(BC\hat{q} * Bm_i * BC\hat{q}^{-1}) \times \delta e_q \\ &\quad \quad - BC\hat{q} * (2[MBq * Mm_i * MBq^{-1}] \times \delta q_E - \delta d) * BC\hat{q}^{-1}) \end{aligned}$$

where $F_i = C\hat{p} + BC\hat{q} * Bm_i * BC\hat{q}^{-1}$. We can rewrite the linearized output error as

$$\begin{aligned} \delta e_{y,i} &= G_i (\delta e_p - 2(BC\hat{q} * Bm_i * BC\hat{q}^{-1}) \times \delta e_q + BC\hat{q} * \delta d * BC\hat{q}^{-1} \\ &\quad - 2[BC\hat{q} * MBq * Mm_i * MBq^{-1} * BC\hat{q}^{-1}] \times BC\hat{q} * \delta q_E * BC\hat{q}^{-1}) \\ &= G_i (\delta e_p - 2(BC\hat{q} * Bm_i * BC\hat{q}^{-1}) \times \delta e_q + BC\hat{q} * \delta d * BC\hat{q}^{-1} \\ &\quad - 2[BC\hat{q} * (Bm_i - Bd) * BC\hat{q}^{-1}] \times BC\hat{q} * \delta q_E * BC\hat{q}^{-1}) \\ &= G_i (\delta e_p - 2(R(BC\hat{q})^B m_i) \times (\delta e_q + R(BC\hat{q}) \delta q_E) \\ &\quad + 2R(BC\hat{q})(Bd \times \delta q_E) + R(BC\hat{q}) \delta d) \end{aligned} \quad (4.73)$$

Replacing (4.73) in (4.68) gives

$$\begin{aligned}
 \delta e_v &= - \sum_i K_{pi} G_i (\delta e_p - 2(R^{(BC}\hat{q})^B m_i) \times (\delta e_q + R^{(BC}\hat{q})\delta q_E) \\
 &\quad + 2R^{(BC}\hat{q})(^B d \times \delta q_E) + R^{(BC}\hat{q})\delta d) \\
 \Leftrightarrow \delta e_p &= (\sum_j K_{pj} G_j)^{-1} (-\delta e_v + 2 \sum_j K_{pj} G_j (R^{(BC}\hat{q})^B m_j) \times (\delta e_q + R^{(BC}\hat{q})\delta q_E)) \\
 &\quad - 2R^{(BC}\hat{q})(^B d \times \delta q_E) - R^{(BC}\hat{q})\delta d
 \end{aligned} \tag{4.74}$$

Replacing the expression for δe_p in the linearized output error gives

$$\begin{aligned}
 \delta e_{y,i} &= G_i \left((\sum_j K_{pj} G_j)^{-1} (-\delta e_v + 2 \sum_j K_{pj} G_j (R^{(BC}\hat{q})^B m_j) \times (\delta e_q + R^{(BC}\hat{q})\delta q_E)) \right. \\
 &\quad \left. - 2R^{(BC}\hat{q})(^B d \times \delta q_E) - R^{(BC}\hat{q})\delta d - 2(R^{(BC}\hat{q})^B m_i) \times (\delta e_q + R^{(BC}\hat{q})\delta q_E) \right. \\
 &\quad \left. + 2R^{(BC}\hat{q})(^B d \times \delta q_E) + R^{(BC}\hat{q})\delta d \right) \\
 &= G_i \left((\sum_j K_{pj} G_j)^{-1} (-\delta e_v + 2 \sum_j K_{pj} G_j (R^{(BC}\hat{q})^B m_j) \times (\delta e_q + R^{(BC}\hat{q})\delta q_E)) \right. \\
 &\quad \left. - 2(R^{(BC}\hat{q})^B m_i) \times (\delta e_q + R^{(BC}\hat{q})\delta q_E) \right) \\
 &= G_i \left(-(\sum_j K_{pj} G_j)^{-1} \delta e_v \right. \\
 &\quad \left. + \left[2(\sum_j K_{pj} G_j)^{-1} \sum_j K_{pj} G_j [(R^{(BC}\hat{q})^B m_j]_{\times} - 2[R^{(BC}\hat{q})^B m_i]_{\times}] (\delta e_q + R^{(BC}\hat{q})\delta q_E) \right] \right) \\
 &= G_i (H_v \delta e_v + H_{qi} (\delta e_q + R^{(BC}\hat{q})\delta q_E))
 \end{aligned}$$

The linearized output error can be expressed as a function of δe_v and $\delta e_q + R^{(BC}\hat{q})\delta q_E$. Thus, according to (4.68)-(4.72) and (4.74), all the linearized errors can be expressed as functions of δe_v , δe_q , δq_E and δd . We will now solve for δe_v and $\delta e_q^* := \delta e_q + R^{(BC}\hat{q})\delta q_E$.

Starting with equation (4.71), we have:

$$\sum_i K_{ai} \delta e_{y,i} = \sum_i K_{ai} G_i (H_v \delta e_v + H_{qi} \delta e_q^*) = 0 \tag{4.75}$$

Equation (4.72) gives:

$$\sum_i K_{\omega i} \delta e_{y,i} = \sum_i K_{\omega i} G_i (H_v \delta e_v + H_{qi} \delta e_q^*) = 0 \tag{4.76}$$

Equations (4.76) and (4.75) form a homogeneous linear system in δe_v and δe_q^* :

$$\underbrace{\begin{bmatrix} \sum_i K_{ai} G_i H_v & \sum_i K_{ai} G_i H_{qi} \\ \sum_i K_{\omega i} G_i H_v & \sum_i K_{\omega i} G_i H_{qi} \end{bmatrix}}_H \begin{bmatrix} \delta e_v \\ \delta e_q^* \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{4.77}$$

We consider H is invertible because it is a submatrix of $A - KC$. This matrix is constant in steady-state and since we look at the case where the filter has converged, all the eigenvalues have negative real parts and the matrix is invertible. We verified numerically that the submatrix H of $A - KC$ is also invertible. Consequently, $\delta e_v = 0$, $\delta e_q^* = 0$ and $\delta e_{y,i} = 0$. The complete linear state error reads

$$\text{from (4.74):} \quad \delta e_p = -2R^{(BC}\hat{q})({}^B d \times \delta q_E) - R^{(BC}\hat{q})\delta d \quad (4.78)$$

$$\delta e_v = 0 \quad (4.79)$$

$$\delta e_q = -R^{(BC}\hat{q})\delta q_E \quad (4.80)$$

$$\text{from (4.69):} \quad \delta e_a = -2[R^{(BC}\hat{q})\bar{a}]_{\times}\delta e_q = 2R^{(BC}\hat{q})[\bar{a}]_{\times}\delta q_E \quad (4.81)$$

$$\text{from (4.70):} \quad \delta e_{\omega} = 0 \quad (4.82)$$

An error Δd in the Marker-Body translation produces an error in the estimated position but does not affect the other estimated states.

An error q_E in the Marker-Body rotation causes errors in the estimated position, quaternion and accelerometer biases but does not affect the other estimated states. When using calibrated accelerometer measurements we know the true biases and can use this information to determine the error caused by the rotation error. From the error, we can calculate the rotation error and the true Marker-Body rotation. In Section 4.4.2, we propose a method for calibration of the Marker-Body rotation using this reasoning.

Here, we have considered the case where a single (right) camera is used. We omitted the second (left) camera from the calculations to avoid further complicating the equations, but using two cameras would lead to the same result.

4.4.1.2 Error on Principal Point

True position of marker i :

$${}^C m_i = {}^C p + {}^{BC} q * \underbrace{({}^{MB} R^M m_i + {}^B d)}_{{}^B m_i} * {}^{BC} q^{-1}$$

In the estimated position of marker i :

$${}^C \hat{m}_i = {}^C \hat{p} + {}^{BC} \hat{q} * \underbrace{({}^{MB} R^M m_i + {}^B d)}_{{}^B m_i} * {}^{BC} \hat{q}^{-1}$$

The true principal point is ${}^S u$ and the estimated principal point is ${}^S \hat{u} = {}^S u + \Delta u$ where $\Delta u = [\Delta u_0, \Delta v_0]^T$ is the principal point error.

For the error analysis, we follow the same reasoning as in the previous section. We start by examining the steady-state error system with the same state error as in the

previous Section in Equation (4.62) with an additional error $e_u = \Delta u$. We linearize the error system around

$$(\bar{e}_p, \bar{e}_v, \bar{e}_q, \bar{e}_a, \bar{e}_\omega, \bar{e}_u) = (0, 0, 1, 0, 0, 0), \quad \bar{a}_m = \bar{a} + a_b = {}^B G + a_b, \quad \bar{\omega}_m = \omega_b$$

which gives the same result as in equations (4.68)-(4.72). The output error $e_{y,i} = y_i - \hat{y}_i$ for marker i reads:

$$e_{y,i} = \frac{1}{\langle {}^C m_i, E_1 \rangle} \left[\begin{array}{c} f_u \langle {}^C m_i, E_2 \rangle \\ f_v \langle {}^C m_i, E_3 \rangle \end{array} \right] + (S \hat{u} - \Delta u) - \frac{1}{\langle {}^C \hat{m}_i, E_1 \rangle} \left[\begin{array}{c} \hat{f}_u \langle {}^C \hat{m}_i, E_2 \rangle \\ \hat{f}_v \langle {}^C \hat{m}_i, E_3 \rangle \end{array} \right] - S \hat{u} \quad (4.83)$$

For the linearized output error we obtain:

$$\delta e_{y,i} = \underbrace{\frac{1}{\langle F_i, E_1 \rangle^2} \left[\begin{array}{ccc} f_u \langle F_i, E_2 \rangle & -f_v \langle F_i, E_1 \rangle & 0 \\ f_v \langle F_i, E_3 \rangle & 0 & -f_v \langle F_i, E_1 \rangle \end{array} \right]}_{G_i} \cdot (\delta e_p - 2({}^{BC} \hat{q} \star {}^B m_i \star {}^{BC} \hat{q}^{-1}) \times \delta e_q) - \delta u$$

where $F_i = {}^C \hat{p} + B_i = {}^C \hat{m}_i$. Note that the matrix G_i is the same as in the previous section, except for the focal distances which are taken into account here. We now rewrite the term $-\delta u$, using the expression $\delta U = [0, \delta u_0/f_u, \delta v_0/f_v]^T$:

$$\begin{aligned} -\delta u &= - \begin{bmatrix} \delta u_0 \\ \delta v_0 \end{bmatrix} = - \begin{bmatrix} f_u \langle \Delta U, E_2 \rangle \\ f_v \langle \Delta U, E_3 \rangle \end{bmatrix} = - \begin{bmatrix} 0 & f_u & 0 \\ 0 & 0 & f_v \end{bmatrix} \delta U \\ &= \frac{1}{\langle F_i, E_1 \rangle^2} \begin{bmatrix} 0 & -f_u \langle F_i, E_1 \rangle & 0 \\ 0 & 0 & -f_v \langle F_i, E_1 \rangle \end{bmatrix} \langle F_i, E_1 \rangle \delta U \end{aligned}$$

Since $\langle \delta U, E_1 \rangle = 0$, we can finally write

$$-\delta u = \frac{1}{\langle F_i, E_1 \rangle^2} \left[\begin{array}{ccc} f_u \langle F_i, E_2 \rangle & -f_u \langle F_i, E_1 \rangle & 0 \\ f_v \langle F_i, E_3 \rangle & 0 & -f_v \langle F_i, E_1 \rangle \end{array} \right] \langle F_i, E_1 \rangle \delta U = G_i \langle F_i, E_1 \rangle \delta U$$

This yields for the linearized output error:

$$\delta e_{y,i} = G_i (\delta e_p - 2({}^{BC} \hat{q} \star {}^B m_i \star {}^{BC} \hat{q}^{-1}) \times \delta e_q + \langle F_i, E_1 \rangle \delta U)$$

which we can approximate to

$$\delta e_{y,i} = G_i (\delta e_p - 2B_i \times \delta e_q + \langle \bar{F}, E_1 \rangle \delta U) \quad \text{for all } i.$$

The principal point error Δu causes an error of the same form as in (4.73) for the Marker-Body translation error. Thus all the errors are zero except for the position error which is

$$\delta e_p = -\langle \bar{F}, E_1 \rangle \delta U = -\langle \bar{F}, E_1 \rangle \begin{bmatrix} 0 \\ \Delta u_0/f_u \\ \Delta v_0/f_v \end{bmatrix}$$

This is the result for an optical-inertial system with a single (right) camera. Using two cameras would lead to a different error on the estimated state because the error has an effect on the projection to the right camera only. We do not consider this case further because we cannot exploit this result to determine the principal point in the absence of the reference of the true position.

4.4.1.3 Error on Focal Distance

The true focal distances are f_u and f_v and the estimated ones $\hat{f}_u = f_u + \Delta f_u$ and $\hat{f}_v = f_v + \Delta f_v$. The focal distance errors are noted Δf_u and Δf_v .

For the error analysis, we follow the same reasoning as in the Section 4.4.1.1. We start by examining the steady-state error system with the same state error as in the previous Section in equation (4.62) with additional errors $e_{f_u} = \Delta f_u$ and $e_{f_v} = \Delta f_v$. We linearize the error system around

$$(\bar{e}_p, \bar{e}_v, \bar{e}_q, \bar{e}_a, \bar{e}_\omega, \bar{e}_{f_u}, \bar{e}_{f_v}) = (0, 0, 1, 0, 0, 0, 0), \quad \bar{a}_m = \bar{a} + a_b = {}^B G + a_b, \quad \bar{\omega}_m = \omega_b$$

which gives the same result as in equations (4.68)-(4.72). The output error $e_{y,i} = y_i - \hat{y}_i$ for marker i reads:

$$e_{y,i} = \frac{1}{\langle {}^C m_i, E_1 \rangle} \left[\begin{array}{l} (\hat{f}_u - \Delta f_u) \langle {}^C m_i, E_2 \rangle \\ (\hat{f}_v - \Delta f_v) \langle {}^C m_i, E_3 \rangle \end{array} \right] + s_u - \frac{1}{\langle {}^C \hat{m}_i, E_1 \rangle} \left[\begin{array}{l} \hat{f}_u \langle {}^C \hat{m}_i, E_2 \rangle \\ \hat{f}_v \langle {}^C \hat{m}_i, E_3 \rangle \end{array} \right] - s_u$$

For the linearized output error we obtain:

$$\begin{aligned} \delta e_{y,i} &= \underbrace{\frac{1}{\langle F_i, E_1 \rangle^2} \begin{bmatrix} f_u \langle F_i, E_2 \rangle & -f_u \langle F_i, E_1 \rangle & 0 \\ f_v \langle F_i, E_3 \rangle & 0 & -f_v \langle F_i, E_1 \rangle \end{bmatrix}}_{G_i} (\delta e_p - 2B_i \times \delta e_q) \\ &\quad + \frac{1}{\langle F_i, E_1 \rangle} \begin{bmatrix} \delta f_u \langle F_i, E_2 \rangle \\ \delta f_v \langle F_i, E_3 \rangle \end{bmatrix} \\ &= G_i (\delta e_p - 2B_i \times \delta e_q) + \begin{bmatrix} \frac{\langle F_i, E_2 \rangle}{\langle F_i, E_1 \rangle} & 0 \\ 0 & \frac{\langle F_i, E_3 \rangle}{\langle F_i, E_1 \rangle} \end{bmatrix} \begin{bmatrix} \delta f_u \\ \delta f_v \end{bmatrix} \\ &= G_i (\delta e_p - 2B_i \times \delta e_q) + \text{diag}(\hat{y}_i) \delta f . \end{aligned}$$

The term with δf depends on the marker i which prevents simple expressions for the linearized state errors like we found in sections 4.4.1.1 and 4.4.1.2 for Marker-Body transformation and principal point errors. Also, we only considered one (right) camera in the above calculations. The use of two cameras would further complicate the analysis since an error on the focal distances of one camera only affects the projection to this camera.

4.4.2 Calibration of Marker-Body Rotation with RIEKF

The analysis of the influence of marker-body transformation errors on the estimated state in Section 4.4.1.1 shows that the errors are propagated to certain state variables only. We would like to use this fact to determine the marker-body transformation errors and thus calibrate the marker-body transformation. We will propose here a method to determine the marker-body rotation using the RIEKF estimation for several static measurements.

The position, quaternion and accelerometer biases are affected by the transformation errors. Since we do not know the true position and orientation, we cannot compute the position and quaternion error. However, we can determine the true accelerometer biases from one of the accelerometer calibration routines described in Section 3.3. From the biases estimated by the RIEKF, we can compute the bias error. The accelerometer bias error was calculated in (4.81):

$$\delta e_a = 2R({}^{BC}\hat{q})[a_m - {}^B\hat{a}_b] \times \delta q_E$$

Also, from the definition of the bias error in Section 4.3.4.3 we have

$$e_a = {}^{BC}q * ({}^B\hat{a}_b - {}^B a_b) * {}^{BC}q^{-1} \quad \Rightarrow \quad \delta e_a = {}^{BC}\hat{q} * ({}^B\hat{a}_b - {}^B a_b) * {}^{BC}\hat{q}^{-1}$$

in steady state. These equations give

$$2R({}^{BC}\hat{q})[a_m - {}^B\hat{a}_b] \times \delta q_E = R({}^{BC}\hat{q})({}^B\hat{a}_b - {}^B a_b) \quad (4.84)$$

$$\Leftrightarrow 2[a_m - {}^B\hat{a}_b] \times \delta q_E = {}^B\hat{a}_b - {}^B a_b \quad (4.85)$$

This equation permits us to compute δq_E from the accelerometer measurements a_m , the true biases a_b as determined by an accelerometer calibration procedure and the biases ${}^B\hat{a}_b$ estimated by an RIEKF. This is done for a static case.

Equation (4.85) provides only two linearly independent equations for the three elements of vector δq_E . Consequently we have to measure a_m and estimate ${}^B\hat{a}_b$ for at least two different orientations. The equations (4.85) for different orientations are then stacked to give

$$2 \underbrace{\begin{bmatrix} [a'_m - {}^B\hat{a}'_b]_{\times} \\ [a''_m - {}^B\hat{a}''_b]_{\times} \\ \dots \end{bmatrix}}_A \delta q_E = \underbrace{\begin{bmatrix} {}^B\hat{a}'_b - {}^B a'_b \\ {}^B\hat{a}''_b - {}^B a''_b \\ \dots \end{bmatrix}}_b$$

A is a $3N \times 3$ matrix and b is a vector of length $3N$ where N is the number of different orientations. The rank of A is 3 if the orientations are different. We solve this over-determined set of linear equations with a linear least-squares solution using the SVD [Hartley and Zisserman, 2003, p.589].

The above equations and the analysis of the calibration result can be simplified by using calibrated accelerometer measurements in which the biases have been compensated for. The calibrated measurements are noted

$$\tilde{a}_m = a_m - {}^B a_b .$$

Equation (4.85) then becomes

$$2\tilde{a}_m \times \delta q_E = {}^B \hat{a}_b .$$

With calibrated accelerometer measurements, the accelerometer biases estimated by the RIEKF should be zero.

This method is implemented and tested for the experimental setup in Section 5.4.4.2.

Chapter 5

Implementation and Experimental Results

Ce chapitre porte sur l'implémentation d'un système de tracking optique-inertiel et des algorithmes de fusion de données proposés.

La première partie décrit le dispositif expérimental avec des capteurs optiques et inertiels, l'acquisition de données par un microcontrôleur et l'implémentation des algorithmes en Simulink et avec xPC Target pour une exécution en temps réel.

Nous présentons ensuite les expériences faites avec le dispositif expérimental et les résultats obtenus. Ils montrent que le système optique-inertiel peut suivre le mouvement de la Sensor Unit avec une grande bande passante. Nous montrons aussi l'intérêt du calibrage du système optique-inertiel.

Finalement, nous traitons des différentes méthodes utilisées pour calibrer les différents capteurs et le système optique-inertiel.

5.1 Experimental Setup

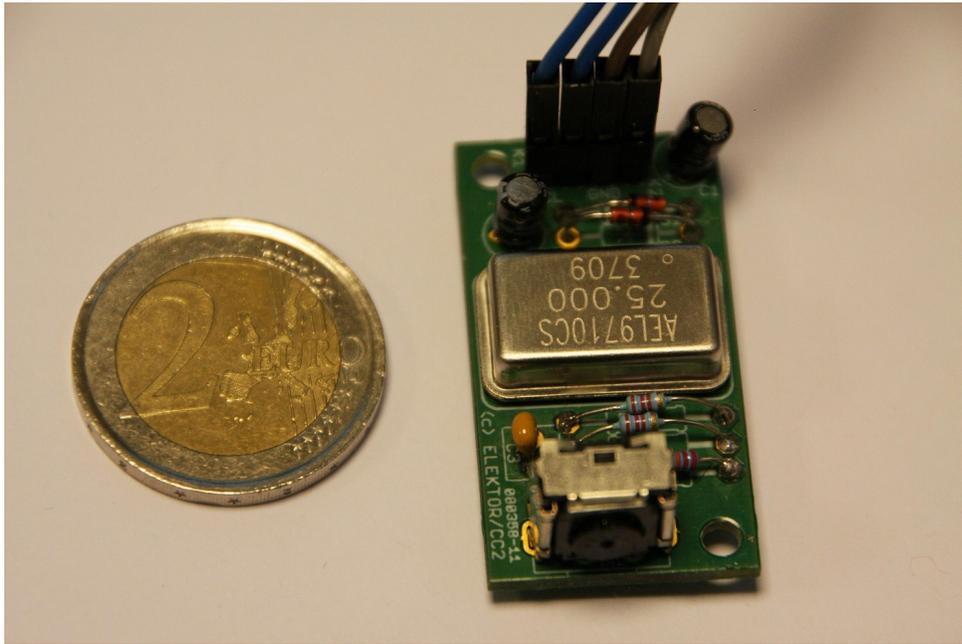
5.1.1 Cameras and Optical Markers

The camera used in the experimental setup is a Wiimote image sensor. The Wiimote is the remote control of the Wii game console (Nintendo Co., Ltd., Japan [Nintendo Co., Ltd., 2011]) and is depicted in Figure 5.1. The image sensor was unsoldered and put on a dedicated PCB board [Jürss and Rudolph, 2008] to be used on its own. Figure 5.2(b) shows the PCB board with the image sensor and Figure 5.2(a) the sensor on its own.

