Estimating the orientation of a game controller from inertial and magnetic measurements

Peng He

Doctorat en génie mécanique

Québec, Canada

Résumé

L'estimation de l'orientation d'un corps rigide en mouvement dans l'espace joue un rôle indispensable dans les technologies de navigation, par exemple, les systèmes militaires de missiles, les avions civils, les systèmes de navigation chirurgicale, la cartographie faite par des robots, les véhicules autonomes et les contrôleurs de jeux. Cette technique est maintenant utilisée dans certaines applications qui nous touchent directement, notamment dans les contrôleurs de jeux tels que la Wii-mote. Dans cette veine, la recherche présentée ici porte sur l'estimation de l'orientation d'un corps rigide à partir des mesures de capteurs inertiels et magnétiques peu coûteux.

Comme les capteurs inertiels permettent de mesurer les dérivées temporelles de l'orientation, il est naturel de commencer par l'estimation de la vitesse angulaire. Par conséquent, nous présentons d'abord une nouvelle façon de déterminer la vitesse angulaire d'un corps rigide à partir d'accéléromètres. Ensuite, afin d'estimer l'orientation, nous proposons une nouvelle méthode d'estimation de l'orientation d'un corps rigide dans le plan vertical à partir des mesures d'accéléromètres, en discernant ses composantes inertielle et gravitationnelle. Mais, ce n'est sûrement pas suffisant d'estimer l'orientation dans le plan vertical, parce que la plupart des applications se produisent dans l'espace tridimensionnel. Pour estimer les rotations dans l'espace, nous présentons d'abord la conception d'un contrôleur de jeu, dans lequel tous les capteurs nécessaires sont installés. Ensuite, ces capteurs sont étalonnés pour déterminer leurs facteurs d'échelle et leurs zéros, de manière à améliorer leurs exactitudes. Ensuite, nous développons une nouvelle méthode d'estimation de l'orientation d'un corps rigide se déplaçant dans l'espace, encore en discernant les composantes gravitationnelle et inertielle des accélérations.

Finalement, pour imiter le contrôleur de jeu Wii, nous créons une interface usager simple de sorte qu'une représentation virtuelle du contrôleur de jeu puisse suivre chaque mouvement du contrôleur de jeu conçu (réalité virtuelle). L'interface usager conçue montre que l'algorithme proposé est suffisamment précis pour donner à l'usager un contrôle fidèle de l'orientation du contrôleur de jeu virtuel.

Abstract

Estimating the orientation of a rigid-body moving in space is an indispensable component of navigation technology, e.g., military missile systems, civil aircrafts, surgical navigation systems, robot mapping, autonomous vehicles and game controllers. It has now come directly into some aspects of our lives, notoriously in game controllers, such as the Wiimote. In this vein, this research focuses on the development of new algorithms to estimate the rigid-body orientation from common inexpensive inertial and magnetic sensors.

As inertial sensors measure the time derivatives of the orientation, it is natural to start with the estimation of the angular velocity. More precisely, we present a novel way of determining the angular velocity of a rigid body from accelerometer measurements. This method finds application in crashworthiness and motion analysis in sports, for example, where impacts forbid the use of mechanical gyroscopes. Secondly, in an attempt to estimate the orientation in a simplified setting, we propose a novel method of estimating the orientation of a rigid body in the vertical plane from point-acceleration measurements, by discerning its gravitational and inertial components. Thirdly, it is surely not enough to estimate the orientation in the vertical plane, because most applications take place in three dimensions. For estimating rotations in space, we first present the game controller design, in which all necessary sensors are installed. Then, these sensors are calibrated to determine their scale factors and offsets so as to improve their performances. Thence, we develop a novel method of estimating the orientation of a rigid body moving in space from inertial sensors, also by discerning the gravitational and inertial components of the acceleration.

Finally, in order to imitate the game controller Wii, we create a simple user interface in which a virtual representative of the game controller follows every orientation of the true game controller (virtual reality). The user interface shows that the proposed algorithm is sufficiently accurate to give the user a transparent control of the orientation of the virtual game controller.

Contents

List of Tables

List of Figures

List of Multimedia Extensions

- The video *manette_ GUI.avi* presents a simple virtual environment to demonstrate usefulness of the proposed algorithm.

List of Symbols

Symbol	Units	Description
$\widehat{a}$	$\mathrm{m/s^2}$	The accelerometer measurement
a	$\mathrm{m/s^2}$	The true acceleration component along the accelerometer sensitive direction
$\mathbf{b}$	m	The position vector of point B with respect to point O
$\mathcal{B}$		The moving frame
$\ddot{\mathbf{c}}$	$\mathrm{m/s^2}$	The combined inertial and gravitational acceleration
$\mathbf{e}$		The unit vector representing the sensitive direction of the accelerometer
$\mathcal{F}$		The fixed frame
g	$\mathrm{m/s^2}$	The gravitational acceleration
$K_{d,b}, K_{d,v}$	$\mathrm{s^{-1}}$	The tuning parameters for position in the interface
$\widehat{\mathbf{m}}$	T	The magnetometer measurement
$\mathbf{m}$	T	The true magnetic North
$\mathcal{M}$		The magnetometer frame
$\mathcal{O}$		The frame defined by the Optitrack
$\mathbf{p}$	m	The position vector of the accelerometer with respect to fixed point O
$\mathbf{q}_M$		The quaternion representing rotations
$\mathbf{Q}$		The rotation matrix from the fixed frame $\mathcal{F}$ to the moving frame $\mathcal{B}$
$\mathbf{r}$	m	The position vector of the accelerometer with respect to reference point B
$\mathbf{R}_{\mathcal{BO}}$		The rotation matrix from frame $\mathcal{O}$ to frame $\mathcal{B}$
v	V	The voltage output of the accelerometer
$\mathbf{v}$	V	The vector of sensor output voltage