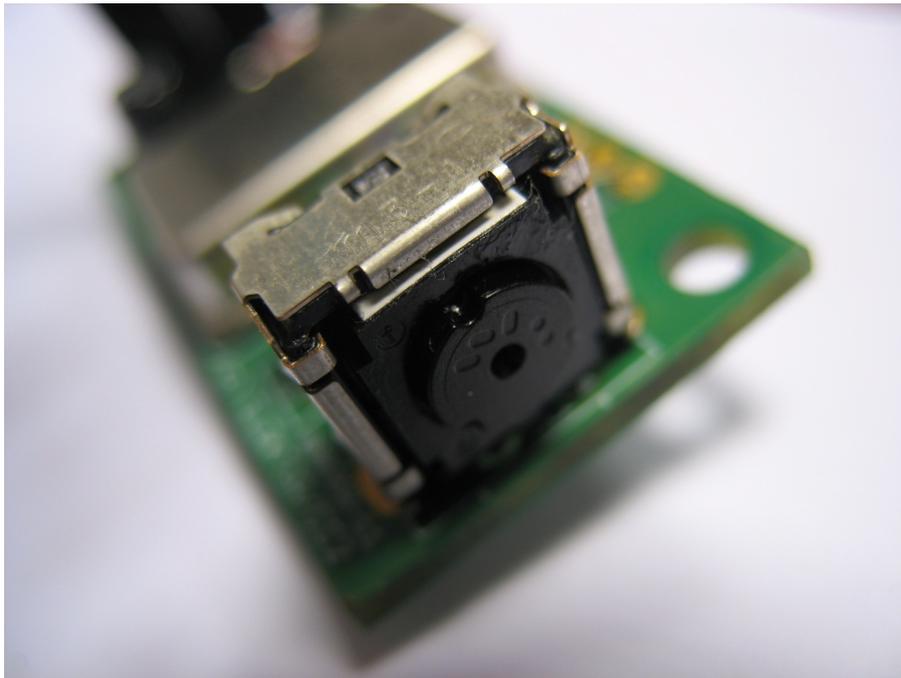
The image sensor sees up to four luminous points which are called blobs and outputs the 2D image coordinates and the blob size. In [Hay et al., 2008] the sensors were sampled



Figure 5.1: Wiimote



(a) Image sensor on PCB board



(b) Image sensor

Figure 5.2: Wiimote image sensor

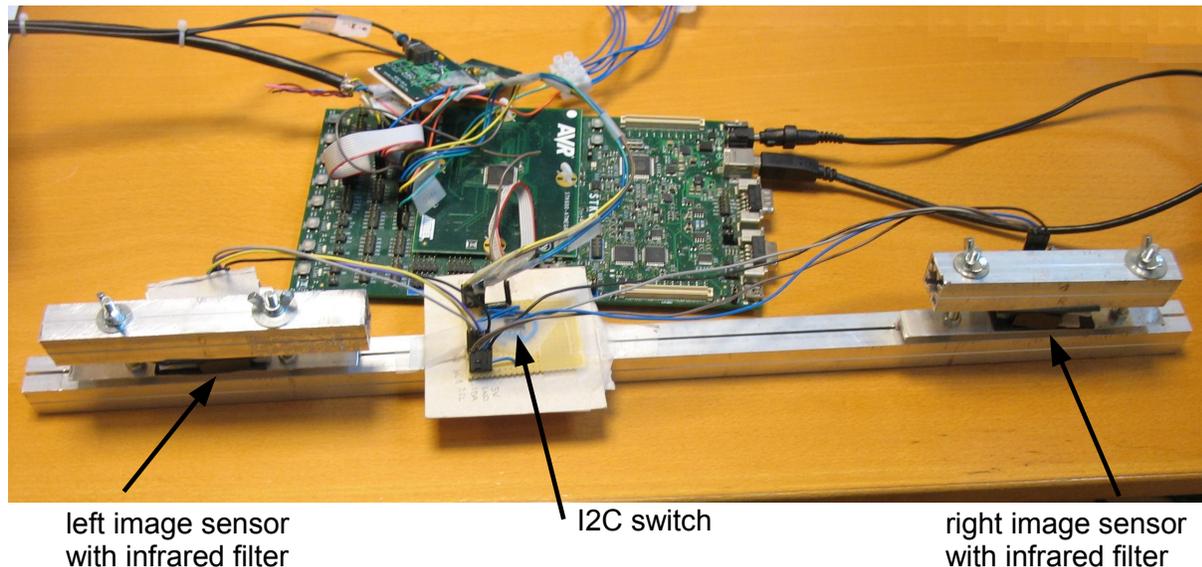


Figure 5.3: stereo rig

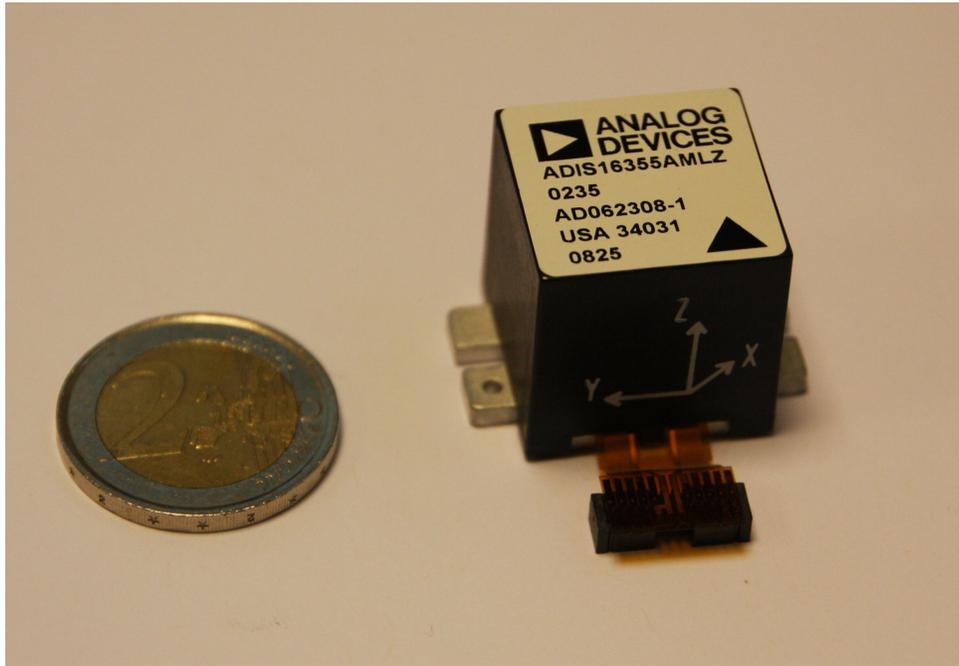
at 120Hz but this rate seems to be too high for such a low-cost sensor; the true rate at which the sensor processes new data is probably lower. We decided to sample the image sensor at 50Hz. Each sensor reading consists of three bytes for each of the four blobs. The first two bytes correspond to the lower eight bits of the two blob coordinates. The third byte includes the coordinates' higher two bits and the blob size.

The image sensor communicates via the Inter-Integrated Circuit (I2C) protocol which is a two-wire interface with a clock (SCL) and a data (SDA) line.

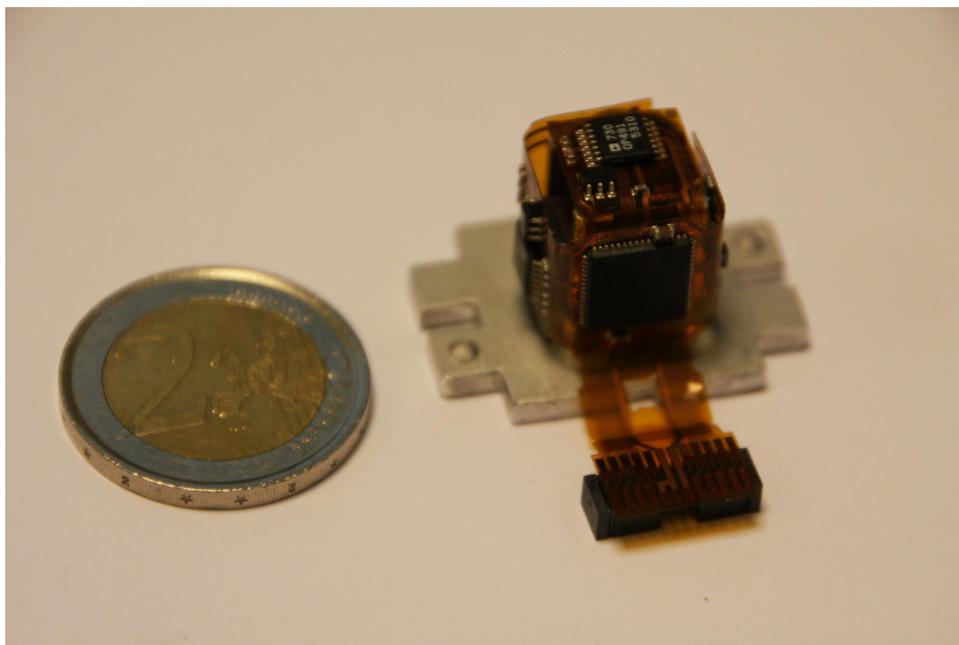
To build a stereo system, two image sensors are fixed relative to each other in a rig with a baseline of approximately 30cm. Since all the image sensors have the same I2C address we use an I2C switch to commute between the two sensors. Figure 5.3 depicts the stereo rig with the I2C switch and the STK600 development board.

Four Vishay light-emitting diodes (LEDs) are used as optical markers. An infrared filter in front of the image sensor ensures that it only detects the LEDs and no other light sources.

This image sensor was chosen as a camera in the experimental setup because it is low-cost (a Wiimote costs about 35€), can be easily interfaced and synchronized with a microcontroller and because it does image processing internally.



(a) ADIS 16355 IMU



(b) inside of ADIS 16355 IMU

Figure 5.4: ADIS 16355 IMU from Analog Devices

5.1.2 Inertial Sensors

The inertial measurement unit "ADIS 16355" [ADIS16355, 2009] from Analog Devices is used in this experimental setup. It consists of 3 orthogonal accelerometers and 3 orthogonal gyroscopes. The sensor with its casing is shown in figure 5.4(a). Figure 5.4(b) illustrates the positioning of the different sensors and other components in the inside of the ADIS IMU.

The ADIS 16355 outputs a 14-bit value per sensor which has to be transformed to a specific acceleration resp. an angular velocity following the format given in the datasheet [ADIS16355, 2009]. The measurements are temperature-compensated and have a bandwidth of 350Hz. Default settings were used for all parameters (maximum internal sample rate of 810.2 samples/s, dynamic range of $\pm 300^\circ/\text{s}$, no automatic calibration) except for the linear acceleration bias compensation which was enabled for the gyroscopes.

The IMU communicates via the Serial Peripheral Interface (SPI) Bus. This is a four wire serial bus, consisting of the four logic signals SCLK (serial clock), MOSI (master output, slave input), MISO (master input, slave output) and SS (slave select).

5.1.3 Sensor Unit

Four LEDs are mounted on a board (see Figure 5.5) and connected at the back to an external power supply. They are arranged in a plane with distances such that they can be detected by the image sensor. The ADIS IMU is fixed on the back of the board as shown in Figure 5.5. Figure 5.6 pictures the Sensor Unit without its casing and illustrates how the IMU is attached to the Sensor Unit front. The Sensor Unit is connected to a microcontroller for communication and power supply.

Note that the size of this experimental Sensor Unit is much bigger than that of the final Sensor Unit which would be attached to a handheld tool. This size was chosen to be able to adapt the setup during the development of this experimental setup and to fix it to the rail carriage. Also, the low-cost camera we use demands a distance of several centimeters between the LEDs.

5.1.4 Data Acquisition

An Atmega2560 microcontroller [ATmega2560, 2007] from Atmel on a STK600 development board reads sensor data from the image sensor via I2C and from the IMU via SPI and sends them to a PC via its UART. A counter triggers the different sensor readings at the chosen frequency.

The SPI protocol is intended for use over short distances of a few centimeters, but in this case a distance of 1m has to be overcome between the IMU and the microcontroller.

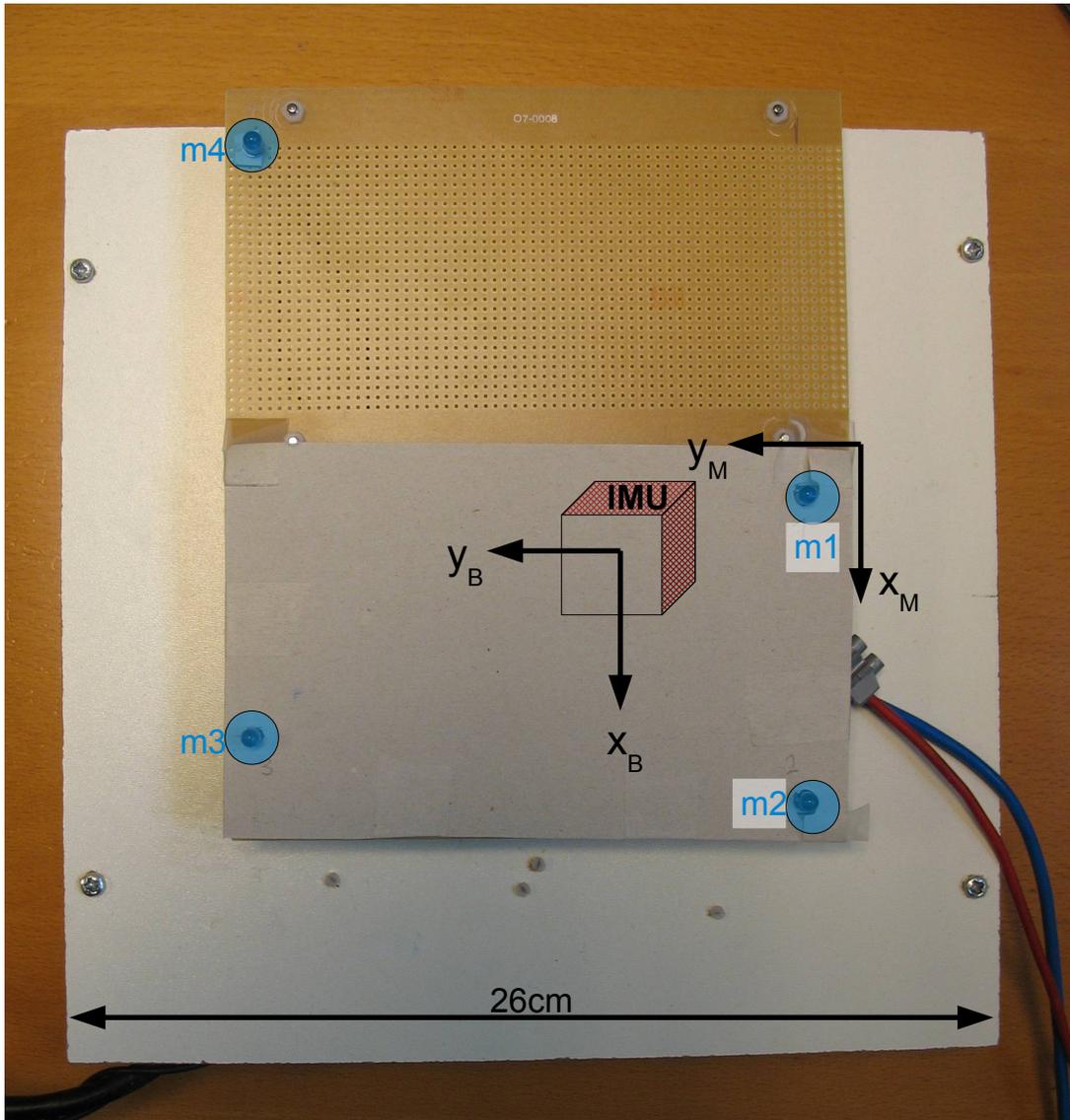
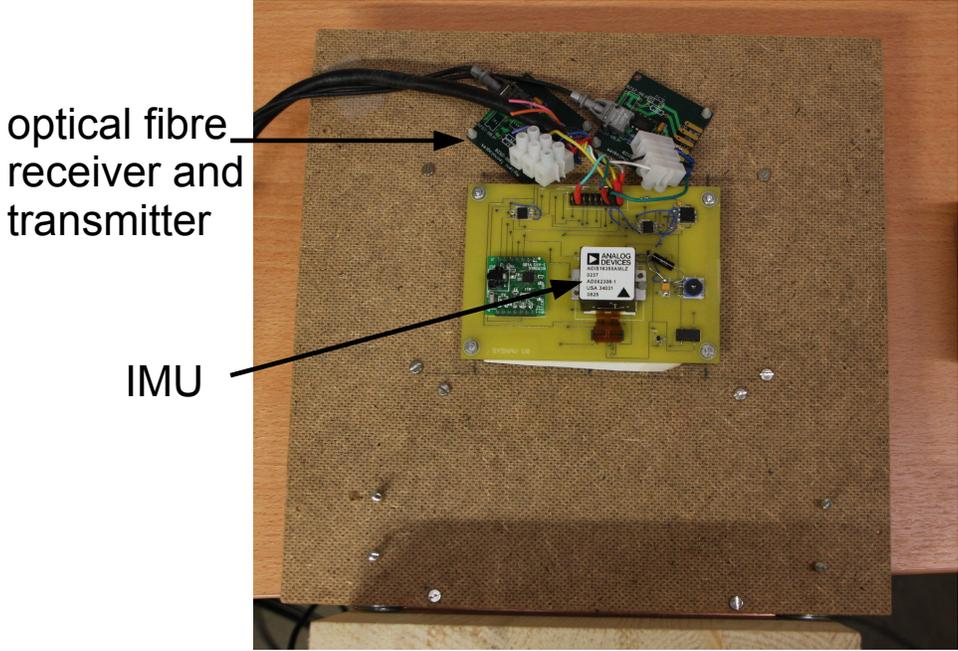
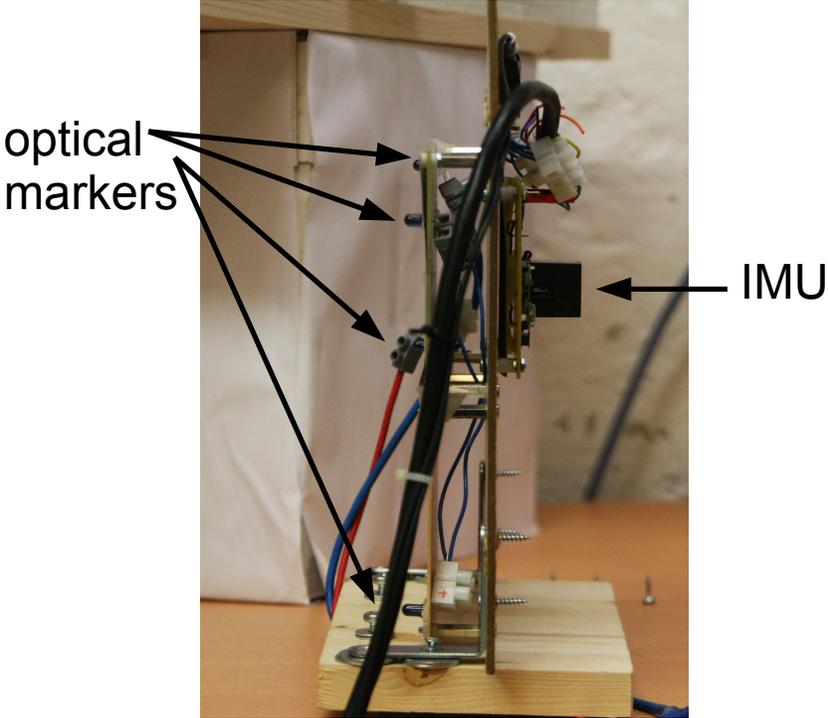


Figure 5.5: Sensor unit



(a) inside



(b) side

Figure 5.6: Sensor Unit without its casing

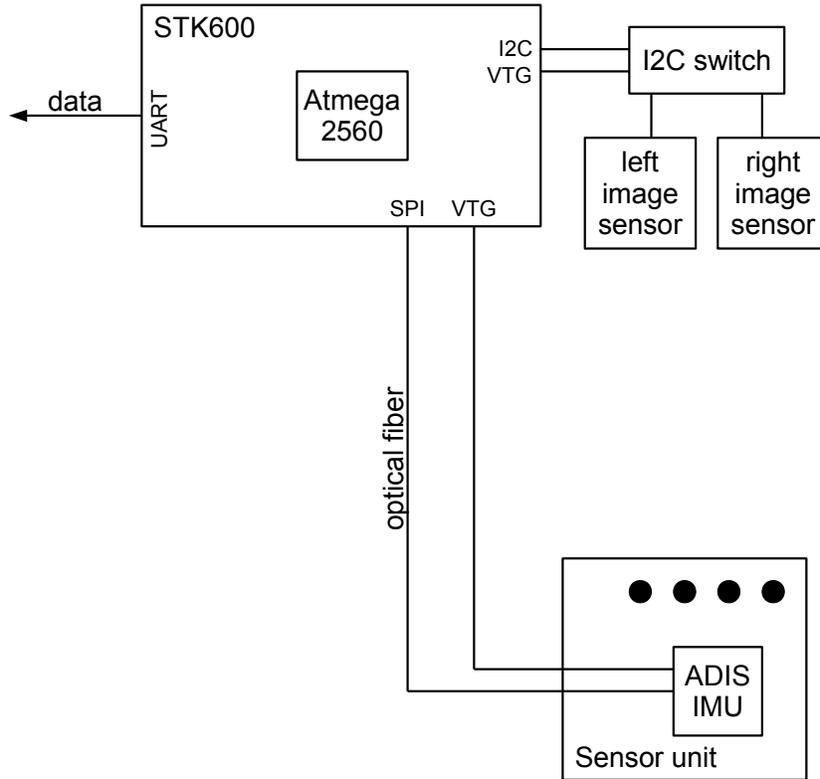


Figure 5.7: Setup

To prevent problems due to interferences, we used optical fiber for transmission of MISO and MOSI signals.

The setup in figure 5.7 shows how the IMU and image sensors are connected to the microcontroller.

5.1.5 Algorithm Implementation

The different data fusion algorithms for the optical-inertial tracking systems presented in Section 4.3 were implemented in two ways: at first offline with Simulink and secondly in real-time.

5.1.5.1 Offline implementation with Simulink

Several data fusion algorithms were implemented in their continuous-discrete forms (see Section 4.3.6) in Simulink [Mathworks, 2011] for offline analysis of saved data sequences.

The data sequences have been acquired by the ATmega2560 microcontroller, then read by the PC via its serial port and saved in a Matlab file. In the data sequences, each frame is time-stamped. Correction steps are triggered at the chosen optical sample rate by a pulse function.

Figure 5.8 shows the Simulink model for the RIEKF. Subfigure 5.8(a) gives an overview of the model. The measured data are read in "from workspace" blocks in the upper left corner and inertial data are transformed from raw data to physical values. The lower left corner contains several constant parameters. The block "RIEKF" represents the data fusion algorithm and outputs the estimated vector "xhat" which is saved in a Matlab file. The inside of the RIEKF block is illustrated in Subfigure 5.8(b). Corrections and predictions are placed in separate blocks. The correction block is executed according to the optical sample rate; it is triggered by signal generated in the "pulse generator". The integrators are reset at the same frequency.

5.1.5.2 Real-time implementation with xPC Target

The RIEKF was implemented in real-time using xPC Target, the real-time software environment in Matlab/Simulink [Mathworks, 2011]. Sensor data are acquired by the ATmega2560 microcontroller and sent via UART and are then read on the serial port by the xPC Target application. This application executes the RIEKF (as implemented in Simulink in Section 5.1.5.1) with a step of 1ms and outputs the estimated state. Predictions are executed when new inertial measurements are available. Corrections are triggered by new optical measurements.

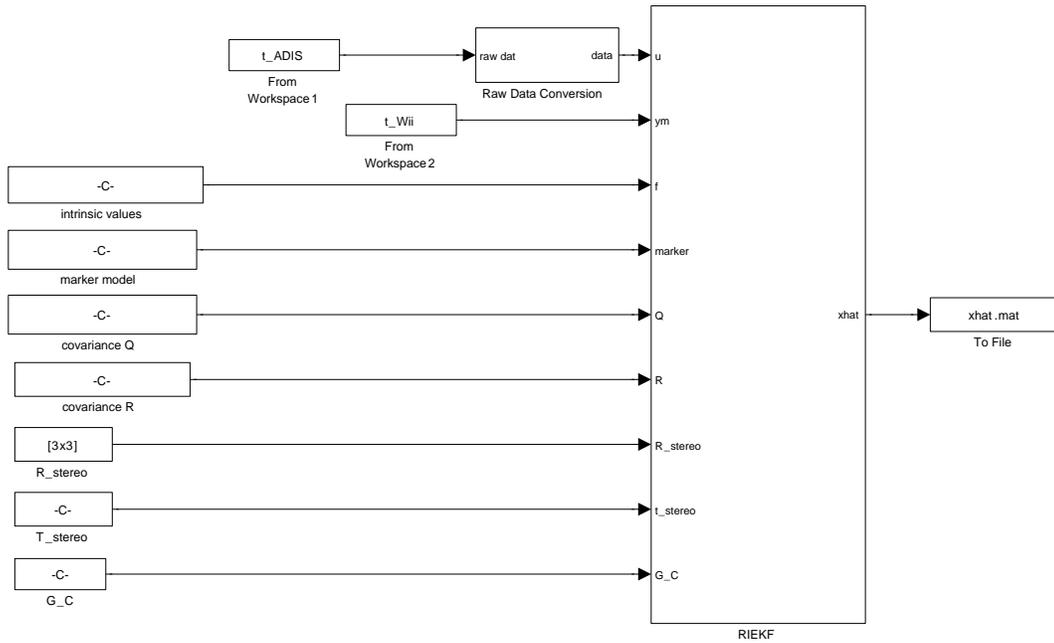
Figure 5.9 shows the setup for the real-time implementation with xPC Target.

Data acquisition timing in xPC Target: Sending one frame via the UART with a baud rate of 115200bps takes a little under 4ms. This latency is not taken into account in the analysis. However, the frames comprise a variable latency of up to 1ms due to the fact that xPC Target samples the serial port every 1ms and that this sampling is not synchronized with the microcontroller. This leads to variable time gaps between consecutive predictions and between consecutive corrections.

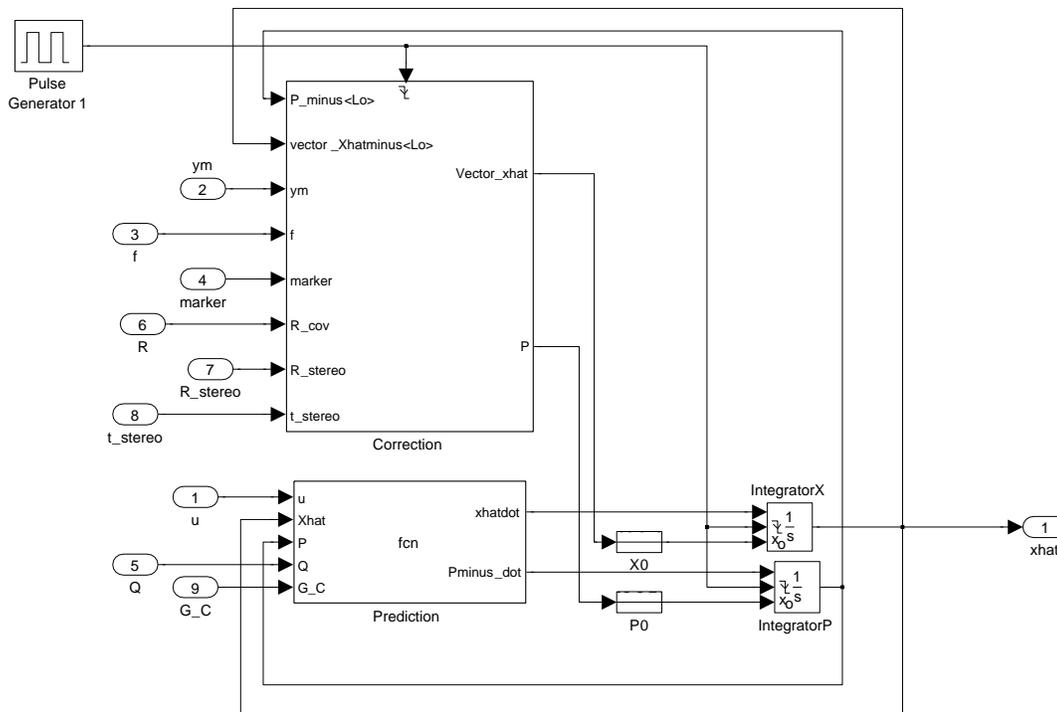
The xPC target application logs the state estimated by the RIEKF but also the data frames read on the serial port. These data are sent to the host PC and can then be used for offline analysis for optical tracking or other data fusion algorithms under the same conditions as for the real-time RIEKF.

5.1.5.3 Filter Tuning

The filter's covariance parameters are presented in Section 4.3.5. For the ADIS16355, the white noise parameters can be read from the Allan variance plots given in the datasheet [ADIS16355, 2009] which give the Allan variance on a certain time interval.



(a) Overview



(b) Inside of block "RIEKF"

Figure 5.8: RIEKF Simulink model

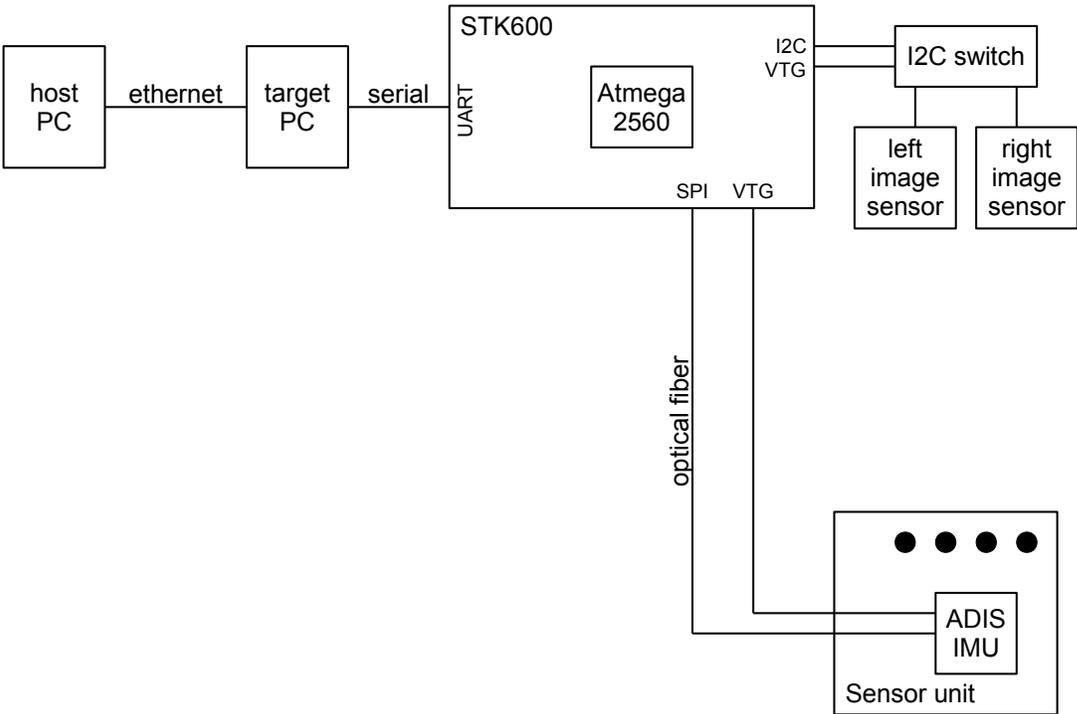


Figure 5.9: Experimental setup for real-time implementation with xPC Target

The remaining parameters were found by graphically approximating the Allan variance values given in the datasheet by the two lines representing the two terms we use in the sensor model.

Figure 5.10(a) shows the points read from the datasheet plot for the accelerometer Allan variance, the two lines and the approximated Allan variance. The velocity random walk ξ_a is equal to $2.84mg/\sqrt{Hz}$ and the accelerometer rate random walk ξ_{ab} has been determined as $0.04mg/s/\sqrt{Hz}$.

Figure 5.10(b) shows the points read from the datasheet plot for the gyroscope Allan variance, the two lines and the approximated Allan variance. The angle random walk ξ_ω is equal to $0.07^\circ/s/\sqrt{Hz}$ and the gyroscope rate random walk $\xi_{\omega b}$ has been determined as $9 \cdot 10^{-4}^\circ/s^2/\sqrt{Hz}$.

The output noise density ξ_y of the Wiimote image sensor is not known and can only be estimated. We chose $\xi_y = 1.5\text{pixels}/\sqrt{Hz}$ for an optical sample rate of 50Hz which takes into account the estimated noise density but also calibration errors.

5.1.5.4 Data Association

Data association is the process of finding the correspondence between a) markers in stereo images and b) marker images and the marker model.

Stereo Matching Left and right images can be matched using epipolar geometry. A corresponding pair of right and left images (y_R, y_L) satisfies [Hartley and Zisserman, 2003, p. 245] the relation

$$y_L^T F y_R = 0$$

where F is the fundamental matrix. For a stereo rig with known camera matrices $P_R = K_R [I \ 0]$ and $P_L = K_L [R_{St} \ t_{St}]$, the fundamental matrix reads [Hartley and Zisserman, 2003, p. 244]

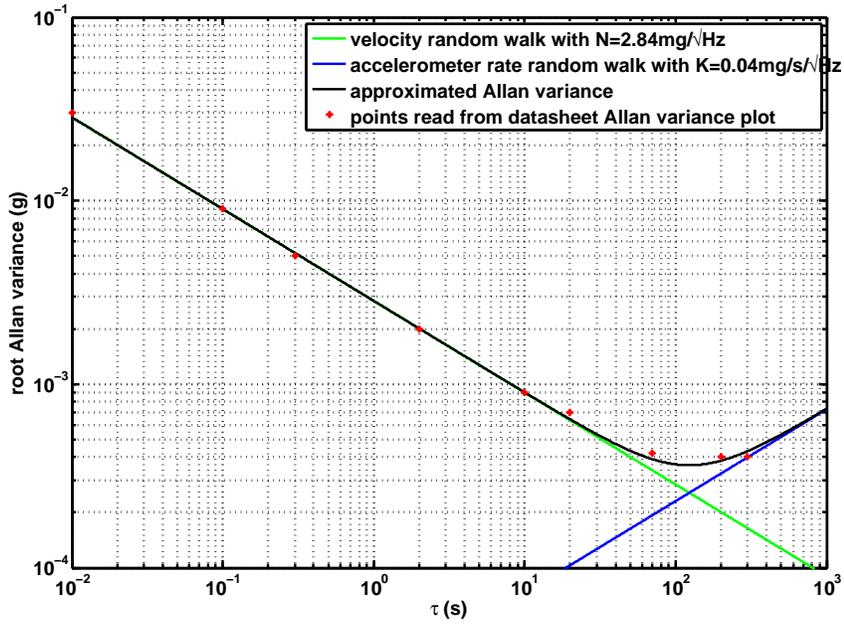
$$F = [P_L B]_{\times} P_L P_R^+ . \quad (5.1)$$

P_R^+ is the pseudo-inverse of P_R , i.e. $P_R P_R^+ = I$ [Hartley and Zisserman, 2003, p. 244], and $B = [0, 0, 0, 1]^T$. Equation (5.1.5.4) can be developed to

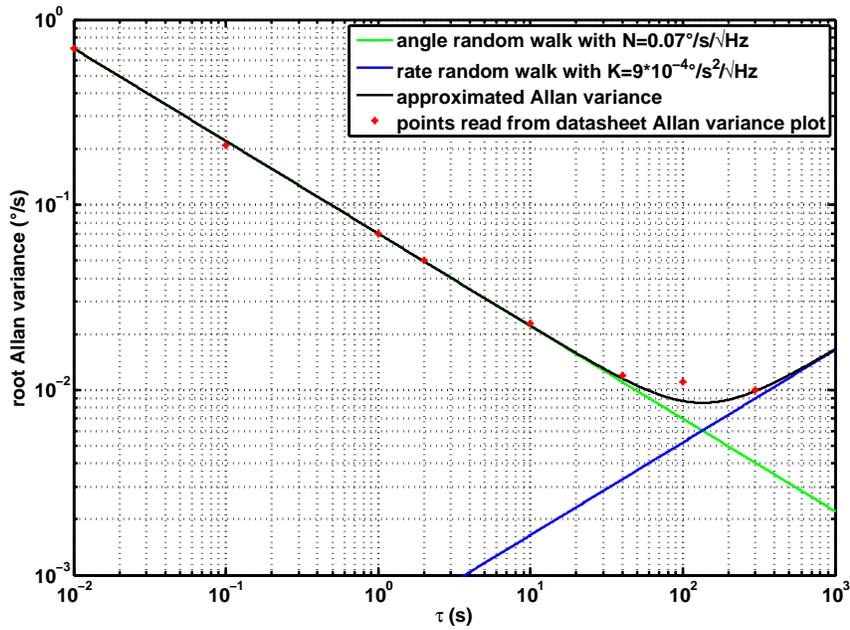
$$F = [K_L t_{St}]_{\times} K_L R_{St} K_R^{-1} .$$

Association of Marker Images with Marker Model This association can be achieved in the Kalman filter (see for example Section 4.3.4.3 for the RIEKF) using the Mahalanobis distance [Jung and Wohn, 1997] which depends on the difference between measured and estimated outputs and on their respective covariances:

$$d_{ij} = (y_{m,i} - \hat{y}_j)^T (R_i + C_j P C_j^T)^{-1} (y_{m,i} - \hat{y}_j)$$



(a) Allan variance plot for the accelerometer



(b) Allan variance plot for the gyroscope

Figure 5.10: Allan variance plots with points read from the sensor datasheet and approximated curves

$y_{m,i}$ is the 2D measurement vector of the i th image and R_i its measurement covariance. C_j are two lines of matrix C , corresponding to j th model marker. P is the estimate error covariance.

5.1.6 Tracking References

Obviously, one of the main goals for the experimental setup and the implementation of the optical-inertial data fusion algorithm was to study the accuracy of the pose tracking and to show that our proposed optical-inertial system determines the object motion correctly. For this, we needed a reference to which to compare the optical-inertial pose estimation. This reference had to give the true pose at a high sample rate.

One possibility was to use an independent optical tracking system and attach this system's rigid body and our Sensor Unit to the same object. With this setup interferences occurred between the two optical systems and the transformation between the rigid body and the Sensor Unit could only be determined approximately. The same was true for the transformation between the two reference frames in which the pose estimation were expressed. The main problem which prevented this setup from being used as reference was that the optical tracking bandwidth was too low to follow fast motion which is the case we are interested in.

The best solution we found was to use a motorized linear rail. The Sensor Unit is attached to the rail carriage. The rail measures the carriage position with a resolver with a sample rate of 250Hz. We know the movements executed by the rail carriage to be linear, that is the Sensor Unit moves along a line, which permits us to interpolate the true carriage position between resolver measurements. Note that there is no automatic synchronization between the rail resolver measurement and the optical-inertial setup. The resolver values are shifted in time until they are superposed with the optical-inertial data where the beginning and end of a movement serve as reference points.

A second goal for this implementation was to compare the performance of our optical-inertial tracking system and of purely optical tracking to show the advantages of the first one. For this, we used the optical data from our setup to determine the Sensor Unit position using computer vision algorithms presented in Section 3.1.

In the following we present a few details on how we obtained references from the linear rail and for optical tracking.

5.1.6.1 Linear Rail

A 3m motorized linear rail was used to execute repeatable experiments with known linear motions. Figure 5.11 shows the linear rail.

The rail can be controlled manually in the "JOG mode" or automatically in the "Positioning mode". In the latter a motion profile can be defined. This motion profile

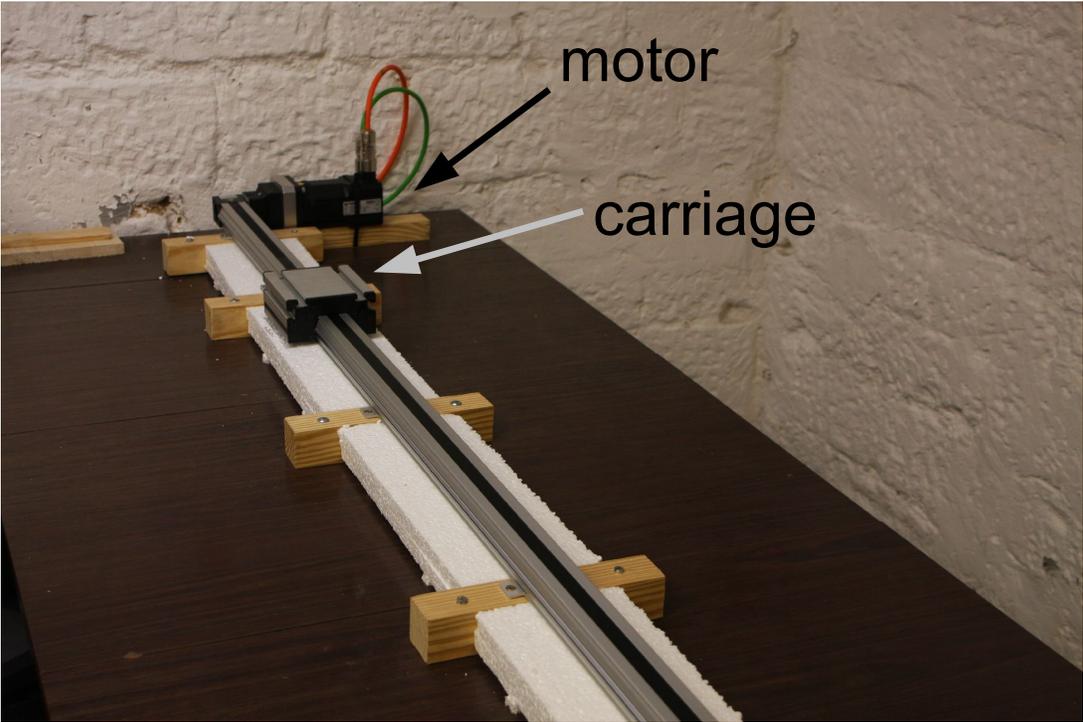


Figure 5.11: linear rail

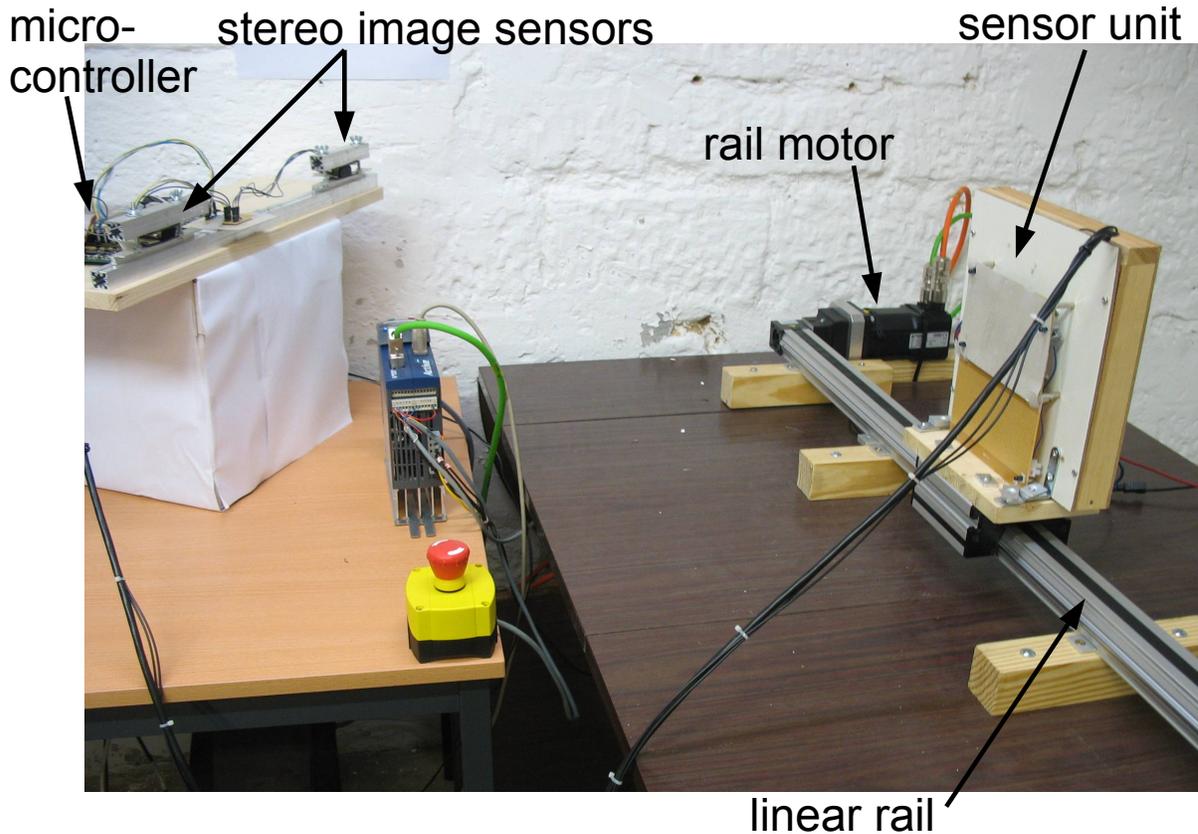


Figure 5.12: Experimental setup with stereo camera rig, Sensor Unit and linear rail

or trajectory consists of ramps and static parts. For each ramp, the desired position is defined together with the maximal acceleration, deceleration and velocity values allowed for attaining the desired position.