$\mathbf{w}$		The weighting factor for estimating the angular velocity
$\boldsymbol{\alpha}$	$\mathrm{rad/s^2}$	The vector of the angular acceleration
α	$\mathrm{rad/s^2}$	One component of the angular acceleration
$\boldsymbol{\beta}$	$\mathrm{m/s^3}$	The time rate of the inertial acceleration
γ	$\mathrm{rad/s^3}$	The time rate of the angular acceleration α, a.k.a. angular jerk
δa	$\mathrm{m/s^2}$	The accelerometer measurement noise
$\delta \mathbf{a}$	$\mathrm{m/s^2}$	The vector of accelerometer measurement noise
Δt	s	The sampling time
$\delta\alpha$	$\mathrm{rad/s^2}$	The angular acceleration error
$\delta\omega$	rad/s	The angular velocity error
η	$\mathrm{m/s^2}$	The offset of the accelerometer
θ	deg	The pitch angle
μ	$\frac{\mathrm{m/s^2}}{\mathrm{V}}$	The scale factor of the accelerometer
ρ		The weighting factor in the MSE method
σ_a^2	$\left(\mathrm{m/s^2}\right)^2$	The measurement noise variance of a single accelerometer
σ_b^2	$\mathrm{m^2}$	The variance of the position noise
σ_m^2	$\mathrm{T^2}$	The variance of the magnetometer measurement noise
σ_β^2	$\left(\mathrm{m/s^3}\right)^2$	The variance of one component of $\boldsymbol{\beta}$
σ_γ^2	$\left(\mathrm{rad/s^3}\right)^2$	The variance of γ
σ_ϕ^2	$\mathrm{deg^2}$	The variance of the angle ϕ
$\Sigma_{\boldsymbol{\omega}}$	$\mathrm{(rad/s)^2}$	The covariance matrix of $\widehat{\boldsymbol{\omega}}$
ϕ	deg	The roll angle
$\widetilde{\phi}$	deg	The estimated orientation in the vertical plane from the time-integration method
$\overline{\phi}$	deg	The estimated orientation in the vertical plane from the tilt-sensor method
$\widehat{\phi}$	deg	The estimated orientation in the vertical plane from the MSE method
ϕ_X, ϕ_Y, ϕ_Z	deg	The rotation angles
ψ	deg	The azimuth angle
ω	rad/s	One component of the angular velocity
$\boldsymbol{\omega}$	rad/s	The vector of the angular velocity

$\widehat{\boldsymbol{\omega}}$	rad/s	The estimated angular velocity or gyroscope measurement

Pas à pas, on va loin.

France et Chine

Acknowledgements

Life should be colorful, and four years and a half of living with the research and the thesis is a part of my colorful life. Some people say that the Ph.D. life is boring and some others disagree with this. In my opinion, whether it is boring or interesting, Ph.D gives us a way of thinking, creating and proposing our own methods. This is of much importance in our future life.

During these four years, I would say that the most interesting and fortunate thing is working with many gracious, amiable, creative and lovely people. Foremost, I would like to show great gratitude to my supervisor Prof. Philippe Cardou. His patience, diligence and intelligence left me a deep impression. Everytime when I got in difficulty in research, his opening mind and creative thinking always made me out of it, which is like the light at the end of the tunnel. In addition, his comprehensive expertise in robotics and navigation systems in such a young age always inspired me to go forward.

Meanwhile, I want to thank my co-supervisor Prof. André Desbiens for his tremendous help in my research. His technical and experimental support helped me save much time, especially the laboratory and the Optitrack system. His outstanding knowledge of electrical engineering and control systems was also of much help in the development of new algorithms. In addition, special thanks are also due to Prof. Eric Gagnon who is from Defence Research and Development Canada (DRDC). In the supervisor group, his constructive suggestions always pushed my research into further and deeper steps.

Moreover, I am also grateful to members of my thesis defence committee, Prof. Yves St-Amant and Prof. Lionel Birglen. Thank you for your valuable time to read my thesis and give your significant comments and suggestions. This will help improve much for my thesis.

The support of all the members in the Laboratoire de robotique is dutifully acknowledged. Special thanks are given to Boris Mayer-St-Onge, Simon Foucault, Thierry Laliberté and

Marc-Antoine Lacasse. Thanks for their great help with the installation of software and their technical support. Particularly, Michel Dominique from the Department of Mechanical Engineering and Guillaume Chauvet from École Centrale de Nantes are also acknowledged, as their significant help on design of PCB and game controller allowed to validate my sensor calibration method and the final human-machine interface. Thank you very much.