The Sensor Unit is mounted rigidly on the carriage. Figure 5.12 shows the experimental setup with the Sensor Unit mounted on the carriage.

The rail measures the carriage position with a resolver with a sample time of 4ms. The measurements may contain small errors due to the fact that it does not measure the absolute position of the carriage on the rail but deduces the position from the resolver measurements in the motor which introduces uncertainties caused by the transmission. The resolver measurement gives this carriage position in 1D, that is its position along the linear rail. We use this information to determine the 3D position of the Sensor Unit by taking the position estimated by the RIEKF at the beginning and at the end of a motion along the rail to be the beginning and end of the linear motion in 3D. The resolver measurement then gives the position along this line.

5.1.6.2 Optical Tracking

Computer Vision algorithms for calculating the pose of the Sensor Unit using only optical data from the experimental setup were implemented in Matlab [Mathworks, 2011]. Image data were first triangulated by basic triangulation [Hartley and Zisserman, 2003] as presented in Section 3.1.2.1 to obtain 3D marker positions. The Sensor Unit pose was calculated by Walker's solution [Walker et al., 1991] to the absolute orientation problem as presented in Section 3.1.2.2.

5.2 Experiments

The following experiments were conducted using the setup presented in Section 5.1.

5.2.1 Experiment 1: static case

The Sensor Unit was set on a horizontal surface during several minutes.

5.2.2 Experiment 2: slow linear motion

In this experiment, the Sensor Unit was rigidly fixed to the carriage of the linear rail and moved along the rail according to a pre-defined trajectory.

Figure 5.13 shows the trajectory for the linear rail. The position varies between 0 and 10cm with a maximal acceleration/deceleration of $4m/s^2$ and maximal velocity of $30cm/s$.

5.2.3 Experiment 3: fast oscillating linear motion

The linear rail was used to execute an oscillating motion with an amplitude of $1cm$.

Figure 5.14 shows the planned trajectory for the linear rail. The position oscillates between 0 and 1cm with a maximal acceleration/deceleration of $10m/s^2$ and maximal velocity of $30cm/s$. Note that due to the high accelerations and decelerations overshoots occurred during the oscillations. A detail of the actual trajectory executed by the rail carriage and measured by the resolver is shown in Figure 5.15.

5.2.4 Experiment 4: static orientations

For the calibration of marker-body rotation 5.4.4.2, the Sensor Unit was fixed in 7 different static orientations and kept static for about 90s in each orientation. A tripod was used to rotate the Sensor Unit as shown in figure 5.16.

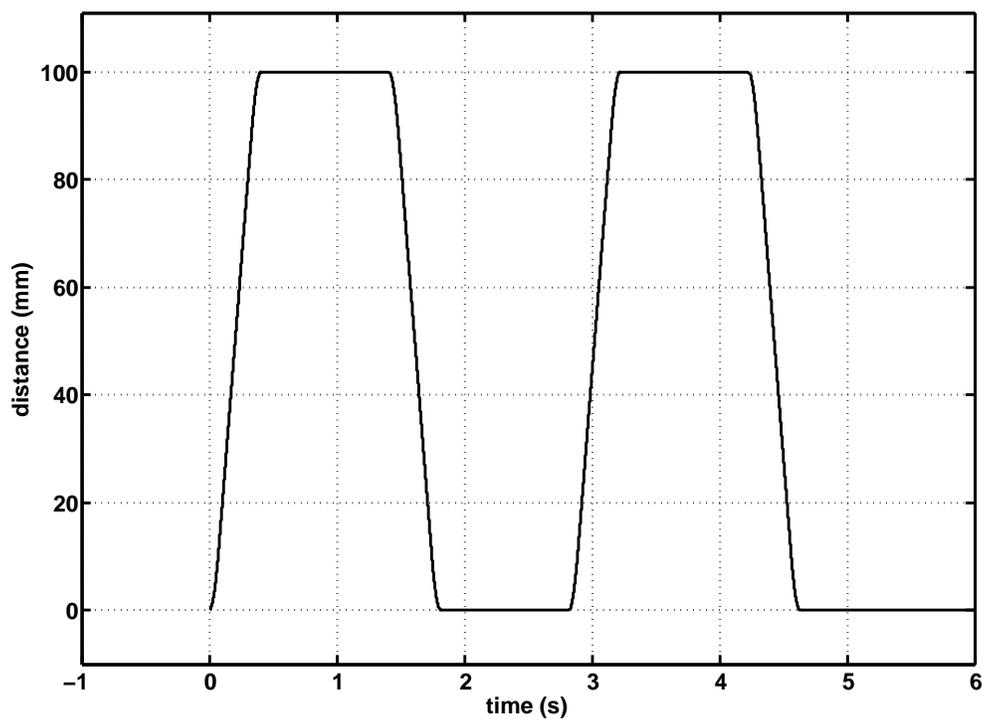


Figure 5.13: Trajectory for the linear rail in positioning mode for experiment 2 (slow linear motion)

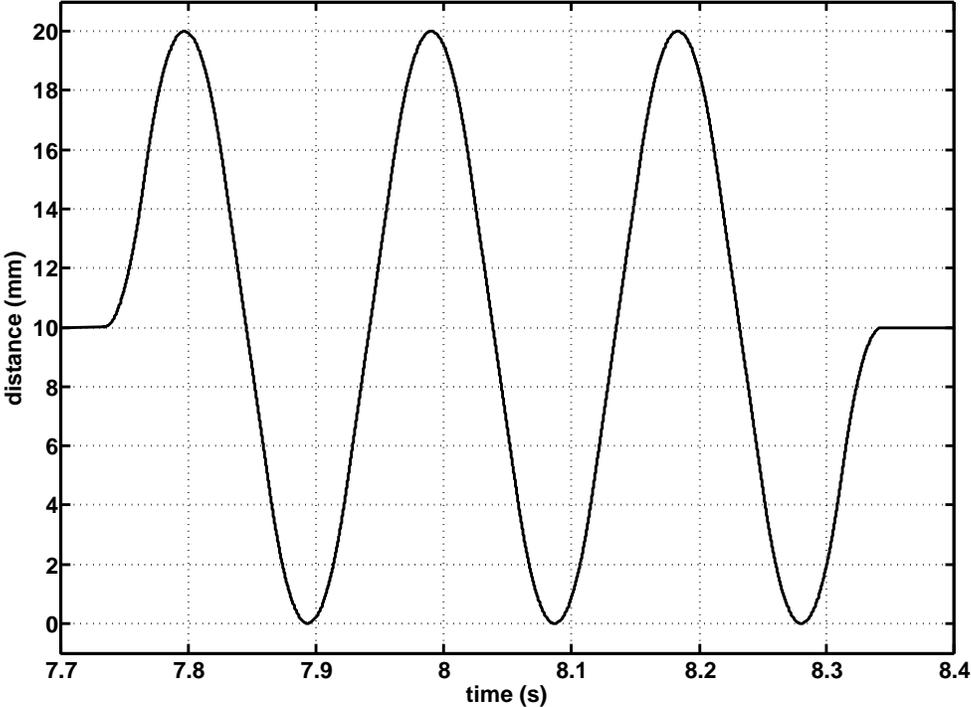


Figure 5.14: Planned trajectory for experiment 3 (fast linear oscillating motion)

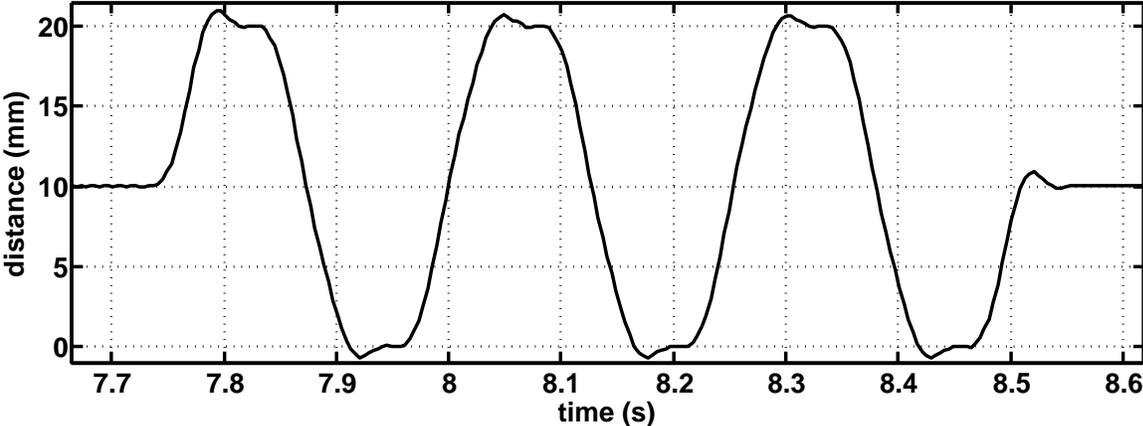


Figure 5.15: Actual trajectory for the linear rail in positioning mode for experiment 3 (fast oscillating linear motion)

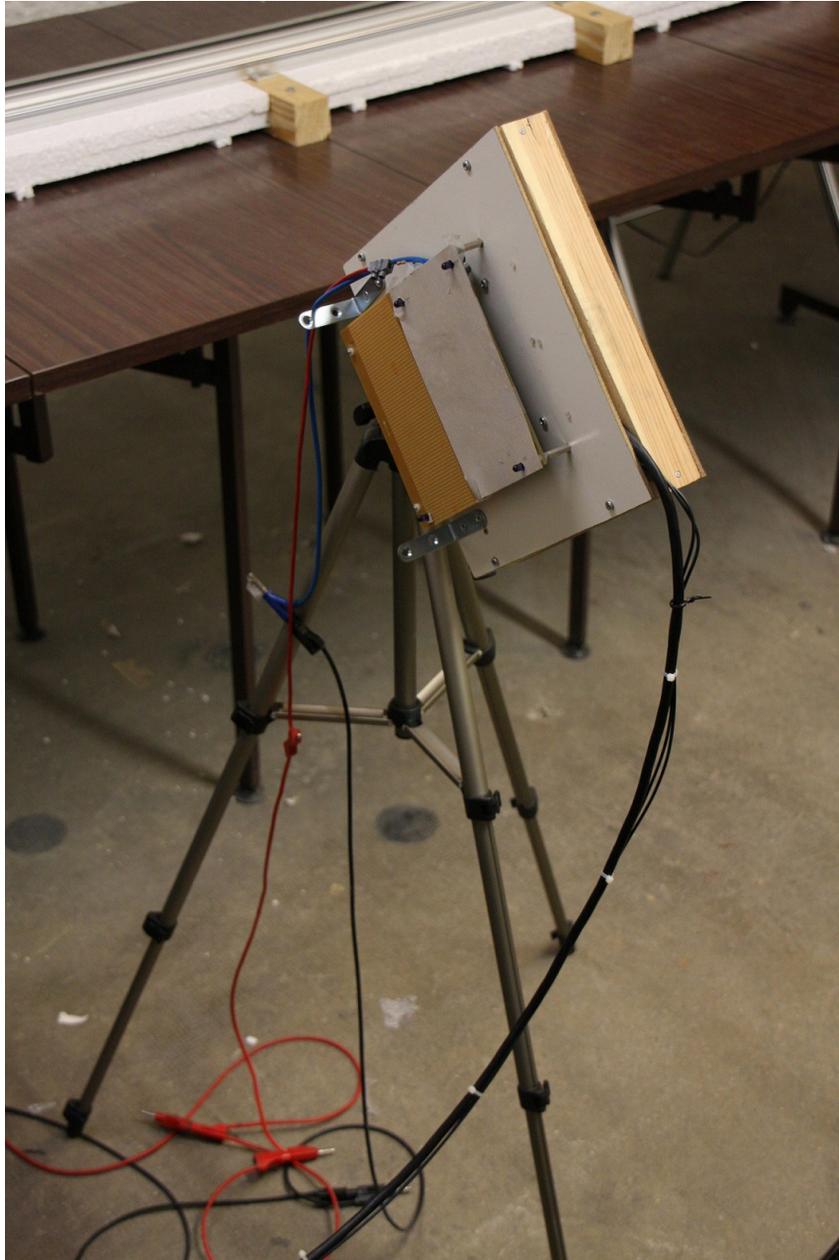


Figure 5.16: Sensor unit attached to tripod for calibration

5.3 Results

The algorithm can be executed offline with Simulink or in real time on an xPC Target. The experimental results shown in this Section have all been obtained in real-time. To compare different algorithms under the same conditions, the sensor data from the real-time run was saved and then used to re-run a different algorithm with the same data sequence. Data association works correctly if the state estimation is not too far from the truth; for this, the position has to be initialized with values which are approximately less than 20mm from the true position.

Most of the results in this Section were obtained with the Right-Invariant Extended Filter presented in Section 4.3.4.3. We chose this filter over the left-invariant one (see Section 4.3.4.4) because it expresses all errors in camera coordinates and this seems sensible since we are interested in the Sensor Unit motion expressed in the camera frame. However, our analysis has not shown any significant difference between estimations from the Right or Left-Invariant EKF.

5.3.1 General Observations

5.3.1.1 Position and Orientation

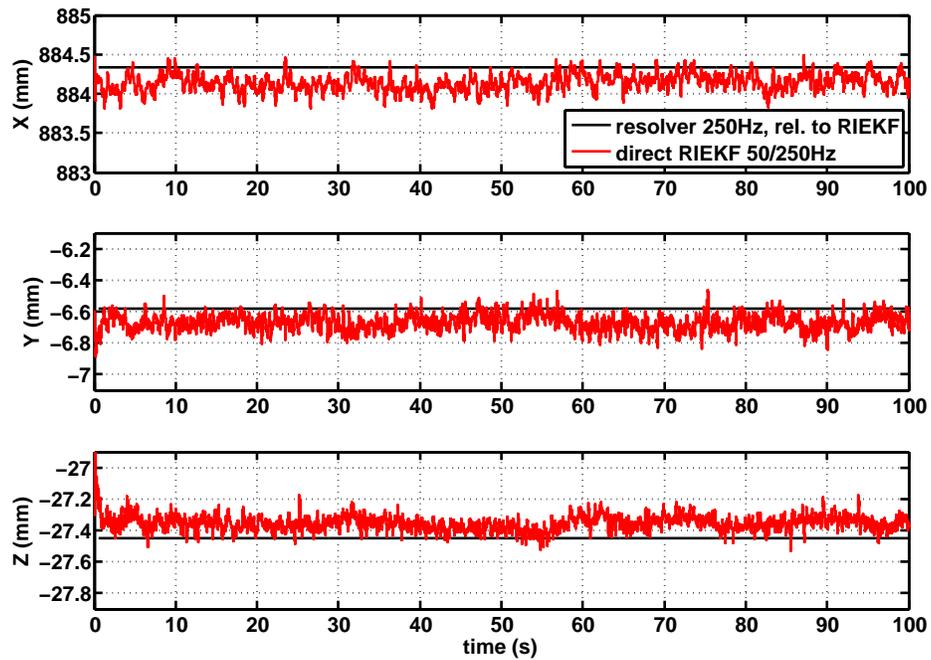
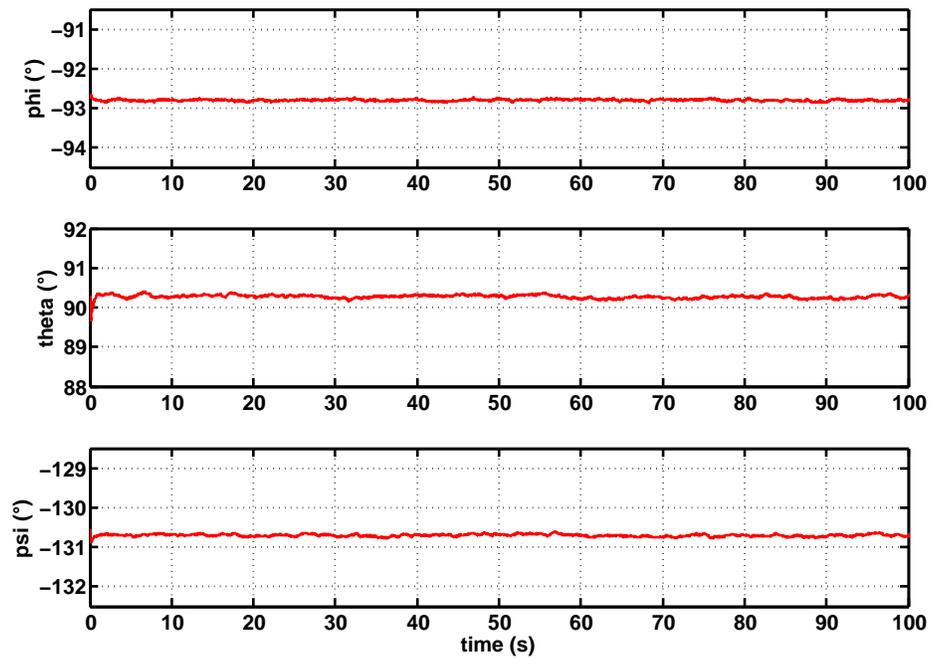
Figure 5.17 shows the estimated position and the Euler angles calculated from the estimated quaternions for the static case in experiment 1. The position is compared to the resolver measurement which shows that the RIEKF estimates the position correctly. We do not have a reference for the orientation; we can only say that the RIEKF estimates a constant orientation which is correct.

Figures 5.18 and 5.19 show the estimated position and Euler angles resp. for the linear motion in experiment 2. A comparison with the rail resolver measurement shows that the RIEKF estimates the motion correctly, except for occasional small deviations at the beginning or end of the motion. The RIEKF also detects a change in orientation, but again we do not have a reference to compare this estimation to.

5.3.1.2 Filtering of Inertial Measurements

The RIEKF filters noisy measurements. Figures 5.20 and 5.21 show the measured inertial data for experiment 1. All the signals are noisy and the accelerometer signals also show numerous peaks. These peaks and most of the noise are filtered by the RIEKF in the position and orientation as illustrated in Figure 5.17.

Figure 5.22 show the accelerometer data for experiments 2 and 3, illustrating what kind of signals the IMU measures for these motions.