Last but not least, I sincerely want to thank my parents who are far away in China. Regardless of the distance, their encouragement and support forever make me positive, energetic and confident.

Foreword

This thesis consists of six chapters, which are the introduction, four chapters with algorithm development and sensor calibration, and the conclusion. Among them, the first five chapters are almost entirely drawn from scientific journal or conference articles already published by the author. In these chapters, some of the contents have been reorganized for the sake of rationality and logic. In particular, the literature reviews of the articles are collected into one chapter at the beginning of the thesis so that one can get a better overview of the field before going through each particular method. This also allows to avoid some repetitions and facilitates the explanation of the interconnections between each topic. In addition to those in the published papers, some more texts and figures, e.g., new simulation and experimental results, have been added in the thesis. Here, we detail the contributions of the author of this thesis to the published articles, and compare them to those of his coauthors.

The first journal paper (P. He and P. Cardou. Estimating the angular velocity from body-fixed accelerometers. J. Dyn. Sys., Meas., Control, 134(6): 061015-1-061015-10, Nov., 2012.), shown in Fig. 1, focuses on the estimation of the angular velocity of a rigid body using an accelerometer-array. This accelerometer-array was designed by two former members of the robotics laboratory in Université Laval, Guillaume Fournier and Philippe Gagnon. The general idea behind the method of the mean squared error using a weighting factor was suggested by Professor Philippe Cardou. The mathematics and computer code necessary to make the method work were developed by the author of this thesis. The same performed all the associated experiments, analysed their results and wrote the associated journal article. The paper was reviewed by Professor Philippe Cardou before the initial submission.

The conference paper (P. He, P. Cardou, and A. Desbiens. Estimating the orientation of a game controller moving in the vertical plane using inertial sensors. In ASME International Design Engineering Technical Conferences, Chicago, Illinois, USA, 2012.), shown in Fig. 2, is devoted to the estimation of the orientation of a rigid body moving in the vertical plane. The initial idea was proposed by the author of this thesis, while the use of the Kalman filter was suggested by Professor André Desbiens. The trajectory models of random walk and ∞ were

Estimating the Angular Velocity From Body-Fixed Accelerometers

Peng He
e-mail: peng.he.1@ulaval.ca

Philippe Cardou
Mem. ASME
e-mail: pcardou@gmc.ulaval.ca

Robotics Laboratory,
Department of Mechanical Engineering,
Laval University,
Quebec City, QC, G1V 0A6, Canada

This paper presents a novel way of determining the angular velocity of a rigid body from accelerometer measurements. This method finds application in crashworthiness and motion analysis in sports, for example, where impacts forbid the use of mechanical gyroscopes. Based on previous work, the time-integration (TI) and polynomial-roots (PR) estimates of the angular velocity are first computed. The TI and PR estimates are then linearly combined through a weighted sum whose weighting factor is chosen so as to minimize the 'variance of the resulting estimate. The proposed method is illustrated in an experiment, where the twelve accelerometer array (OCTA) is moved manually. A comparison of the angular-velocity estimates obtained from the proposed method and those obtained from a magnetic displacement sensor shows that the resulting estimates are robust and do not suffer from the drift problems that hinder the TI method. Moreover, comparison with a previously reported method indicates that the method proposed here is less sensitive to measurement errors, especially at low angular velocities. [DOI: 10.1115/1.4006364]

1 Introduction

In biomechanics, anthropomorphic test devices are typically instrumented with arrays of accelerometers in order to estimate the trajectory of the head after an impact [1]. This was proposed by Mertz in 1967 who showed that an accelerometer array could be used to measure not only point accelerations but also rigid-body rotations [2]. Many biomechanics researchers went further in this direction, e.g., see Refs. [3–9].

Accelerometer arrays were also used in other biomechanics applications, such as gait analysis [10–12]. Morris (1973) and Hayes et al. (1983) used accelerometers to track the motions of the human shank. Impact accelerations measured by accelerometer arrays are widely used, like in the assessment of protective equipment, etc. Other biomechanics applications include the assessment of appropriate head protection with helmets for various human activities (e.g., aircrew helmets, American-football helmets, bicycling helmets) have also been studied [13].

Accelerometer arrays were also proposed in other fields, such as robotics [14–16], vehicle guidance and control [17–22], and inertial navigation systems [23]. Optical motion capture, such as the Vicon [24] or even high-speed cameras, are also commonly used in biomechanics to measure the limb displacements. Optical and inertial systems, such as accelerometer arrays, may be seen as complementary. Indeed, optical motion capture generally offers accurate pose estimates, but suffers from occlusion, lighting, and portability problems. On the other hand, inertial systems do not rely on any external reference, and therefore provide reliable displacement–rate estimates, while the displacement estimates they provide are plagued by drift. Moreover, analyzing the motion that results from impacts requires high sampling rates, which generally impact more the price of optical systems than that of inertial ones. Indeed, optical systems generally involve more data acquisition and processing in a given iteration, which makes them more sensitive to a step-length reduction.

There are currently two main methods for computing the angular velocity from accelerometer-array measurements. The first one uses the time-integration of the angular acceleration: we call it the TI method. Use of this method is reported in the works of Chen et al. [25], Mostov et al. [19], and Parsa et al. [26]. In the second method, the angular velocity is obtained from the quadratic forms of the angular-velocity components: we call it the PR method. Schuler [27], Grammatikos [2], Parsa [28], Peng and Golnaraghi [20] have all reported on this method.

However, the TI method is plagued by an important disadvantage. As mentioned above, the rigid-body angular velocity has to be calculated through a time-integration of the angular acceleration, a process that is, by nature, unstable—it has a pole at the origin of the complex plane. This results in an error build-up, or drift, in the estimates, a phenomenon that was noted by Mital and King [5], among others. This behavior can also be observed from the well known 3-2-2-2 accelerometer-array geometry [3], which is always coupled with the TI method. The inevitable consequence of this drift problem is that the angular-velocity estimates become inaccurate after less than a second, limiting the applicability of the TI method to short experiments.

The PR method also comes with its share of problems. The quadratic forms of the angular-velocity components, which are used in this method, are invariant to a sign change of the angular-velocity vector. When the values of these quadratic forms are known, this leaves ambiguity in the solution for the angular velocity, which is generally overcome by a comparison with the TI estimate. Moreover, the PR estimates are very sensitive to errors at low angular velocities, as shown in Ref. [1]. This lack of robustness at low angular velocities seems to restrict the use of the PR method to applications with high angular rates. An example of such an application is the guidance of autonomous naval support rounds [29,30]. However, for most applications, its high sensitivity to errors at low angular velocities makes the PR method unreliable.

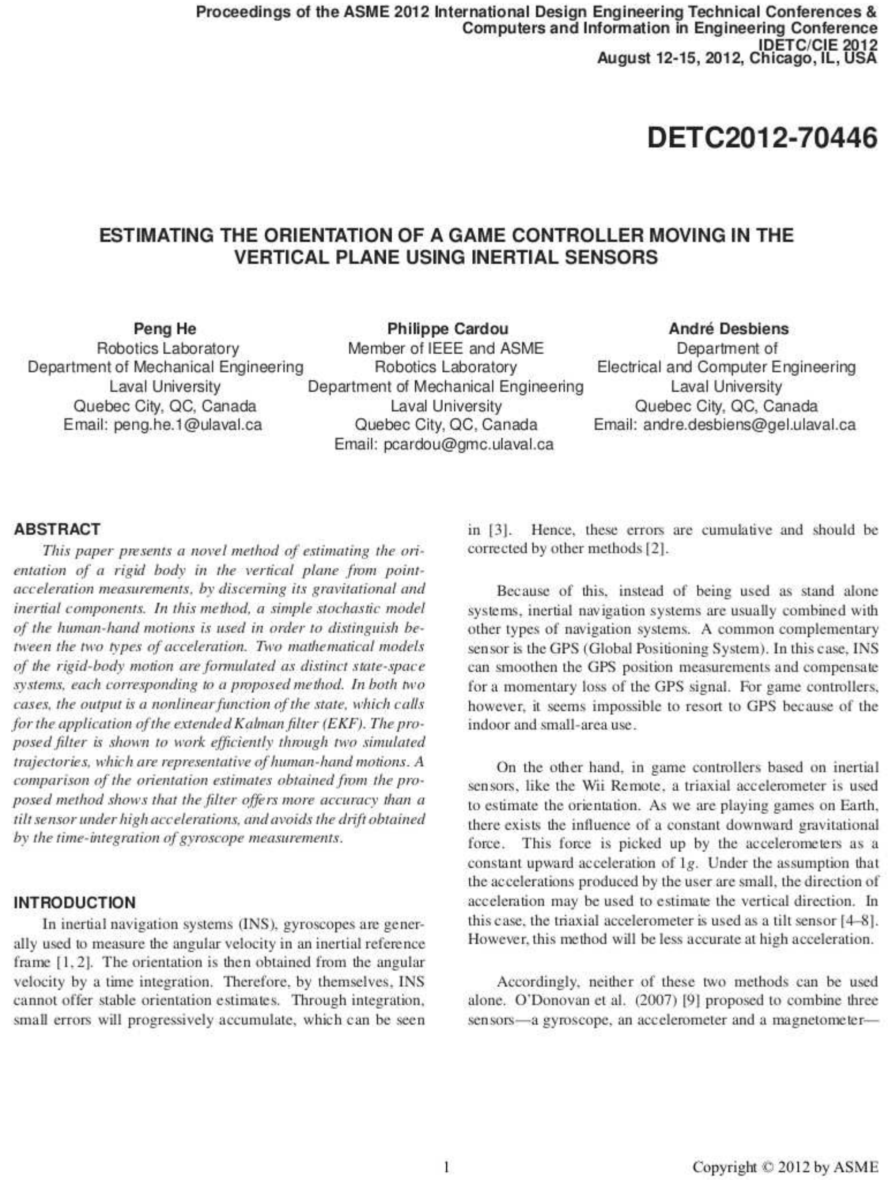
From this portrait, it is apparent that the drawbacks of the PR and TI methods are different, which suggests that their combination would be beneficial. Parsa et al. [31] proposed such a combination, which is described in the Appendix. However, in their method, the relative accuracies of the TI and PR estimates are not taken into account. Cardou and Angeles [32] also proposed a combination based on extended Kalman filtering, which minimizes the variance of the errors. However, this method was only devised for planar motions, and, to our knowledge, it has not yet been extended to the spatial case. Perhaps more importantly, this method was found to be difficult to use in practice. Indeed, it models the angular jerk of the accelerometer array as white noise and requires prior knowledge of the variance of this "noise." In other words, with this