(a) Position: X coordinate (depth), horizontal Y coordinate, vertical Z coordinate

(b) Euler angles

Figure 5.17: RIEKF estimation of position and Euler angles (calculated from estimated quaternions) for experiment 1 (static)

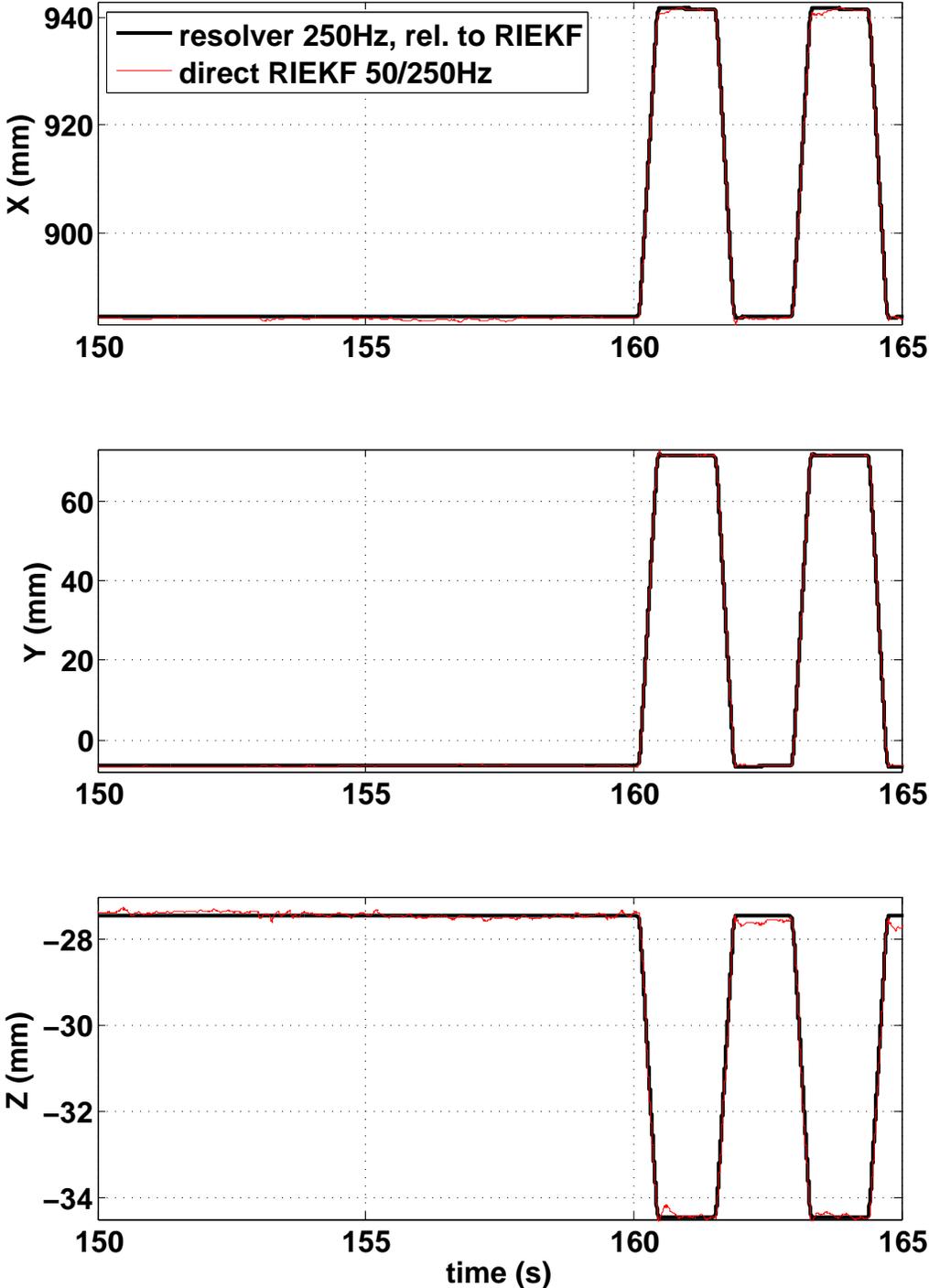


Figure 5.18: RIEKF position estimation for experiment 2 (linear motion)

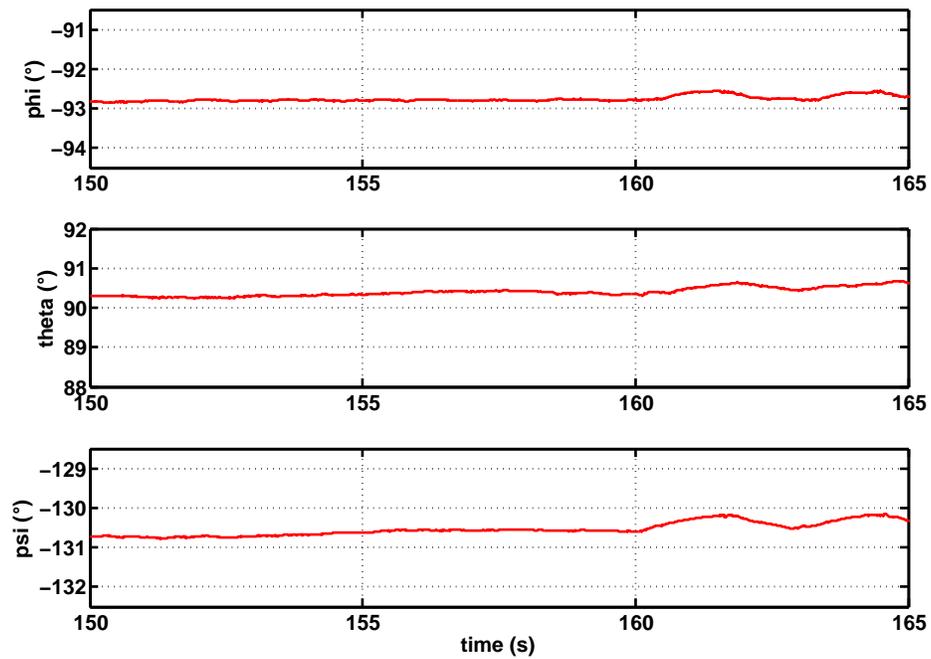


Figure 5.19: Euler angles calculated from quaternions estimated by the RIEKF for experiment 2 (linear motion)

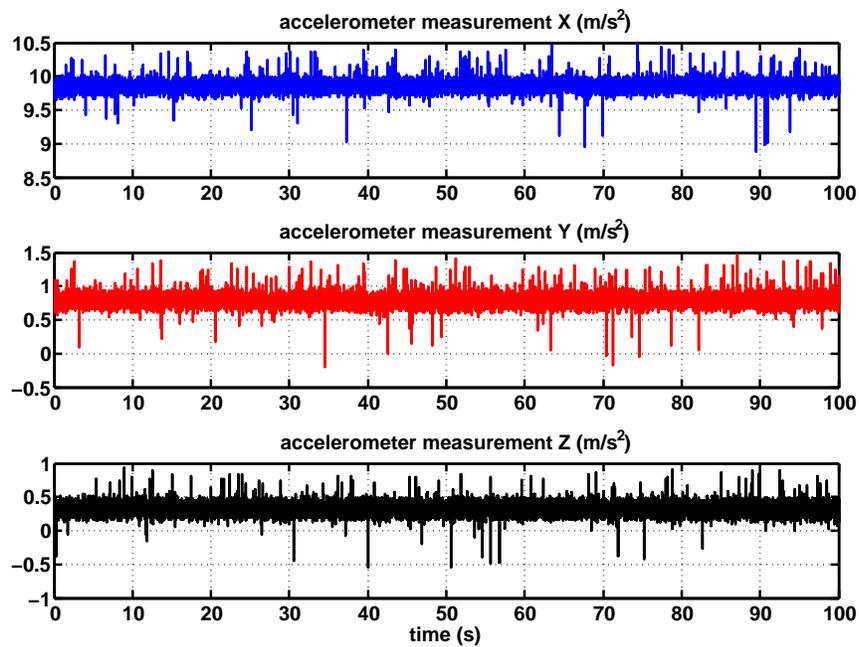


Figure 5.20: Accelerometer measurements for experiment 1

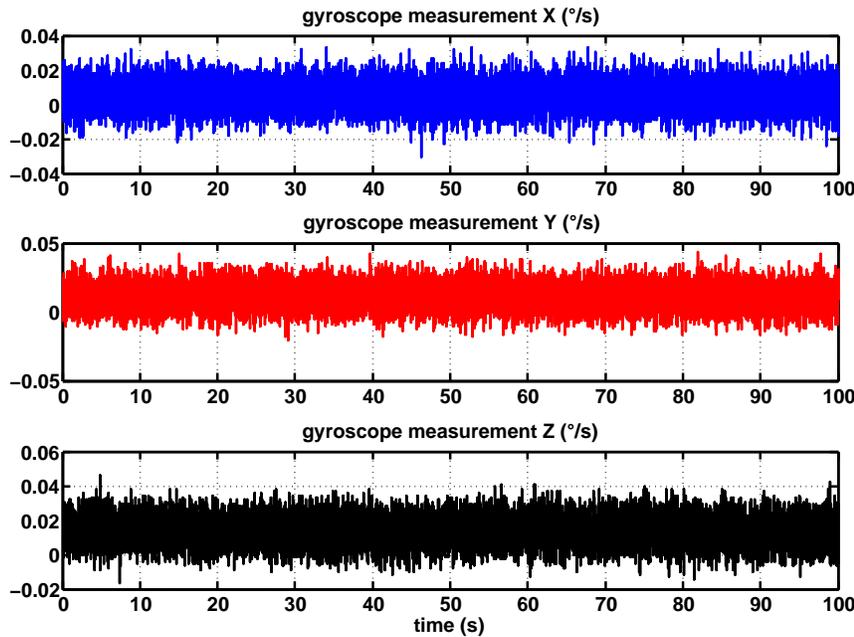


Figure 5.21: Gyroscope measurements for experiment 1

5.3.1.3 Bias estimation

The estimated accelerometer and gyroscope biases should theoretically be identical to the true biases present in the sensors. In fact, the estimated biases might also absorb parameter errors as mentioned in Section 4.4. The additional error terms can be constant or depend on the Sensor Unit position and/or orientation, implying that the estimated biases might change when the Sensor Unit moves although the true bias stays constant.

Figure 5.23 shows the estimated biases for an experiment in which the Sensor Unit was kept static for 150s and was then moved along the rail (several runs of the trajectories from experiments 2 and 3). Since the accelerometer biases need longer than 150s to converge, they have not been initialized with 0 but with their final values. The gyroscope biases have been initialized with 0.

The gyroscope biases converge quickly and stay almost constant during the Sensor Unit motions. The largest variation of about 7% occurs in the bias X . The accelerometer biases are affected more strongly by the Sensor Unit motion. They vary by 9% for the X axis, by 5% for the Y axis and by 15% for the Z axis.

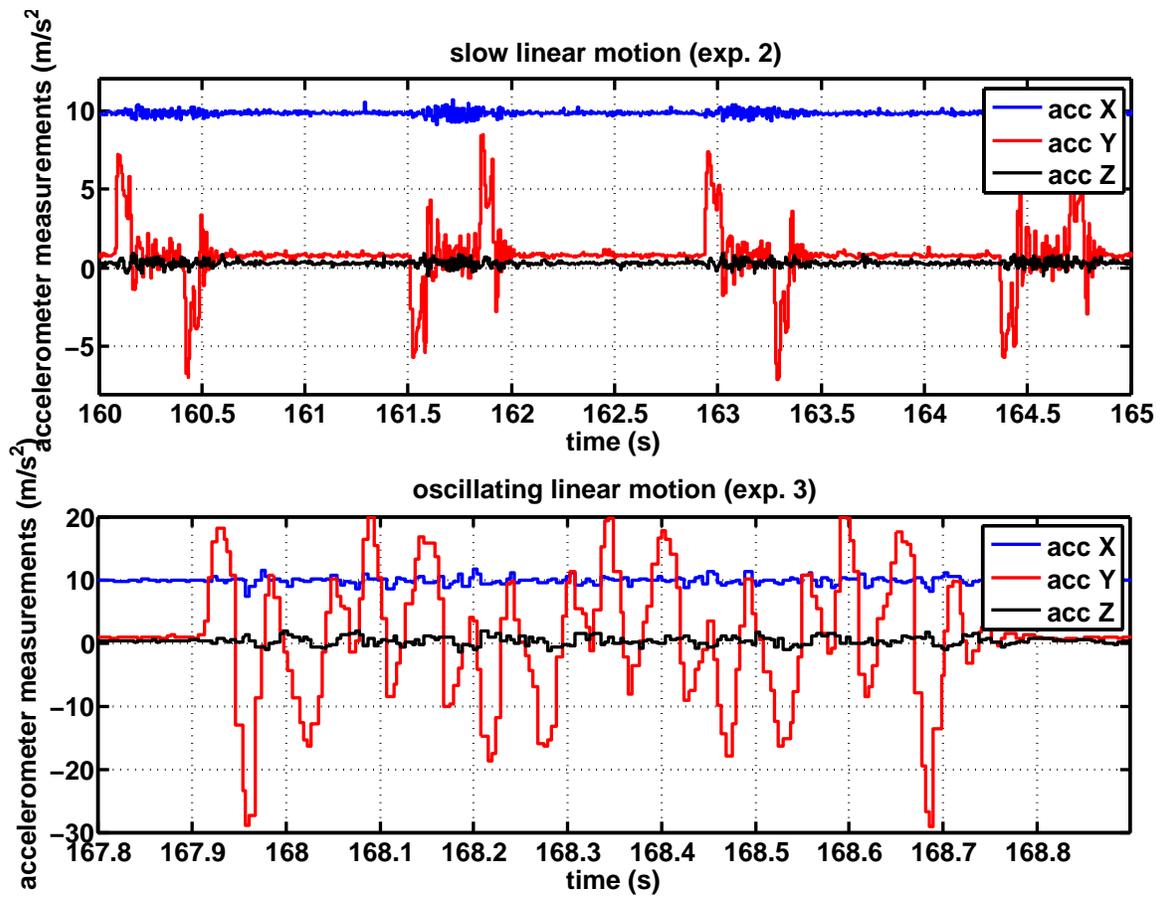


Figure 5.22: Accelerometer measurements for experiment 2 (above) and experiment 3 (below)

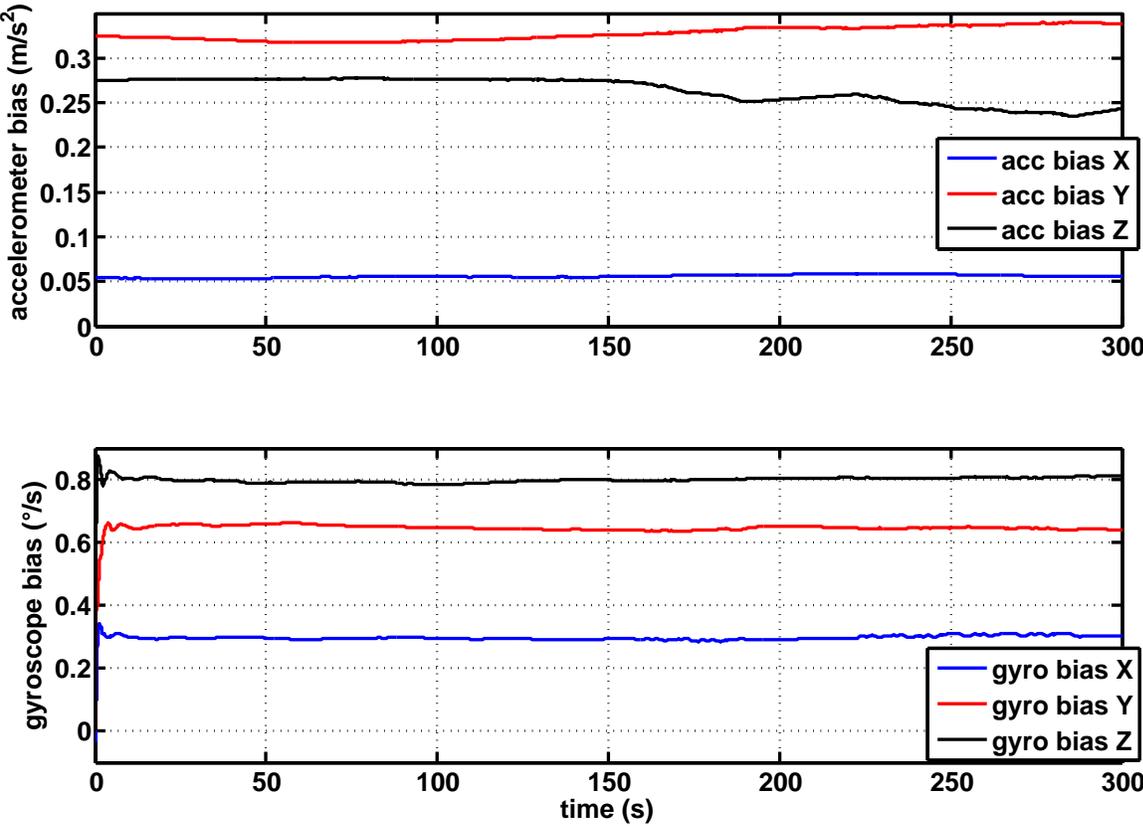


Figure 5.23: Estimated accelerometer and gyroscope biases

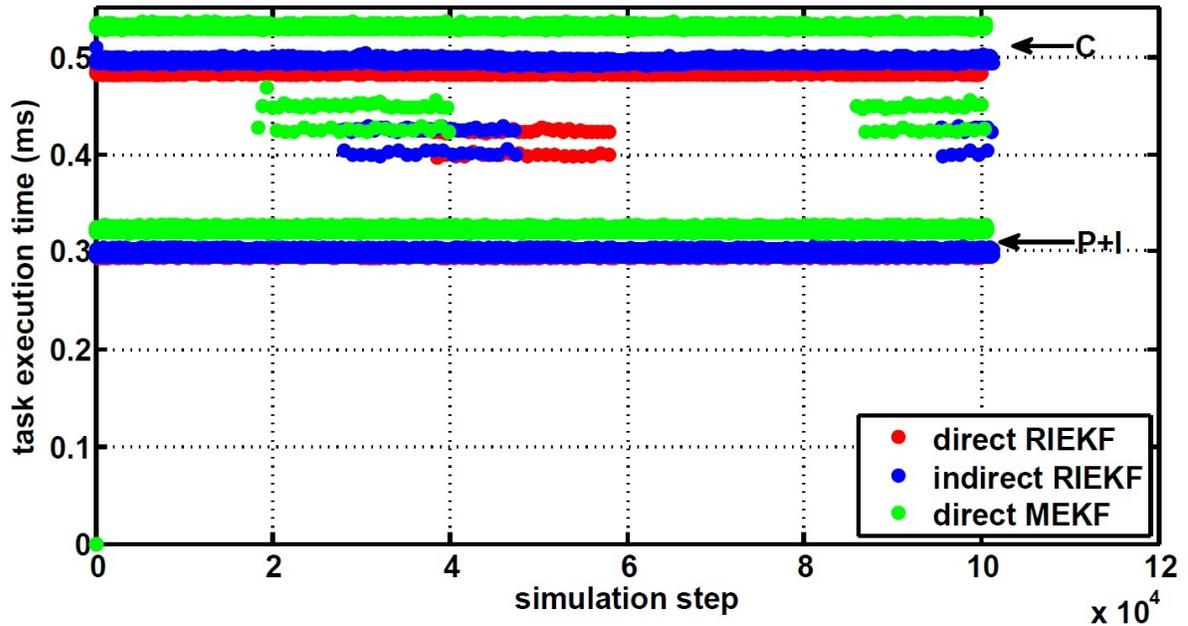


Figure 5.24: Real-time execution times for direct and indirect RIEKF and direct MEKF

5.3.1.4 Latency

The real-time implementation with xPC Target permits us to study the computational cost of a data fusion algorithm. The xPC Target application runs with a step of 1ms and measures the execution time for each step, called target execution time (TET). For an algorithm to run in real-time, the TET for each step has to be below 1ms. The TET measures the algorithm latency because it gives the time between a new measurement and the corresponding filter output.

Our first goal is to show that the direct approach reduces latency compared to the indirect approach. For this we implemented an indirect RIEKF and compared the TET for the direct and indirect RIEKF.

The second goal is to show that the RIEKF presented in Section 4.3.4.3 has a smaller latency than the MEKF from Section 4.3.3 because we consider that it is computationally less complex since it takes into account system symmetries.

Figure 5.24 illustrates task execution times for the direct and indirect RIEKF and the direct MEKF. The upper lines (marked "C") indicate times for corrections. The lower lines (marked "P+I") and the intermediate points represent times for prediction and integration steps (note that the red line for direct RIEKF is masked almost completely by the blue line for the indirect RIEKF). The direct RIEKF yields the smallest execution time, thus it has the lowest latency of the three algorithms. However, the differences

between the algorithms are too small to be significant. A more precise measurement would be necessary to evaluate latencies.

5.3.1.5 Discussion

In this Section, we made general observations about the RIEKF estimations. We showed that the optical-inertial tracking system with the RIEKF can estimate the motion of the Sensor Unit while filtering the noise inertial measurements. Gyroscope biases converge quickly and stay constant during the motion of the Sensor Unit, which is in accordance with our model. The accelerometer biases however depend on the Sensor Unit motion which is probably due to parameter errors. Finally, we also studied the latency of the direct RIEKF and compared to the indirect RIEKF and to the MEKF.

5.3.2 Precision and Accuracy

Measurement systems are often evaluated regarding their precision and accuracy, and these terms are also applied to indirect measurements where values are obtained by some computation from measured data [Miller et al., 2009]. Precision is the degree to which repeated measurements under unchanged conditions show the same results [Miller et al., 2009]. For our optical-inertial tracking system for example, we would like to study the precision of the position estimation, that is by how the estimated position varies when the true position is constant. Accuracy describes how close a measurement is to its true value [Miller et al., 2009]. In our case, accuracy describes whether the estimated position is equal to the true position or whether it is offset by a systematic error.

5.3.2.1 Precision (static case)

We study the precision using data from experiment 1 (static case) from $t=10s$ to $t=100s$ (we leave out the first 10s because the filter has not converged yet during this time). Figure 5.17(a) shows the position estimation and Figure 5.17(b) the Euler angles calculated from the estimated quaternions.

The position has the following mean and standard deviation:

$$\overline{{}^C\hat{p}} = \begin{bmatrix} 884.1436 \\ -6.6690 \\ -27.3574 \end{bmatrix} mm \quad std({}^C\hat{p}) = \begin{bmatrix} 0.0996 \\ 0.0452 \\ 0.0404 \end{bmatrix} mm .$$

The position varies from

$${}^C\hat{p}_{min} = \begin{bmatrix} 883.8088 \\ -6.8435 \\ -27.5295 \end{bmatrix} mm \quad \text{to} \quad {}^C\hat{p}_{max} = \begin{bmatrix} 884.4879 \\ -6.4642 \\ -27.1763 \end{bmatrix} mm$$

which gives a difference of

$$\Delta^C \hat{p} = \begin{bmatrix} 0.6790 \\ 0.3794 \\ 0.3532 \end{bmatrix} mm .$$

Both the standard deviation and the difference between minimum and maximum values show that the precision is approximately the same in Y and Z and that it is twice as good in these directions than depth precision (coordinate X).

The angles have the following mean and standard deviation:

$$\begin{bmatrix} \bar{\phi} \\ \bar{\theta} \\ \bar{\psi} \end{bmatrix} = \begin{bmatrix} -1.6196 \\ 1.5756 \\ -2.2812 \end{bmatrix} \text{degrees} \quad std\left(\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}\right) = \begin{bmatrix} 0.3628 \cdot 10^{-3} \\ 0.6661 \cdot 10^{-3} \\ 0.5064 \cdot 10^{-3} \end{bmatrix} \text{degrees} .$$

The angles vary from

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}_{min} = \begin{bmatrix} -1.6208 \\ 1.5735 \\ -2.2825 \end{bmatrix} \text{degrees} \quad \text{to} \quad \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix}_{max} = \begin{bmatrix} -1.6185 \\ 1.5774 \\ -2.2793 \end{bmatrix} \text{degrees}$$

which gives a difference of

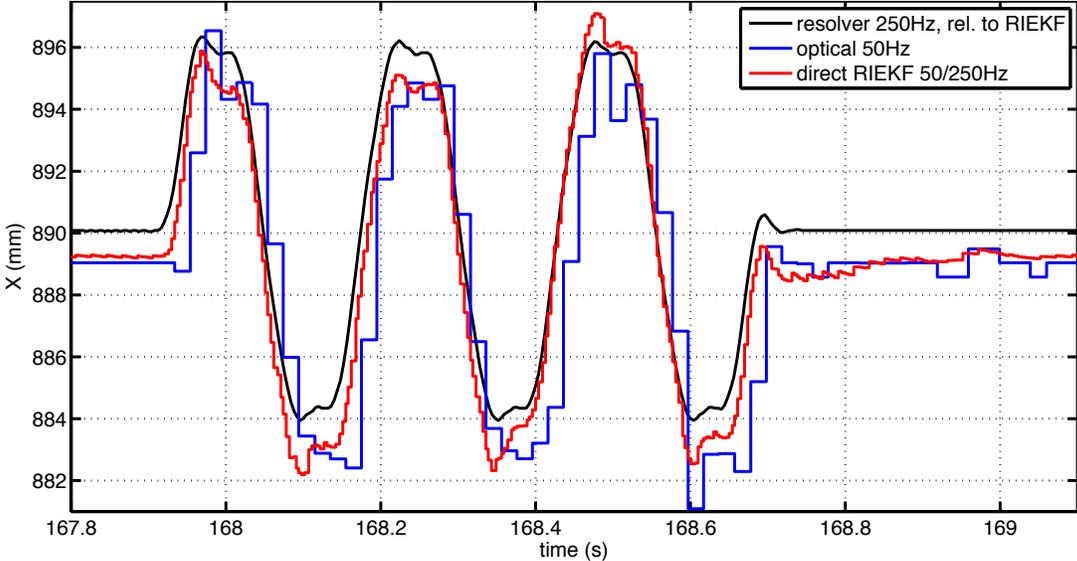
$$\begin{bmatrix} \Delta\phi \\ \Delta\theta \\ \Delta\psi \end{bmatrix} = \begin{bmatrix} 0.0023 \\ 0.0038 \\ 0.0032 \end{bmatrix} \text{degrees} .$$

5.3.2.2 Accuracy

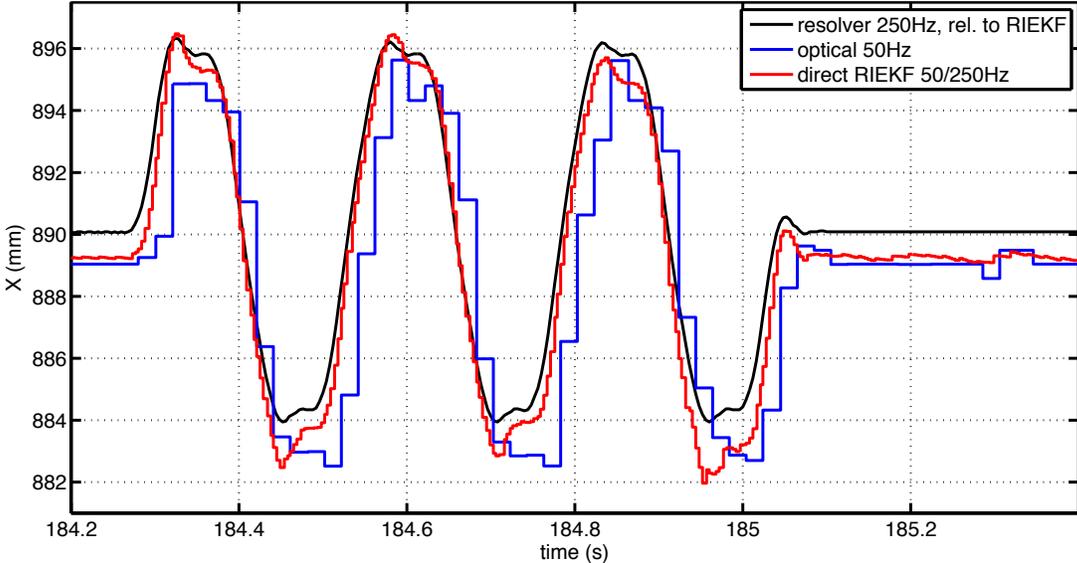
Since we do not have an absolute reference for the Sensor Unit position and orientation, we cannot evaluate the accuracy of the optical-inertial system with the RIEKF.

5.3.3 High-Bandwidth Tracking

Figures 5.25-5.27 show the position estimated by the RIEKF (with an optical frequency of 50Hz and an inertial frequency of 250Hz) for two repeats of the oscillation with 1cm amplitude during experiment 3. Figure 5.25 depicts the X coordinate (which represents depth), Figure 5.26 the horizontal Y coordinate and Figure 5.25 the vertical Z coordinate of the position. The RIEKF follows the oscillating motion well. There are a few deviations, namely at the points where the rail carriage changes direction. At these inversions, there might be a problem with the resolver measurement which is used as ground truth. Since it is only an indirect measurement counting the number of turns of the motor, it possibly does not give the correct value at inversions due to transmission errors. Thus we cannot

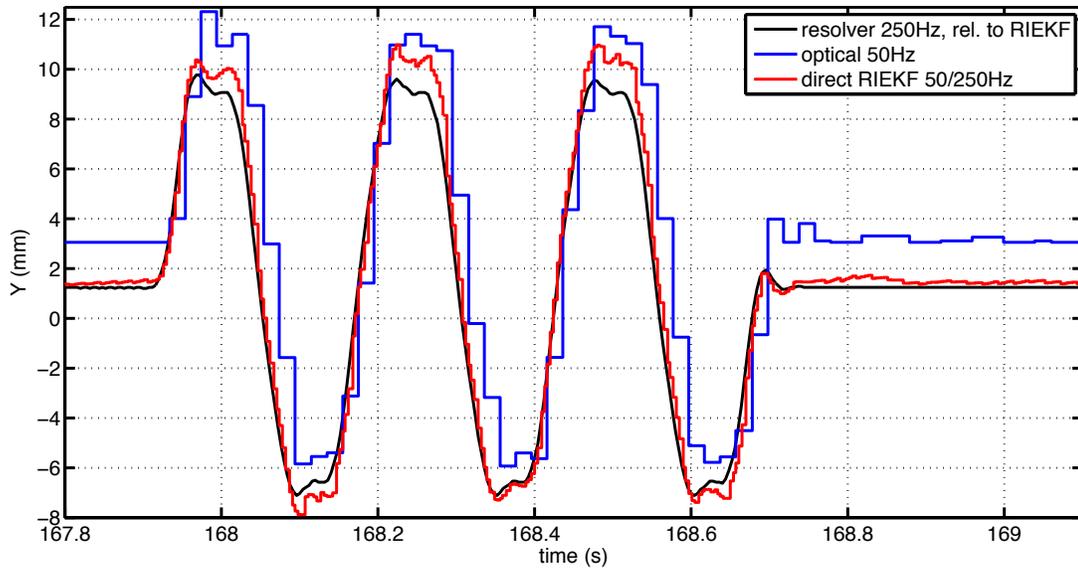


(a) first run

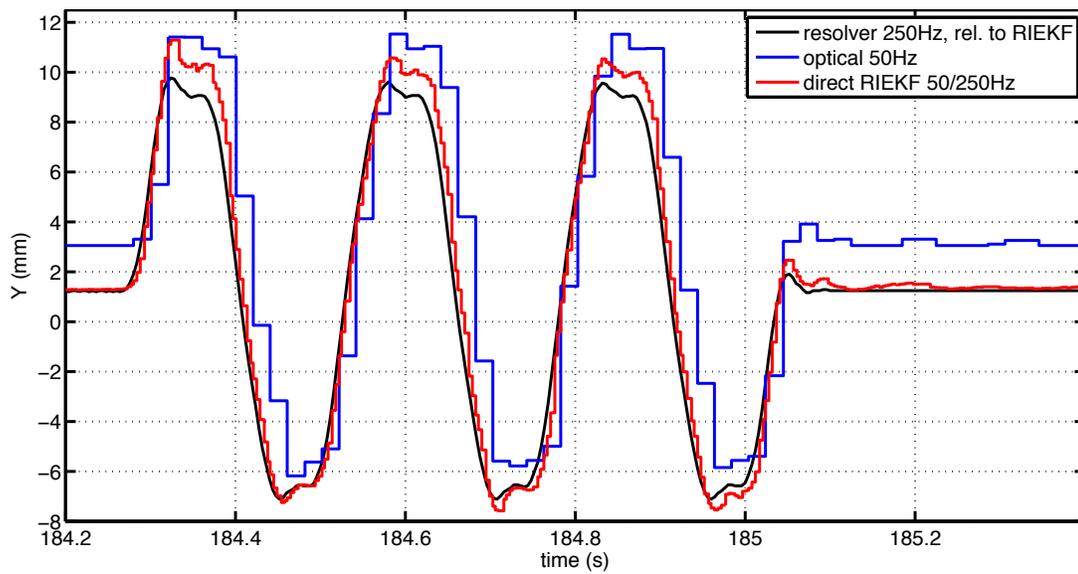


(b) second run

Figure 5.25: X position coordinate for oscillation in experiment 3

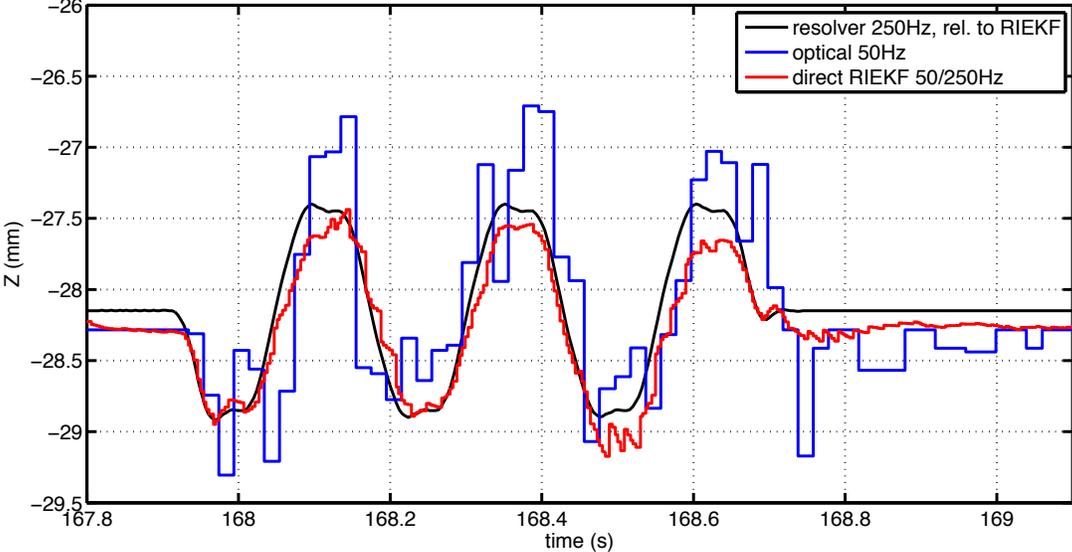


(a) first run

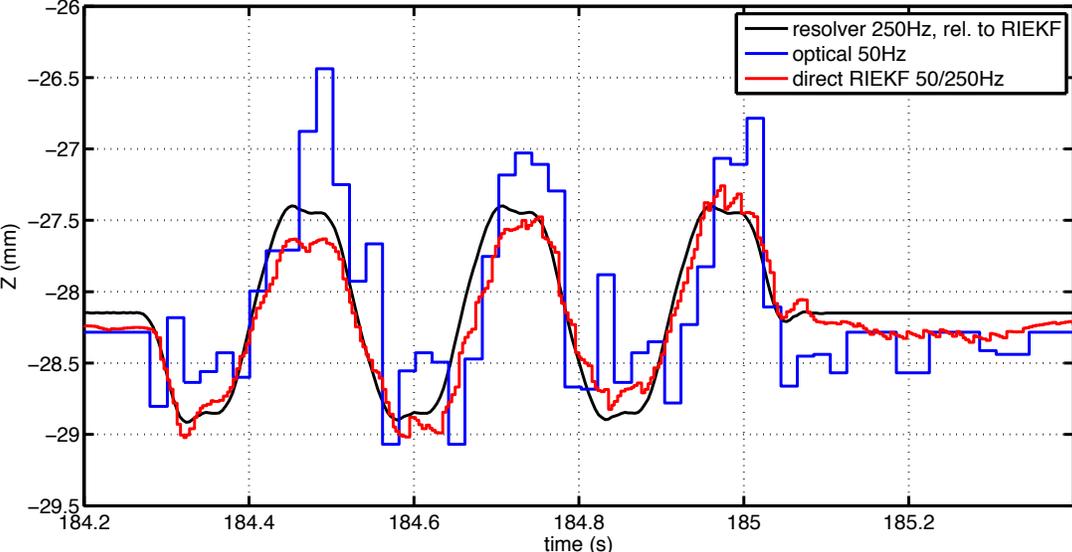


(b) second run

Figure 5.26: Y position coordinate for oscillation in experiment 3



(a) first run



(b) second run

Figure 5.27: Z position coordinate for oscillation in experiment 3

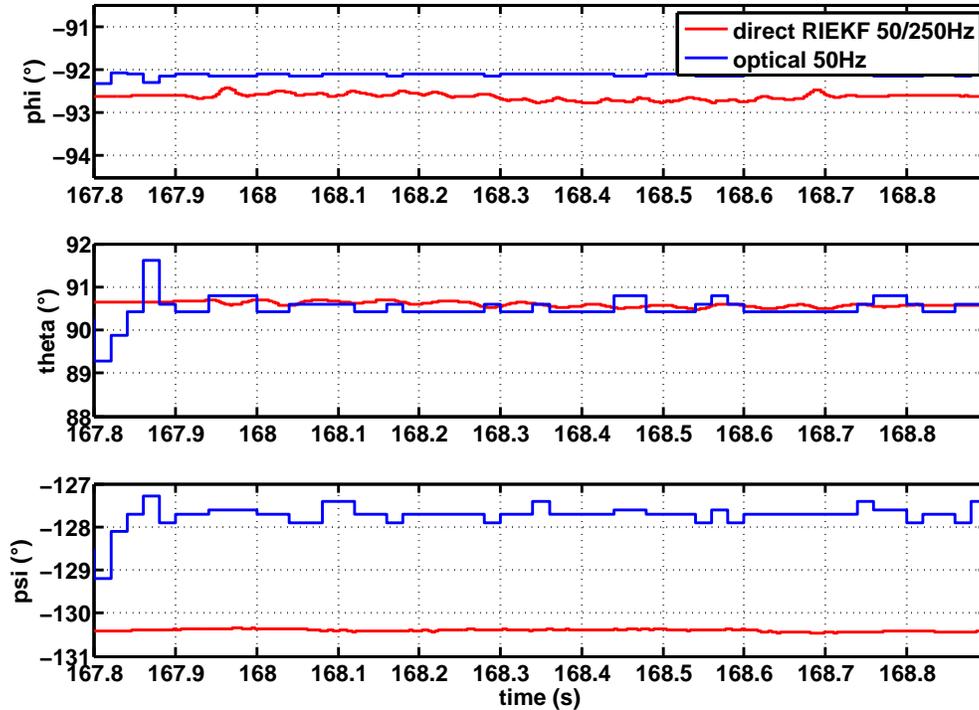


Figure 5.28: Angles for oscillation in experiment 3

judge the RIEKF estimation at these points because we might not have a correct ground truth.

Having compared the RIEKF estimation to the rail reference, we next look at it in contrast to optical tracking. The optical position is determined as described in Section 5.1.6.2 and plotted in Figures 5.25-5.27. We start by examining the Y coordinate in Figure 5.26 which is the axis in which most of the motion took place. The optical position gives only a very imprecise approximation of this curve. This is even more evident when looking at the Z coordinate in Figure 5.27. The Sensor Unit moved by only approximately 1.5mm in this axis and the optical system does not see this motion correctly. This shows that the optical-inertial tracking system with its high bandwidth is more suitable than an optical tracking system to follow fast motion.

Figure 5.28 illustrate the Euler angles estimated by the RIEKF and from optical tracking for the 1cm oscillation in experiment 3. We cannot judge the accuracy of this estimation since we do not have a reference but the optical Euler angles obviously display

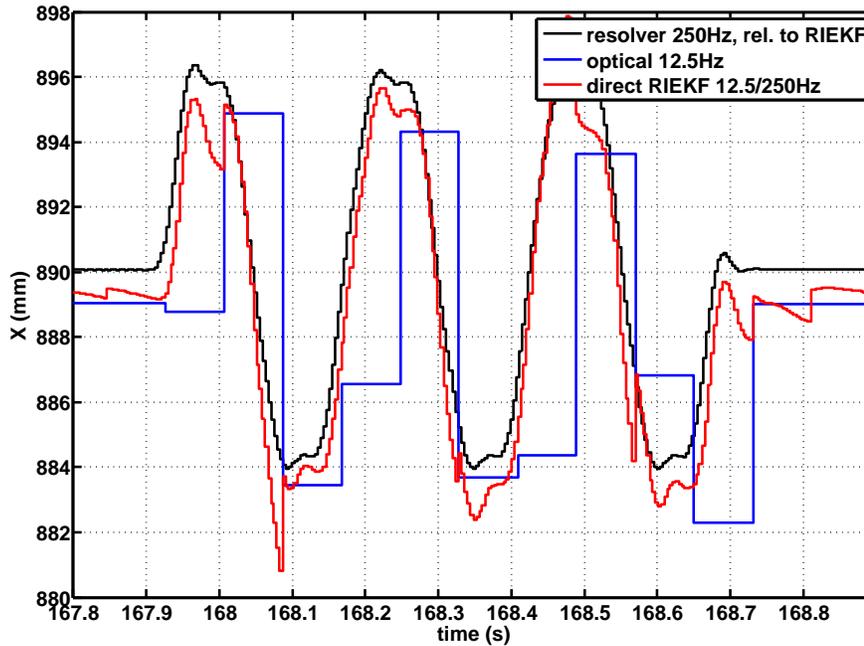


Figure 5.29: X position coordinate for oscillation in exp. 3 with an optical rate of 12.5Hz

sudden changes which is not correct.

Having determined that the optical-inertial tracking system with an inertial sample rate of 250Hz and an optical sample rate of 50Hz can in fact follow the fast motion of the Sensor Unit we would now like to study if it is possible to further augment the system bandwidth. The experimental setup described in Section 5.1 does not permit us to increase the inertial sample rate beyond 250Hz. Augmenting the inertial rate means augmenting the number of predictions between two corrections. Since we cannot increase the inertial rate we lower the optical rate to 12.5Hz instead. This gives a factor of 20 between the optical and inertia sensor rates and consequently 20 predictions are executed in the RIEKF between two corrections. Figures 5.29-5.31 show the position estimation from the RIEKF and optical tracking for this case. The RIEKF estimation is of course less good than for the case of a 50Hz optical rate, but it still detects the oscillating motion and does so much better than the optical tracking.

5.3.4 Influence of Sensor Unit Calibration

In Section 4.4.1.1 we studied the influence of Marker-Body transformation errors on the RIEKF estimation in theory. Now we look at the effect of a change of Marker-Body transformation on the RIEKF estimations obtained from real data and will

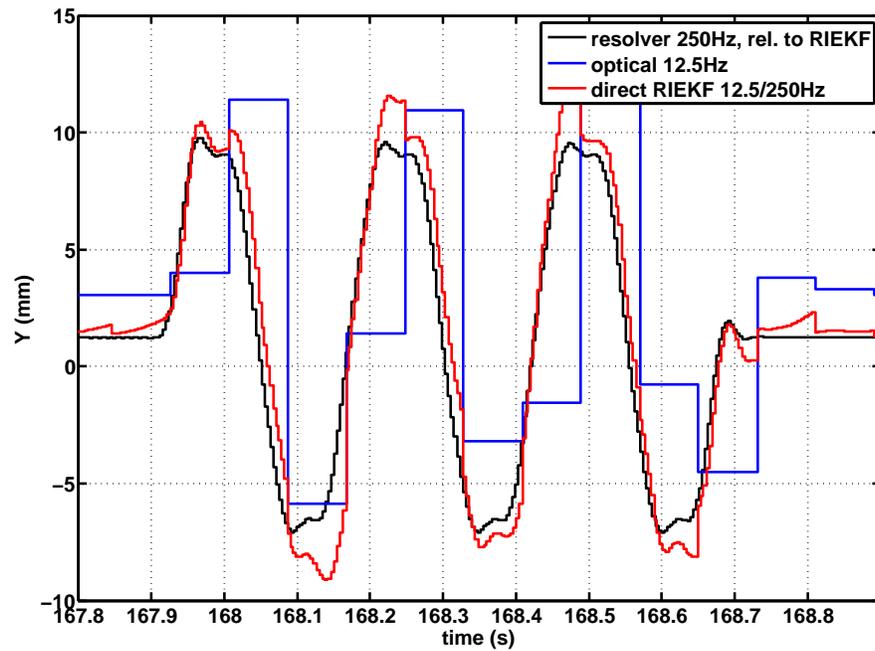


Figure 5.30: Y position coordinate for oscillation in exp. 3 with an optical rate of 12.5Hz

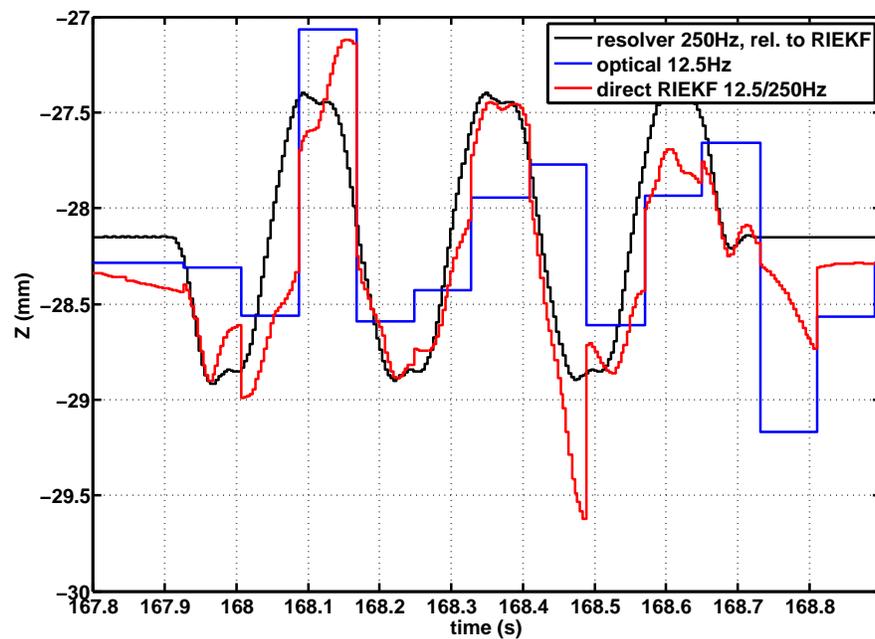


Figure 5.31: Z position coordinate for oscillation in exp. 3 with an optical rate of 12.5Hz

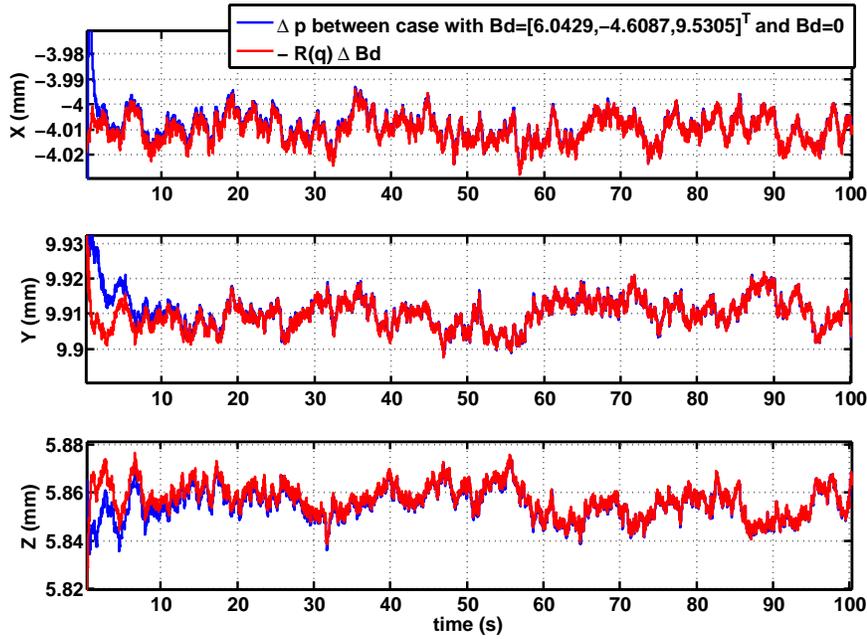


Figure 5.32: Difference of the position estimated with $Bd = [6.0429, -4.6087, 9.5305]^T$ and $Bd = 0$ (blue), and error $-R({}^{BC}\hat{q})\Delta Bd$ as calculated in the error analysis (red), static case

see that the theoretical values are confirmed. We compare the case of the marker model measured with a ruler (here, $Bd = 0$ and ${}^{MB}R = I_3$) and that with a Marker-Body transformation determined from calibration procedures. The different models are described in Section 5.4.4.