Contributed by the Dynamic Systems Division of ASME for publication in the Journal of Dynamic Systems, Measurement, and Control. Manuscript received October 25, 2011; final manuscript received February 24, 2012; published online October 8, 2012. Assoc. Editor: YangQuan Chen.

Figure 1: J. Dyn. Sys., Meas., Control, 2012

proposed after the discussion with Professors Philippe Cardou and André Desbiens. All the simulations were implemented in Matlab and Simulink by the author of this thesis, in addition to the following analysis and conclusions. The paper was reviewed by Professors as well before its submission.

The second journal paper (P. He, P. Cardou, A. Desbiens, and E. Gagnon. Estimating the



Proceedings of the ASME 2012 International Design Engineering Technical Conferences &
Computers and Information in Engineering Conference
IDETC/CIE 2012
August 12-15, 2012, Chicago, IL, USA

DETC2012-70446

ESTIMATING THE ORIENTATION OF A GAME CONTROLLER MOVING IN THE VERTICAL PLANE USING INERTIAL SENSORS

Peng He
Robotics Laboratory
Department of Mechanical Engineering
Laval University
Quebec City, QC, Canada
Email: peng.he.1@ulaval.ca

Philippe Cardou
Member of IEEE and ASME
Robotics Laboratory
Department of Mechanical Engineering
Laval University
Quebec City, QC, Canada
Email: pcardou@gmc.ulaval.ca

André Desbiens
Department of
Electrical and Computer Engineering
Laval University
Quebec City, QC, Canada
Email: andre.desbiens@gel.ulaval.ca

ABSTRACT

This paper presents a novel method of estimating the orientation of a rigid body in the vertical plane from point-acceleration measurements, by discerning its gravitational and inertial components. In this method, a simple stochastic model of the human-hand motions is used in order to distinguish between the two types of acceleration. Two mathematical models of the rigid-body motion are formulated as distinct state-space systems, each corresponding to a proposed method. In both two cases, the output is a nonlinear function of the state, which calls for the application of the extended Kalman filter (EKF). The proposed filter is shown to work efficiently through two simulated trajectories, which are representative of human-hand motions. A comparison of the orientation estimates obtained from the proposed method shows that the filter offers more accuracy than a tilt sensor under high accelerations, and avoids the drift obtained by the time-integration of gyroscope measurements.

INTRODUCTION

In inertial navigation systems (INS), gyroscopes are generally used to measure the angular velocity in an inertial reference frame [1, 2]. The orientation is then obtained from the angular velocity by a time integration. Therefore, by themselves, INS cannot offer stable orientation estimates. Through integration, small errors will progressively accumulate, which can be seen in [3]. Hence, these errors are cumulative and should be corrected by other methods [2].

Because of this, instead of being used as stand alone systems, inertial navigation systems are usually combined with other types of navigation systems. A common complementary sensor is the GPS (Global Positioning System). In this case, INS can smoothen the GPS position measurements and compensate for a momentary loss of the GPS signal. For game controllers, however, it seems impossible to resort to GPS because of the indoor and small-area use.

On the other hand, in game controllers based on inertial sensors, like the Wii Remote, a triaxial accelerometer is used to estimate the orientation. As we are playing games on Earth, there exists the influence of a constant downward gravitational force. This force is picked up by the accelerometers as a constant upward acceleration of $1g$. Under the assumption that the accelerations produced by the user are small, the direction of acceleration may be used to estimate the vertical direction. In this case, the triaxial accelerometer is used as a tilt sensor [4–8]. However, this method will be less accurate at high acceleration.

Accordingly, neither of these two methods can be used alone. O'Donovan et al. (2007) [9] proposed to combine three sensors—a gyroscope, an accelerometer and a magnetometer—

1

Copyright © 2012 by ASME

Figure 2: ASME International Design Engineering Technical Conferences, 2012

orientation of a rigid body moving in space using inertial sensors. Multibody System Dynamics, 2014. DOI 10.1007/s11044-014-9425-8.), shown in Fig. 3, proposed a novel method for estimating the orientation of a game controller moving in space, using inertial and magnetic sensors. Similarly, the initial idea was proposed by the author of this thesis, with the help of Professor Philippe Cardou. The game controller used in the paper was designed by an intern Guillaume Chauvet and a technician Michel Dominique from the Department of Mechanical Engineering. The target-pointing trajectory was suggested by Professor Eric Gagnon. All the

Multibody Syst Dyn
DOI 10.1007/s11044-014-9425-8

Estimating the orientation of a rigid body moving in space using inertial sensors

Peng He · Philippe Cardou · André Desbiens · Eric Gagnon

Received: 12 December 2013 / Accepted: 30 May 2014

Abstract This paper presents a novel method of estimating the orientation of a rigid body moving in space from inertial sensors, by discerning the gravitational and inertial components of the accelerations. In this method, both a rigid-body kinematics model and a stochastic model of the human-hand motion are formulated and combined in a nonlinear state-space system. The state equation represents the rigid body kinematics and stochastic model, and the output equation represents the inertial sensor measurements. It is necessary to mention that, since the output equation is a nonlinear function of the state, the extended Kalman filter (EKF) is applied. The absolute value of the error from the proposed method is shown to be less than 5 deg in simulation and in experiments. It is apparently stable, unlike the time-integration of gyroscope measurements, which is subjected to drift, and remains accurate under large accelerations, unlike the tilt-sensor method.

Keywords Accelerometer · Magnetometer · Orientation · Game controller · Data fusion

1 Introduction

In inertial navigation systems (INS), the attitude of the moving vehicle is usually obtained by time integrating the angular velocity, which is measured directly by gyroscopes in an

P. He (✉) · P. Cardou
Robotics Laboratory, Department of Mechanical Engineering, Université Laval, Quebec City, QC, Canada
e-mail: peng.he.1@ulaval.ca

P. Cardou
e-mail: pcardou@gmc.ulaval.ca

A. Desbiens
Department of Electrical and Computer Engineering, Université Laval, Quebec City, QC, Canada
e-mail: andre.desbiens@gel.ulaval.ca

E. Gagnon
RDDC Valcartier, Quebec City, QC, Canada
e-mail: Eric.Gagnon@drdc-rddc.gc.ca

Published online: 17 July 2014

Springer

Figure 3: Multibody System Dynamics, 2014

simulations were performed by the author of this thesis. In addition, all the experiments were conducted in the lab of Optitrack provided by Professor André Desbiens. Before the submission, Professors Philippe Cardou, André Desbiens and Eric Gagnon reviewed the manuscript prepared by the author of this thesis and gave many important suggestions and comments.

Chapter 1

Introduction

Where there is a will, there is a way.

In this chapter, we first give a general introduction of game-controllers and of the inertial sensors they may contain, such as accelerometers and gyroscopes, which may be used. From that, the objective of our project is put forward and the milestones are listed. Finally, regarding to each milestone, the corresponding detailed literature review is presented.

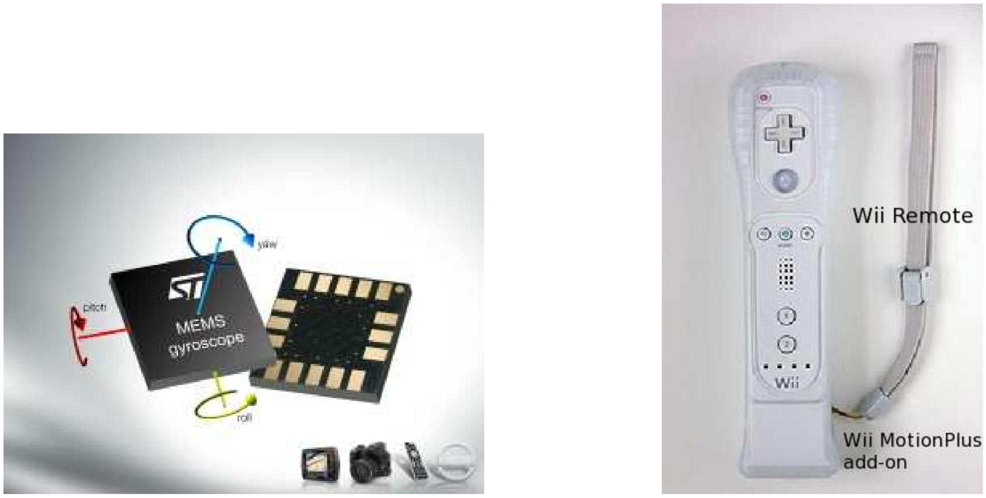
General introduction

A good human-machine interface is one that allows for quick intuitive information exchanges between a machine and its user. Such devices have a variety of applications, from cell phones, to control panels for industrial robots, and from video games to process control in nuclear power plants, that is to say, from daily life to advanced science. In principle, a human-machine interface should be accessible, logical, easy and pleasant to use. Therefore, human-machine interface bears a great importance in the modern world.

Game controllers are used to govern the playable body or control the events in computer or video games. A wide variety of such devices are available, such as light guns, gaming keypads, paddles and so on. The Wii remote [1], is perhaps the most revolutionary of game controllers, as it can track the trajectory of a human limb in the screen. It has become the daily entertainment in many families, thus playing a very important role.

To this day, human-machine interface based on inertial measurements utilize classical IMUs (Inertial Measurement Units), which consist of a triaxial accelerometer and a triaxial gyroscope, shown in Fig. 1.1(a). These latter sensors offer a good accuracy, but also entail some drawbacks, such as a relatively high cost compared with that of accelerometers, and poor robustness under high acceleration. For example, the Wii Remote relies on a triaxial accelerometer alone to estimate its motion. However, in order to obtain more accurate estimates and to take the sensing gameplay movements to a higher level of interaction, Nintendo sells an add-on called the Wii MotionPlus. This device is nothing but a gyroscope, which effectively turns the Wii Remote into a classical IMU.