We start by looking at the effect of a change of the translation vector. Using different translation vectors Bd leads to a translation in the estimated Sensor Unit position as determined in the error analysis in Section 4.4.1.1. Figure 5.33 plots the difference between two positions for experiment 1, one estimated with

$$Bd = [6.0429, -4.6087, 9.5305]^T$$

and the other with $Bd = 0$. This figure also shows the term $-R({}^{BC}\hat{q})\Delta Bd$ where ΔBd is the difference between the two translation vectors considered here. The term $-R({}^{BC}\hat{q})\Delta Bd$ is the one given in Equation (4.78) for a translation error ΔBd . Since both curves overlap after the initialization phase, this illustrates that the theoretical result from the error analysis is confirmed when using experimental data. The error analysis was done for steady-state and consequently it is only true for the static case. This is illustrated in Figure 5.33 where the same variables are shown for the oscillating motion in experiment 3: in this non-static case, the two variables are not equal.

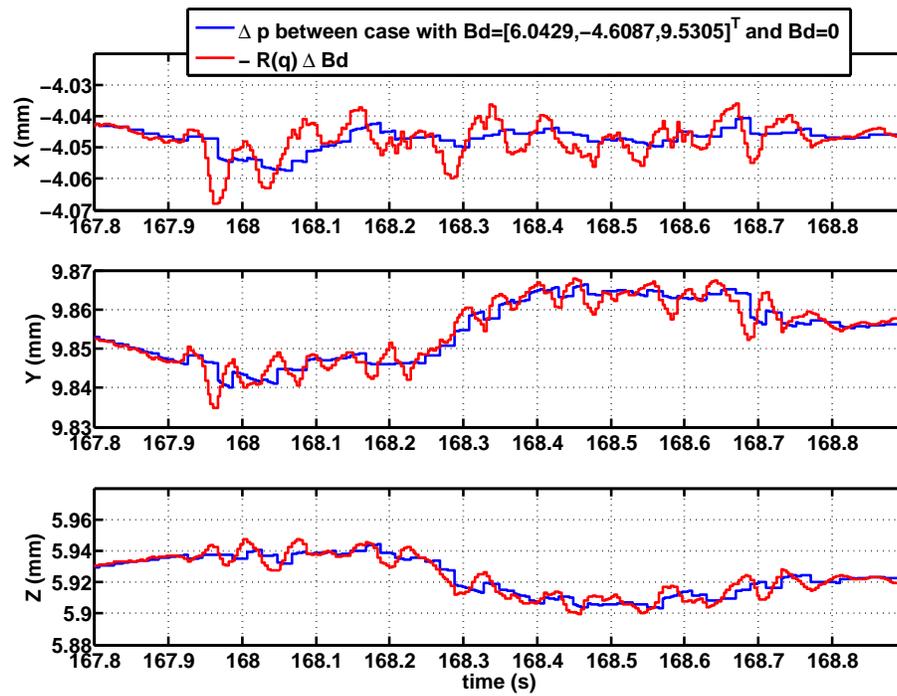


Figure 5.33: Difference of the position estimated with ${}^B d = [6.0429, -4.6087, 9.5305]^T$ and ${}^B d = 0$ (blue), and error $-R({}^{BC}\hat{q})\Delta {}^B d$ as calculated in the error analysis (red), during motion

We now move on the effect of the Marker-Body rotation matrix. Two different rotation matrices ${}^{MB}R$ lead to small changes in position and orientation as given in Equations (4.78) and (4.80). The main difference lies in the estimated accelerometer biases which will be illustrated in Section 5.4.4.2 about the calibration of the Marker-Body rotation.

5.4 Calibration

5.4.1 Optical System

Regarding the calibration of the optical system, we have to determine the camera intrinsic values, the rotation between the world and camera frame and the transformation between the two stereo cameras.

The optical system has to be calibrated only once since its parameters are constant. The rotation between world and camera frame and the stereo transformation have to be re-calibrated when the setup has been changed.

5.4.1.1 Camera

Each camera is calibrated using the method presented in Section 3.3.1 and a planar calibration target consisting of four LEDs. This gives the camera's intrinsic parameters.

For the left camera, we obtained

$$f_{u,L} = 1366.2208, \quad f_{v,L} = 1368.8532, \quad {}^S u_L = [542.6768, 366.07549]^T$$

and for the right camera

$$f_{u,R} = 1366.5736, \quad f_{v,R} = 1358.715, \quad {}^S u_R = [531.61743, 360.94375]^T .$$

All the parameters are in pixels.

Rotation between world and camera frame The orientation WCq of the camera frame C with respect to the world frame W has to be known in order to calculate ${}^CG = {}^WCq * {}^WG * {}^WCq^{-1}$ in the differential equation for the Sensor Unit velocity (2.8). We chose the following procedure to determine WCq .

- 1 The Sensor Unit is placed in such a way in front of the cameras that the marker plane is aligned with the vertical axis.
- 2 The orientation ${}^{BC}q$ is calculated using optical data.

3 Since the world and marker frames are aligned, the rotation matrix

$${}^M p = {}^{WM} R {}^W p$$

can be determined easily; the quaternion ${}^{WM} q$ is then deduced from ${}^{WM} R$.

4 ${}^{WC} q$ can then be calculated from ${}^{BC} q$:

$${}^{WC} q = {}^{BC} q * {}^{MB} q * {}^{WM} q$$

where the rotation ${}^{MB} q$ between marker and body frame has been determined by calibration (see Sensor Unit Calibration below).

5.4.1.2 Stereo rig

The stereo rig is calibrated according to the method [Bouguet, 2010] as described in Section 3.3.1.1. The procedure yielded the following stereo transformation:

$$R_{St} = \begin{bmatrix} 0.9698 & -0.2421 & -0.0283 \\ 0.2410 & 0.9698 & -0.0366 \\ 0.0363 & 0.0287 & 0.9989 \end{bmatrix}, \quad t_{St} = \begin{bmatrix} 42.7094 \\ -374.9868 \\ -2.4414 \end{bmatrix} mm$$

5.4.2 Accelerometers

Axis misalignment, scale factors and biases were calibrated with an iterative algorithm [Dorveaux et al., 2009] as described in Section 3.3.2.

For one calibration procedure we obtained a matrix

$$A = \begin{bmatrix} 1.0003 & -0.0022 & -0.0001 \\ 0.0032 & 0.9997 & -0.0008 \\ 0.0006 & 0.0011 & 0.9977 \end{bmatrix}$$

containing scale factor and misalignments and a bias vector

$$B = \begin{bmatrix} 32.2661 \\ 668.4856 \\ 70.2366 \end{bmatrix} \frac{mm}{s^2} .$$

Note that these values might change every time the sensors are turned on.

5.4.3 Gyroscopes

Regarding the gyroscopes, we do not consider scales or misalignment but only biases. The bias consists of a constant part ω_b and a term depending on the specific acceleration, called g-dependent bias [Groves, 2008]: $B_g^B a$. Most of this g-dependent bias can be compensated in the IMU used here [ADIS16355, 2009], but a small dependency remains. To determine the constant bias term ω_b , the Sensor Unit is rotated into several different orientations and then kept static for several second. The gyroscope measurements are averaged over 10s to give $\bar{\omega}_i$ for each orientation i . Accelerometer measurements are compensated for misalignment, scale factors and biases as described in the previous Section and averaged over 10s to give \bar{a}_i . The g-dependent bias matrix B_g and the gyroscope bias are calculated using linear least squares:

$$\begin{aligned}
 & B_g \bar{a}_i + \omega_b = \bar{\omega}_i \\
 \Leftrightarrow & \begin{bmatrix} g1 \\ g2 \\ g3 \end{bmatrix} \bar{a}_i + \omega_b = \bar{\omega}_i \\
 \Leftrightarrow & \begin{bmatrix} \bar{a}_i^T & 0 & 0 & \parallel \\ 0 & \bar{a}_i^T & 0 & \parallel \\ 0 & 0 & \bar{a}_i^T & \parallel \end{bmatrix} \begin{bmatrix} g1^T \\ g2^T \\ g3^T \\ \omega_b \end{bmatrix} = \bar{\omega}_i
 \end{aligned}$$

5.4.4 Sensor Unit

The calibration of the Sensor Unit consists in determining the optical marker positions in the Body frame which is the frame of the inertial sensors. In a first step, the positions were measured with a ruler as given in Section 5.4.4.1. These measurements are inaccurate due to two reasons: Firstly, it was impossible to perfectly align the IMU and the circuit board containing the optical markers. Secondly, the center of the IMU frame inside the casing is not given in the datasheet and could only be estimated. This explains the need for a calibration of the Marker-Body transformation as described in Section 5.4.4.2.

The marker model and the Marker-Body transformation have to be determined only once since they only depend on the Sensor Unit geometry and are constant.

5.4.4.1 Marker Model

Positions ${}^M m_i$ are measured with a ruler:

$$[{}^M m_1 {}^M m_2 {}^M m_3 {}^M m_4] = \begin{bmatrix} -30 & 48.5 & 31 & -122 \\ -63.5 & -63.5 & 78.5 & 78.5 \\ -58 & -58 & -58 & -60 \end{bmatrix} mm$$

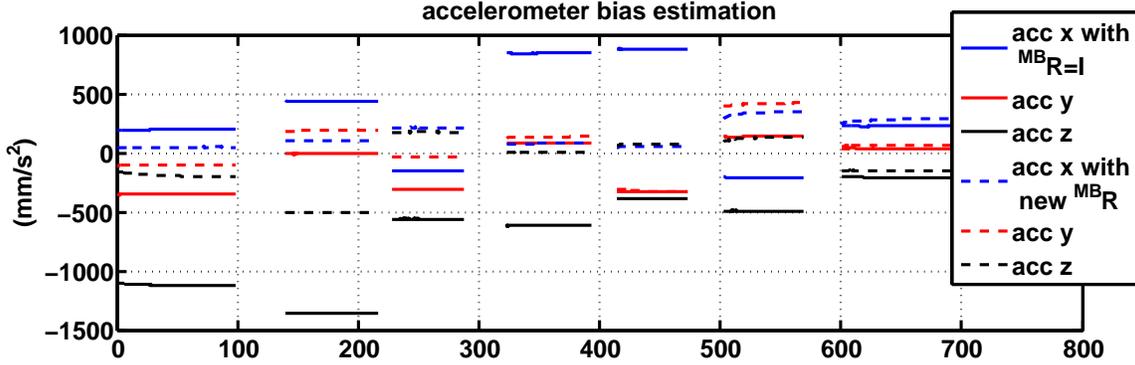


Figure 5.34: Estimated accelerometer biases for different orientations, with ${}^{MB}R = I$ (solid lines) and with ${}^{MB}R$ from calibration (dashed lines).

5.4.4.2 Marker-Body Transformation

The Marker-Body transformation (${}^{MB}R, {}^Bd$) has to be known in order to transform the marker model to body coordinates:

$${}^Bm_i = {}^{MB}R^M m_i + {}^Bd$$

Estimation as Additional State in the RIEKF We adapted the idea of adding the transformation as additional states as presented in [Mirzaei and Roumeliotis, 2008] to the outside-in case. That is we added the quaternion ${}^{MB}q$ and Bd to the state of the system to be estimated and changed the RIEKF accordingly. But this filter did not converge either in simulation or with real data.

Method Using RIEKF Bias Estimates The Marker-Body rotation was estimated with the method described in Section 4.4.2. For this calibration, the Sensor Unit was rotated into seven different orientation as described in Section 5.2.4. For each orientation, the system state was estimated using the RIEKF. The measured calibrated accelerometer measurements and the estimated accelerometer biases were then averaged over 10s to give \bar{a}_m and ${}^B a_b$ which are the values used in the calibration procedure.

The Marker-Body rotation found through this calibration is

$${}^{MB}q = [0.9987 \quad -0.0079 \quad 0.0478 \quad -0.0143]$$

which gives a rotation matrix

$${}^{MB}R = \begin{bmatrix} 0.9950 & 0.0277 & 0.0959 \\ -0.0292 & 0.9995 & 0.0146 \\ -0.0954 & -0.0173 & 0.9953 \end{bmatrix}$$

Figure 5.34 shows the biases estimated by the RIEKF for the 7 different static orientations for ${}^{MB}R = I$ (before calibration) and for the matrix ${}^{MB}R$ determined by calibration using these 7 orientations. In call cases, "calibrated" accelerometer measurements have been used in which the biases have been removed; thus the biases estimated by the RIEKF should be zero.

Firstly, we see that the estimated biases depend on the orientation which is due to parameter errors as discussed in Section 4.4.1. Obviously, the goal of calibration procedures would be to eliminate these errors from the model in order to remove parameter error terms from the accelerometer biases. With correct parameters, the estimated biases would be equal to the true sensor biases. Secondly, and most importantly, the Figure shows that with the calibrated ${}^{MB}R$, the biases are reduced for almost all orientations. They are not equal to zero, probably due to other calibration errors which have not yet been compensated, but this result still shows the positive effect of this calibration method.

Least Squares Approach Using Optical Measurements We adapted the calibration in [Kim and Golnaraghi, 2004] which uses pose information from an optical tracking system as ground truth. [Kim and Golnaraghi, 2004] estimates biases, gains and axes misalignments for a triad of gyroscopes and accelerometers. We use this approach to estimate the transformation only because biases, gains and misalignments have already been compensated for by the accelerometer calibration.

The marker positions are determined by triangulation:

$$\begin{aligned} {}^C m_i &= {}^C p + {}^{BC} q * {}^B m_i * {}^{BC} q^{-1} \\ &= \underbrace{{}^C p + {}^{BC} q * {}^B d * {}^{BC} q^{-1}}_{=: {}^C r} + \underbrace{({}^{BC} q * {}^{MB} q)}_{=: {}^{MC} q} * {}^M m_i * ({}^{BC} q * {}^{MB} q)^{-1} \end{aligned}$$

${}^C r$ is the origin of the marker frame in camera coordinates and ${}^{MC} q$ is the Marker-body rotation. ${}^C r$ and ${}^{MC} q$ are determined by one of the solutions of the Absolute Orientation Problem (see Section 3.1.2.2), in our case by Walker's method [Walker et al., 1991].

We differentiate the Marker-Camera quaternion to get the angular velocity:

$$\begin{aligned} {}^{MC} \dot{q} &= {}^{BC} \dot{q} * {}^{MB} q = \frac{1}{2} {}^{BC} q * \omega * {}^{MB} q \\ &= \frac{1}{2} {}^{MC} q * {}^{MB} q^{-1} * \omega * {}^{MB} q \\ \Leftrightarrow \omega &= 2 {}^{MB} q * {}^{MC} q^{-1} * {}^{MC} \dot{q} * {}^{MB} q^{-1} \end{aligned}$$

and we differentiate ω to get $\dot{\omega}$:

$$\begin{aligned} \dot{\omega} &= 2 {}^{MB} q * ({}^{MC} q^{-1} * {}^{MC} \ddot{q} - {}^{MC} q^{-1} * {}^{MC} \dot{q} * {}^{MC} q^{-1} * {}^{MC} \dot{q}) * {}^{MB} q^{-1} \\ &= 2 {}^{MB} q^{-1} * {}^{MC} q^{-1} * ({}^{MC} \ddot{q} - {}^{MC} \dot{q} * {}^{MC} q^{-1} * {}^{MC} \dot{q}) * {}^{MB} q \end{aligned}$$

The position ${}^C r$ is now derived twice to obtain the acceleration:

$$\begin{aligned} {}^C \ddot{r} &= {}^C \ddot{p} + {}^{BC} q * (\omega \times (\omega \times {}^B d) + \dot{\omega} \times {}^B d) * {}^{BC} q^{-1} \\ &= {}^C G + {}^{MC} q * {}^{MB} q^{-1} * (a + \omega \times (\omega \times {}^B d) + \dot{\omega} \times {}^B d) * {}^{MB} q * {}^{MC} q^{-1} \\ \Leftrightarrow a &= {}^{MB} q * {}^{MC} q^{-1} * ({}^C \ddot{r} - {}^C G) * {}^{MC} q * {}^{MB} q^{-1} - \omega \times (\omega \times {}^B d) - \dot{\omega} \times {}^B d \end{aligned}$$

We have now expressed ω and a as functions of the measured optical position and orientation and their derivatives and of the transformation $({}^B d, {}^{MB} q)$.

The rotation ${}^{MB} q$ has been determined in the previous Section and does not need to be re-estimated here. In this case, the equation for the specific acceleration is linear in ${}^B d$ and can be written as

$$([\omega]_{\times} [\omega]_{\times} + [\dot{\omega}]_{\times}) {}^B d = -a + {}^{MB} q * {}^{MC} q^{-1} * ({}^C \ddot{r} - {}^C G) * {}^{MC} q * {}^{MB} q^{-1}$$

This equation is written for each frame. We obtain an overdetermined linear system which will be of rank 3 if ω and $\dot{\omega}$ are different from zero and vary from frame to frame. The solution of this system will give the translation vector ${}^B d$.

The necessary IMU and optical measurements are obtained from a calibration routine during which the Sensor Unit is moved in front of the stereo camera rig. The optical measurements give ${}^C r$ and ${}^{MC} q$. The accelerometer measurements are "calibrated" (i.e. biases, gains and misalignments are compensated for) to give a for each frame.

This calibration procedure yielded the following translation vector:

$${}^B d = \begin{bmatrix} 6.0429 \\ -4.6087 \\ 9.5305 \end{bmatrix} mm$$

5.4.4.3 Final Sensor Unit Model

The marker coordinates in Body coordinates after calibration read:

$${}^B m_i = {}^{MB} R^M m_i + {}^B d$$

with

$${}^{MB} R = \begin{bmatrix} 0.9950 & 0.0277 & 0.0959 \\ -0.0292 & 0.9995 & 0.0146 \\ -0.0954 & -0.0173 & 0.9953 \end{bmatrix}$$

and

$${}^B d = \begin{bmatrix} 6.0429 \\ -4.6087 \\ 9.5305 \end{bmatrix} mm$$

which gives

$$[{}^B m_1 {}^B m_2 {}^B m_3 {}^B m_4] = \begin{bmatrix} -31.1281 & 46.9800 & 33.5022 & -118.9256 \\ -68.0422 & -70.3373 & 72.0986 & 76.5428 \\ -44.2353 & -51.7264 & -52.5117 & -39.9018 \end{bmatrix} mm .$$

Chapter 6

Conclusion

Dans cette thèse, nous avons présenté un système de tracking qui utilise des capteurs optiques et inertiels pour déterminer la position et l'orientation d'une Sensor Unit. Le tracking a une grande bande passante d'au moins 200Hz et un petit temps de latence.

Le système permet de suivre un outil à main dans un système de chirurgie assistée par ordinateur. Grâce au tracking, l'outil peut être asservi pour rester dans un plan ou sur une trajectoire.

Le cœur du système est un algorithme de fusion de données qui intègre les données des différents capteurs pour déterminer la position et l'orientation de la Sensor Unit. Il a une grande bande passante car il tourne à la fréquence élevée des capteurs inertiels. Le filtre repose sur une approche directe utilisant des données brutes pour réduire le temps de calcul et il prend en compte les symmétries du système.

L'implémentation en temps réel avec un dispositif expérimental avec des caméras et des capteurs inertiels a permis de montrer que le tracking optique-inertiel peut en effet suivre un objet en mouvement avec une grande bande passante. Nous suggérons plusieurs améliorations pour les versions futures du dispositif expérimental.

In this work, we presented an optical-inertial tracking system. It consists of two stationary cameras and a Sensor Unit with optical markers and an inertial measurement unit. This Sensor Unit is attached to the object being tracked.