To explain this, notice that an accelerometer alone can only measure point acceleration without rotation about the X, Y, and Z axes. When the accelerations to be measured are known to be small, one can use the triaxial accelerometer to measure the direction of the gravitational acceleration. In this case, the accelerometer is used as a tilt sensor, but is limited to accelerations that are below $1g$. Gyroscopes measuring rotation directly as mentioned above are capable of compensating this shortcoming. For example, the particular chip—IDG-600 gyroscope is used as the Wii MotionPlus, shown in Fig. 1.1(b), owning two sensor elements with novel vibrating dual-mass bulk silicon configurations which sense the rotation rate about X and Y axes. It can measure up to 1500 degrees per second and provide full-range motion, which is accurate enough for Wii Remote.



(a) A MEMS gyroscope (b) Wii-Remote controller with Wii MotionPlus

Figure 1.1: Gyroscope and accelerometer-array

However, there are some problems with gyroscopes and classical IMUs which currently limit the development of game controllers. First, gyroscopes are approximately 10 times as expensive as accelerometers, and also comsume about 50 times more power, for the same number of sensitive axes. Because game controllers are a kind of mass product, maintaining a low price is of first importance. Furthermore, gyroscopes do not generally perform well when subjected to high accelerations. Hence, there is room for improvement in applications where the game controller undergoes high accelerations or even impacts.

In order to solve these problems, the proposed solution is to use accelerometer arrays instead of gyroscopes. Accelerometer arrays are a class of inertial measurement units, compared to the classical IMU, which allows the estimation of the acceleration field of an object, i.e., accelerations of some or all of its points. It consists of an array of accelerometers that are located at a constellation of points of a rigid body. From these accelerometer measurements, one can estimate the acceleration field of the rigid body, and, in turn, the complete trajectory. To this day, accelerometer arrays have been preferred to gyroscope-based inertial measurement units in certain niche applications such as crashworthiness [2, 3, 4, 5, 6, 7, 8], projectile guidance [9, 10] and gait analysis [11, 12, 13], for instance.

An example of an accelerometer array is shown in Fig. 1.2. Such a device is capable of providing stable estimates of the angular velocity. Because they consist only of accelerometers, they are not plagued by the high cost and poor robustness of mechanical gyroscopes under high accelerations.

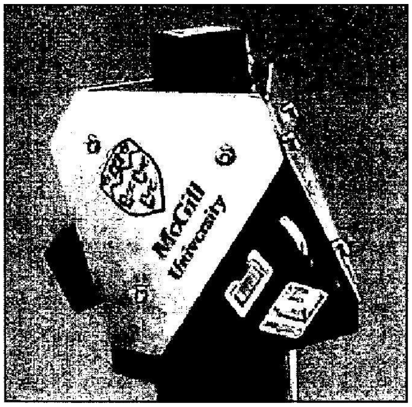


Figure 1.2: An accelerometer array

Nevertheless, to be fair, accelerometer arrays also bear some disadvantages. One is that the performance of an accelerometer array is mainly determined by its geometry. Hence an optimal geometry from the perspective of sensing is not always practical in a given application. Several accelerometer-array geometries have been proposed over the years: Grammatikos, 1965 [14]; Padgaonkar et al., 1975 [15]; Huijser, 1992 [16]; Chen et al., 1994 [17]; Genin et al., 1997 [18]; Parsa et al., 2003 [19]; Parsa et al., 2005 [20]. For instance, the accelerometer array in Fig. 1.2 is called Plato, which was devised and prototyped at McGill University's Robotic Mechanical Systems Laboratory [19]. This accelerometer array is composed of four triaxial accelerometers located at the vertices of a tetrahedron. In this project, we will investigate this problem of geometry and finally design a stable, compact, accurate and high-speed response game controller based on an array of accelerometers.

Objective and milestones of the project

As mentioned above, the objective is to estimate more accurately the orientation of a game controller by relying only on low-cost sensors that do not require external references other than those that are naturally in place, i.e., the gravitational and magnetic fields of the Earth. To this end, we will devise and implement an accelerometer-based game controller that allows for transparent interaction between user and machine. We plan to fulfill this objective through the following milestones:

a) We first estimate the angular velocity in the plane, i.e., in 2D. In this case, we will have four unknowns, i.e., angular acceleration, inertial acceleration and squared angular velocity, to solve for. Therefore, for the plane, the accelerometer array only needs four uniaxial accelerometers. Thus, we can use two biaxial accelerometers for measuring planar displacements. For 3D displacements, however, when we install two triaxial accelerometers at each end of a bar, we cannot estimate the angular velocity about its longitudinal axis. If this quantity is needed, a more elaborate accelerometer array geometry should be designed, e.g., a cross shape or a T-shape.