The tracking system was developed with an application in a computer-assisted surgery system in mind. The goal is to track a handheld tool and to servo-control it to make it stay in a desired plane or make it move along a desired trajectory. The fact of using the tracking for servo-control introduces two main requirements which are not met by the optical tracking systems currently used for computer-assisted surgery. The system has to have a high bandwidth of at least 200Hz and a low latency, i.e. the delay between a measurement and the tracking output has to be small. Our optical-inertial tracking system meets these requirements. Optical tracking systems used today have a bandwidth

of 60Hz. Faster systems exist but are too expensive for applications in computer-assisted surgery. Even if the cost for faster systems were to be reduced in the future, an optical-inertial tracking system like the one we propose would always be less expensive due to the low cost of the inertial sensors. Also, the use of inertial sensors has the advantages of improving disturbance rejection which is important for servo-control.

The heart of the optical-inertial tracking system is the data fusion algorithm which integrates data from the different sensors, that is optical data from the cameras and inertial data from accelerometers and gyroscopes. It uses a mathematical model of the system dynamics and output. The algorithm runs at the inertial sensor frequency of 200Hz or more which ensures high-bandwidth tracking. In order to reduce latency, we propose a direct approach using sensor data directly in the filter without any previous computations and we also present algorithms taking into account system symmetries and thereby reducing the computational complexity. They are called Right-Invariant and Left-Invariant Extended Kalman Filters. This low-latency approach presents the most important distinction of our system from others presented in the literature.

Calibration of individual sensors and of the optical-inertial setup is important for the system performance. Since the inertial sensors are used to provide high-bandwidth information, they have to be correctly calibrated because the tracking depends solely on the inertial data and the system model between two optical measurements. Determination of the relation between optical markers and the IMU in the Sensor Unit also turned out to be crucial to the system performance. We proposed a novel method to determine the rotation between the optical markers and the IMU using estimations from our data fusion algorithm.

To test the proposed data fusion algorithms, we developed an experimental setup with images sensors from the Wiimote and an IMU. Sensor data was read by a microcontroller at 50Hz from the cameras and at 250Hz from the inertial sensors. The algorithms were implemented in Simulink and executed in real time with an xPC Target application. To evaluate the performance, estimations from the optical-inertial system were compared to resolver measurements from a linear rail and to optical tracking. Experiments with movements executed by the linear rail show that our system can track the movements correctly and with a high bandwidth.

In the future, the experimental setup would have to be improved to make a more accurate evaluation possible. The cameras would have to be replaced by more performing ones which can see more than four points and would thus be able to track more than one object. In the final application, the handheld tool has to be tracked relative to the patient, requiring at least two rigid bodies with optical markers attached to the patient. In the OrthoPilot [Aesculap AG, 2011], for example, three rigid bodies are currently used: one attached to the tibia, one to the femur and one which can be fixed to the foot or to an instrument. The size of the Sensor Unit could also be reduced because it is currently

needlessly large. In order to provide real-time tracking, the data transmission speed should be increased.

Ultimately, the goal is to test the tracking system with a servo-controlled tool in order to evaluate its performance.

Bibliography

- [ADIS16355, 2009] ADIS16355 (2009). ADIS16350/ADIS16355 data sheet, rev. B. Analog Devices (Norwood, MA,USA).
- [advanced realtime tracking GmbH, 2011] advanced realtime tracking GmbH (2011). Weilheim, Germany. <http://www.ar-tracking.de/>.
- [Aesculap AG, 2011] Aesculap AG (2011). Tuttlingen, Germany. <http://www.orthopilot.de/>.
- [Allan, 1966] Allan, D. (1966). Statistics of atomic frequency standards. *Proceedings of the IEEE*, 54(2):221 – 230.
- [Ang et al., 2004] Ang, W. T., Khosla, P. K., and Riviere, C. N. (2004). Kalman filtering for real-time orientation tracking of handheld microsurgical instrument. In *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2574–2580, Sendai, Japan.
- [ATmega2560, 2007] ATmega2560 (2007). Data sheet, Rev. 2549L-08/07. Atmel Corporation, San Jose, CA, USA.
- [Besancon, 2007] Besancon, G. (2007). *Nonlinear Observers and Applications*, volume 363/2007 of *Lecture Notes in Control and Information Sciences*. Springer Berlin/Heidelberg.
- [Bleser and Stricker, 2008] Bleser, G. and Stricker, D. (2008). Advanced tracking through efficient image processing and visual-inertial sensor fusion. In *IEEE Virtual Reality Conference (VR)*, pages 137–144.
- [Bonnabel et al., 2008] Bonnabel, S., Martin, P., and Rouchon, P. (2008). Symmetry-preserving observers. *IEEE Transactions on Automatic Control*, 53(11):2514–2526.
- [Bonnabel et al., 2009] Bonnabel, S., Martin, P., and Salaün, E. (2009). Invariant extended Kalman filter: theory and application to a velocity-aided attitude estimation problem. In *Proceedings of the 48th IEEE Conference on Decision and Control*.

BIBLIOGRAPHY

- [Bouguet, 2010] Bouguet, J.-Y. (2010). Camera calibration toolbox for Matlab. http://www.vision.caltech.edu/bouguetj/calib_doc.
- [Brisson et al., 2004] Brisson, G., Kanade, T., Gioia, A. M. D., and Jaramaz, B. (2004). Precision freehand sculpting of bone. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 3217 of *LNCS*, pages 105 – 112. Springer Berlin / Heidelberg.
- [Cartiaux et al., 2010] Cartiaux, O., Paul, L., Docquier, P. L., Raucent, B., Dombre, E., and Banse, X. (2010). Computer-assisted and robot-assisted technologies to improve bone-cutting accuracy when integrated with a freehand process using an oscillating saw. *Journal of Bone and Joint Surgery - Series A*, 92(11):2076–2082.
- [Chum et al., 2005] Chum, O., Pajdla, T., and Sturm, P. (2005). The geometric error for homographies. *Computer Vision and Image Understanding*, 97(1):86–102.
- [Crassidis et al., 2007] Crassidis, J. L., Markley, F. L., and Cheng, Y. (2007). Survey of nonlinear attitude estimation methods. *Journal of Guidance, Control, and Dynamics*, 30(1):12–28.
- [Davies, 1996] Davies, B. L. (1996). *Computer-Integrated Surgery*, chapter 19, pages 287–296. MIT Press, Boston, Massachusetts.
- [DeMenthon et al., 2001] DeMenthon, D., David, P., and Samet, H. (2001). SoftPOSIT: An algorithm for registration of 3D models to noisy perspective images combining Softassign and POSIT. Technical report car-tr-969, cs-tr-4257, Center for Automation Research, University of Maryland, College Park, MD.
- [DiGioia et al., 2004] DiGioia, A. M., Jaramaz, B., Picard, F., and Nolte, L.-P., editors (2004). *Computer and Robotic Assisted Knee and Hip Surgery*. Oxford University Press.
- [Dorveaux et al., 2009] Dorveaux, E., Vissière, D., Martin, A.-P., and Petit, N. (2009). Iterative calibration method for inertial and magnetic sensors. In *IEEE Conference on Decision and Control and 28th Chinese Control Conference*.
- [Eggert et al., 1997] Eggert, D., Lorusso, A., and Fisher, R. (1997). Estimating 3-D rigid body transformations: a comparison of four major algorithms. *Machine Vision and Applications*, 9:272–290.
- [Farrel and Barth, 1998] Farrel, J. A. and Barth, M. (1998). *The Global Positioning System and Inertial Navigation*. McGraw-Hill Companies, Inc.

- [Fischler and Bolles, 1981] Fischler, M. A. and Bolles, R. C. (1981). Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395.
- [Follmann et al., 2010] Follmann, A., Korff, A., Fürtjes, T., Lauer, W., Kunze, S., Schmieder, K., and Radermacher, K. (2010). Evaluation of a synergistically controlled semiautomatic trepanation system for neurosurgery. In *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 2304 –2307.
- [Fong et al., 2008] Fong, W. T., Ong, S. K., and Nee, A. Y. C. (2008). Methods for in-field user calibration of an inertial measurement unit without external equipment. *Measurement Science and Technology*, 19(8):085202.
- [Gebre-Egziabher, 2004] Gebre-Egziabher, D. (2004). *Design and Performance Analysis of a Low-Cost Aided Dead Reckoning Navigator*. PhD thesis, Stanford University.
- [Gebre-Egziabher et al., 2004] Gebre-Egziabher, D., Hayward, R., and Powell, J. (2004). Design of multi-sensor attitude determination systems. *IEEE Transactions on Aerospace and Electronic Systems*, 40(2):627 – 649.
- [Goodwin et al., 2001] Goodwin, G. C., Graebe, S. F., and Salgado, M. E. (2001). *Control System Design*. Prentice Hall.
- [Grewal et al., 2001] Grewal, M. S., Weill, L. R., and Andrews, A. P. (2001). *Global Positioning Systems, Inertial Navigation, and Integration*. John Wiley & Sons, Inc.
- [Groves, 2008] Groves, P. D. (2008). *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House.
- [Haider et al., 2007] Haider, H., Barrera, O. A., and Garvin, K. L. (2007). Minimally invasive total knee arthroplasty surgery through navigated freehand bone cutting: Winner of the 2005 HAP Paul Award. *Journal of Arthroplasty*, 22(4):535 – 542.
- [Hartley and Zisserman, 2003] Hartley, R. and Zisserman, A. (2003). *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition.
- [Hartley and Sturm, 1997] Hartley, R. I. and Sturm, P. (1997). Triangulation. *Computer Vision and Image Understanding*, 68(2):146 – 157.
- [Hartmann et al., 2010] Hartmann, B., Link, N., and Trommer, G. F. (2010). Indoor 3D position estimation using low-cost inertial sensors and marker-based video-tracking. In *IEEE/ION Position Location and Navigation Symposium (PLANS)*, pages 319 –326.

BIBLIOGRAPHY

- [Hay et al., 2008] Hay, S., Newman, J., and Harle, R. (2008). Optical tracking using commodity hardware. In *7th IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 159–160.
- [Hol et al., 2008] Hol, J., Schon, T., and Gustafsson, F. (2008). A new algorithm for calibrating a combined camera and imu sensor unit. In *10th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pages 1857–1862.
- [Horn, 1987] Horn, B. K. P. (1987). Closed-form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 4(4):629–642.
- [IEEE, 1997] IEEE (1997). Std 95-1997: IEEE standard specification format guide and test procedure for single-axis interferometric fiber optic gyros.
- [Jakopec et al., 2003] Jakopec, M., Rodriguez y Baena, F., Harris, S., Gomes, P., Cobb, J., and Davies, B. (2003). The hands-on orthopaedic robot "Acrobot": Early clinical trials of total knee replacement surgery. *IEEE Transactions on Robotics and Automation*, 19(5):902 – 911.
- [Jolesz et al., 1996] Jolesz, F. A., Kikinis, R., and Shtern, F. (1996). *Computer-Integrated Surgery*, chapter 58: The Vision of Image-Guided Computerized Surgery: The High-Tech Operating Room, pages 717–721. MIT Press, Boston, Massachusetts.
- [Jung and Wohn, 1997] Jung, S. and Wohn, K. (1997). Tracking and motion estimation of the articulated object: a hierarchical Kalman filter approach. *Real-Time Imaging*, 3(6):415 – 432.
- [Jürss and Rudolph, 2008] Jürss, U. and Rudolph, W. (2008). Sur la trace des hot-spots. *elektor*, 11:64–69.
- [Kane et al., 2009] Kane, G., Eggers, G., Boesecke, R., Raczkowski, J., Wörn, H., Marmulla, R., and Mühling, J. (2009). System design of a hand-held mobile robot for craniotomy. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 5761 of *LNCS*, pages 402–409. Springer Berlin / Heidelberg.
- [Kim and Golnaraghi, 2004] Kim, A. and Golnaraghi, M. F. (2004). Initial calibration of an inertial measurement unit using an optical position tracking system. In *IEEE Position Location and Navigation Symposium (PLANS)*, pages 96–101.
- [Knappe et al., 2003] Knappe, P., Gross, I., Pieck, S., Wahrburg, J., Kuenzler, S., and Kerschbaumer, F. (2003). Position control of a surgical robot by a navigation system. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3350–3354.

- [Korff et al., 2010] Korff, A., Elfring, R., Lindhord, S., Follmann, A., Schmieder, K., and Radermacher, K. (2010). Hybrid position control for a synergistic trepanation tool. *International Journal of Computer Assisted Radiology and Surgery*, 5:111–113.
- [Langlotz, 2004] Langlotz, F. (2004). *Computer and Robotic Assisted Knee and Hip Surgery*, chapter 5.1: Localizers and trackers for computer assisted freehand navigation, pages 51–53. Oxford University Press.
- [Lefferts et al., 1982] Lefferts, E. J., Markley, F. L., and Shuster, M. D. (1982). Kalman filtering for spacecraft attitude estimation. *Journal of Guidance, Control, and Dynamics*, 5(5):417–429.
- [Leitner et al., 1997] Leitner, F., Picard, F., Minfelde, R., Schulz, H., Cinquin, P., and Saragaglia, D. (1997). Computer-assisted knee surgical total replacement. In Troccaz, J., Grimson, E., and Mösges, R., editors, *CVRMed-MRCAS’97*, volume 1205 of *Lecture Notes in Computer Science*, pages 629–638. Springer Berlin / Heidelberg.
- [Lobo and Dias, 2003] Lobo, J. and Dias, J. (2003). Vision and inertial sensor cooperation using gravity as a vertical reference. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(12):1597 – 1608.
- [Lobo and Dias, 2007] Lobo, J. and Dias, J. (2007). Relative pose calibration between visual and inertial sensors. *The International Journal of Robotics Research*, 26(6):561–575.
- [Lötters et al., 1998] Lötters, J., Schipper, J., Veltink, P., Olthuis, W., and Bergveld, P. (1998). Procedure for in-use calibration of triaxial accelerometers in medical applications. *Sensors and Actuators A: Physical*, 68(1-3):221 – 228.
- [Maillet et al., 2005] Maillet, P., Nahum, B., Blondel, L., Poignet, P., and Dombre, E. (2005). BRIGIT, a robotized tool guide for orthopedic surgery. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 211 – 216.
- [Mako Surgical Corp., 2011] Mako Surgical Corp. (2011). Fort Lauderdale, Florida. <http://www.makosurgical.com/physicians/knee-products/rio.html>.
- [Martin and Salaün, 2010] Martin, P. and Salaün, E. (2010). Design and implementation of a low-cost observer-based attitude and heading reference system. *Control Engineering Practice*, 18(7):712 – 722. Special Issue on Aerial Robotics.
- [Mathworks, 2011] Mathworks (2011). Natick, Massachusetts, USA. <http://www.mathworks.fr/products/>.

BIBLIOGRAPHY

- [Miller et al., 2009] Miller, F. P., Vandome, A. F., and McBrewster, J. (2009). *Accuracy and precision*. Alphascript Publishing.
- [Mirzaei and Roumeliotis, 2008] Mirzaei, F. and Roumeliotis, S. (2008). A Kalman filter-based algorithm for IMU-camera calibration: Observability analysis and performance evaluation. *IEEE Transactions on Robotics*, 24(5):1143–1156.
- [Mösges and Lavallé, 1996] Mösges, R. and Lavallé, S. (1996). *Computer-Integrated Surgery*, chapter 1: Multimodel Information for Computer-Integrated Surgery, pages 5–19. MIT Press, Boston, Massachusetts.
- [NDI, 2011] NDI (2011). Northern Digital Inc., Waterloo, Canada. <http://www.ndigital.com/products.php#OpticalMeasurementSystems>.
- [Nintendo Co., Ltd., 2011] Nintendo Co., Ltd. (2011). Kyoto, Japan. http://www.nintendo.fr/NOE/fr_FR/wii_54.html.
- [Panahandeh et al., 2010] Panahandeh, G., Skog, I., and Jansson, M. (2010). Calibration of the accelerometer triad of an inertial measurement unit, maximum likelihood estimation and Cramer-Rao bound. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–6.
- [Parnian and Golnaraghi, 2008] Parnian, N. and Golnaraghi, F. (2008). A low-cost hybrid SDINS/multi-camera vision system for a hand-held tool positioning. In *IEEE/ION Position, Location and Navigation Symposium (PLANS)*, pages 489–496.
- [Parnian and Golnaraghi, 2010] Parnian, N. and Golnaraghi, F. (2010). Integration of a multi-camera vision system and strapdown inertial navigation system (SDINS) with a modified Kalman filter. *Sensors*, 10(6):5378–5394.
- [Rehbinder and Ghosh, 2003] Rehbinder, H. and Ghosh, B. (2003). Pose estimation using line-based dynamic vision and inertial sensors. *IEEE Transactions on Automatic Control*, 48(2):186 – 199.
- [Roetenberg, 2006] Roetenberg, D. (2006). *Inertial and Magnetic Sensing of Human Motion*. PhD thesis, Universiteit Twente.
- [Salaün, 2009] Salaün, E. (2009). *Filtering Algorithms and Avionics Systems for Unmanned Aerial Vehicles*. PhD thesis, Mines ParisTech.
- [Schwarz et al., 2009] Schwarz, M. L., Wagner, A., El-Shenawy, A., Gundling, R., Köpfle, A., Handel, H., Badreddin, E., Männer, R., Scharf, H. P., Götz, M., Schill, M., and Pott, P. P. (2009). A handheld robot for orthopedic surgery - ITD. In Dössel, O.,

-
- Schlegel, W. C., and Magjarevic, R., editors, *World Congress on Medical Physics and Biomedical Engineering, September 7 - 12, 2009, Munich, Germany*, volume 25/6 of *IFMBE Proceedings*, pages 99–102. Springer Berlin Heidelberg.
- [Simon, 2006] Simon, D. (2006). *Optimal State Estimation: Kalman, H_∞ , and Nonlinear Approaches*. John Wiley & Sons, Inc., Hoboken, New Jersey.
- [Skog and Händel, 2006] Skog, I. and Händel, P. (2006). Calibration of a MEMS inertial measurement unit. In *IMEKO World Congress*.
- [Stevens and Lewis, 2003] Stevens, B. L. and Lewis, F. L. (2003). *Aircraft control and simulation*. John Wiley & Sons, Inc., Hoboken, New Jersey.
- [Stulberg et al., 2004] Stulberg, S. D., Saragaglia, D., and Miehle, R. (2004). *Computer and Robotic Assisted Knee and Hip Surgery*, chapter 14: Total knee replacement: navigation technique intra-operative model system, pages 157–177. Oxford University Press.
- [Szeliski, 2011] Szeliski, R. (2011). *Computer Vision. Algorithms and Applications*. Springer-Verlag London.
- [Taylor and Stoianovici, 2003] Taylor, R. and Stoianovici, D. (2003). Medical robotics in computer-integrated surgery. *IEEE Transactions on Robotics and Automation*, 19(5):765 – 781.
- [Tobergte et al., 2009] Tobergte, A., Pomarlan, M., and Hirzinger, G. (2009). Robust multi sensor pose estimation for medical applications. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 492–497.
- [Tsai and Lenz, 1989] Tsai, R. and Lenz, R. (1989). A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *IEEE Transactions on Robotics and Automation*, 5(3):345 –358.
- [Vicon, 2011] Vicon (2011). Vicon Motion Systems Limited, Oxford, United Kingdom. <http://www.vicon.com/products/system.html>.
- [Walker et al., 1991] Walker, M. W., Shao, L., and Volz, R. A. (1991). Estimating 3-D location parameters using dual number quaternions. *CVGIP: Image Understanding*, 54(3):358–367.
- [Wonham, 1968] Wonham, W. M. (1968). On a matrix Riccati equation of stochastic control. *SIAM Journal on Control*, 6(4):681–397.

- [You et al., 1999] You, S., Neumann, U., and Azuma, R. (1999). Hybrid inertial and vision tracking for augmented reality registration. In *Proceedings of IEEE Virtual Reality*, pages 260–267.
- [Zachariah and Jansson, 2010] Zachariah, D. and Jansson, M. (2010). Joint calibration of an inertial measurement unit and coordinate transformation parameters using a monocular camera. In *International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–7.
- [Zhang, 2000] Zhang, Z. (2000). A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334.

Capteur de mouvement intelligent pour la chirurgie prothétique naviguée

Résumé: Nous présentons un système de tracking optique-inertiel qui consiste en deux caméras stationnaires et une Sensor Unit avec des marqueurs optiques et une centrale inertielle. La Sensor Unit est fixée sur l'objet suivi et sa position et son orientation sont déterminées par un algorithme de fusion de données. Le système de tracking est destiné à asservir un outil à main dans un système de chirurgie naviguée ou assistée par ordinateur. L'algorithme de fusion de données intègre les données des différents capteurs, c'est-à-dire les données optiques des caméras et les données inertielles des accéléromètres et gyroscopes. Nous présentons différents algorithmes qui rendent possible un tracking à grande bande passante avec au moins 200Hz avec des temps de latence bas grâce à une approche directe et des filtres dits invariants qui prennent en compte les symétries du système. Grâce à ces propriétés, le système de tracking satisfait les conditions pour l'application désirée. Le système a été implémenté et testé avec succès avec un dispositif expérimental.

Mots clés: fusion de données optiques-inertielles, filtrage de Kalman, observateurs non-linéaires, chirurgie assistée par ordinateur, outil à main asservi

Smart motion sensor for navigated prosthetic surgery

Abstract: We present an optical-inertial tracking system which consists of two stationary cameras and a Sensor Unit with optical markers and an inertial measurement unit (IMU). This Sensor Unit is attached to the object being tracked and its position and orientation are determined by a data fusion algorithm. The tracking system is to be used for servo-controlling a handheld tool in a navigated or computer-assisted surgery system. The data fusion algorithm integrates data from the different sensors, that is optical data from the cameras and inertial data from accelerometers and gyroscopes. We present different algorithms which ensure high-bandwidth tracking with at least 200Hz with low latencies by using a direct approach and so-called invariant filters which take into account system symmetries. Through these features, the tracking system meets the requirements for being used in the desired application. The system was successfully implemented and tested with an experimental setup.

Keywords: optical-inertial data fusion, Kalman filtering, nonlinear observers, computer-assisted surgery, servo-controlled handheld tool