b) Angular velocity, however, is not the final goal: it is the orientation. Hence, the next step is to estimate the game-controller orientation from the angular velocity. However, it is known that time-integrating directly the angular velocity to compute the orientation is an unstable process. In order to solve this problem, we combine the angular velocity estimates with gravitational orientation estimates, i.e., we use the array in inertial mode and in a tilt sensor mode at the same time. Furthermore, we use a prototype to demonstrate this estimation method.

c) As a matter of fact, game controllers are played in the 3D world. Therefore, we need to generalize the research in 2D to 3D. It is more complicated to solve the 3D problem, the most challenging diffculty being that the azimuth angle cannot be determined from gravitational measurements. Accordingly, we are forced to propose a method or find a simple sensor to estimate this angle. In this thesis, a magnetometer is used for this purpose.

d) As for the 2D case, demonstration by prototyping the game controller is indispensable in 3D. Given the final objective, we will integrate it to a simple virtual environment to allow for a qualitative evaluation of its capabilities. The most important problem here may be the design of this prototype. For example, we could make it similar to the Wii remote bar, compact, but with a different geometry.

Such a device finds application in the video game industry, but also in robot teleoperation and biomechanics, for instance. In this thesis, milestone a) is presented in chapter 1, while the milestone b) is introduced in detail in chapter 2. Milestones c) and d) are reported in chapter 4. Prior to that, the calibration of sensors has been done in chapter 3 for the further experimental work in chapter 4.

Literature review

When using an accelerometer array to estimate the displacements of the hand of the user of a human-machine interface, physics dictate that we start from acceleration measurements and work our way to position measurements. Other authors have reported on the various steps in this process. We review here the most important results by starting from the acceleration measurements in sequence down to the orientation and position measurements.

1.0.1 Overview of the accelerometer-array technology

In biomechanics, anthropomorphic test devices (ATDs) are typically instrumented with arrays of accelerometers in order to estimate the trajectory of the head after an impact [2]. This was proposed by Mertz in 1967, who showed that an accelerometer array could be used to measure not only point accelerations, but also rigid-body rotations [14]. Many biomechanics researchers went further in this direction, e.g., Padgaonkar and King [15], Sinha (1976) [3], Mital and King (1979) [4], Linder et al. (2002) [21], Anderson et al. (2003) [6], Cappa et al. (2005) [22], Baron et al. (2009) [23].

Accelerometer arrays were also used in other biomechanics applications, such as gait analysis [13, 11, 12]. Morris (1973) and Hayes et al. (1983) used accelerometers to track the motions of the human shank. Impact accelerations measured by accelerometer arrays are widely used, like in the assessment of protective equipment, etc. Other biomechanics applications including the assessment of appropriate head protection with helmets for various human activities (e.g., aircrew helmets, American-football helmets, bicycling helmets) have also been studied [24].

Furthermore, accelerometer arrays were also proposed in other fields, such as robotics [25, 26, 27], vehicle guidance and control (Miles, 1986 [28]; Subramanian and Vendhan, 1993 [29]; Mostov et al., 1997 [30]; Peng and Golnaraghi, 2004 [31]; Franck and Nicolas, 2004 [32]; Sankaranaraynan et al., 2008 [33]) and inertial navigation systems (INS) (E. Edwan, 2009 [34]). Optical motion capture such as the Vicon [35] or even high-speed cameras are also commonly used in biomechanics to measure the limb displacements. Optical and inertial systems such as accelerometer arrays may be seen as complementary. Indeed, optical motion capture generally offers accurate pose estimates, but suffers from occlusion, lighting and portability problems. On the other hand, inertial systems do not rely on any external reference, and, therefore, provide reliable displacement-rate estimates, while the displacement estimates they provide are plagued by drift. Moreover, analyzing the motion that results from impacts requires high sampling rates, which generally impact more the price of optical systems than that of inertial ones. Indeed, optical systems generally involve more data acquisition and processing in a given iteration, which makes them more sensitive to a step-length reduction.

1.0.2 Estimating the angular-velocity

Existing methods

From these accelerometer-array designs, one can generally estimate the parameters of the rigid-body acceleration field, i.e., the acceleration of any of its points, its angular acceleration, and the quadratic powers of the components of its angular velocity. Several methods were proposed for estimating the angular velocity. These methods may be filed into two main categories which we call TI and PR, and which will be introduced in detail as follows.

There are currently two main methods for computing the angular velocity from accelerometer-array measurements. The first one uses the time-integration of the angular acceleration acquired from tangential acceleration measurements; We call it the Time Integration (TI) method. Use of this method is reported in the works of Chen et al. (1994) [17], Mostov et al. (1997) [30] and Kourosh Parsa et al. (2005) [20]. In the second method, the angular velocity is obtained from the quadratic forms of the angular velocity components; We call it the Polynomial Roots (PR) method. Schuler (1965) [36], Grammatikos (1965) [14], Parsa (2003), Peng and Golnaraghi (2004) [31] have all reported on this method.

However, the TI method is plagued by an important disadvantage. As mentioned above, the rigid-body angular velocity has to be calculated through a time-integration of the angular acceleration, a process that is unstable. This results in an error build-up, or drift, in the estimates, a phenomenon that was noted by Mital and King (1979) [4], among others. This behaviour can also be observed from the well known 3-2-2-2 accelerometer-array geometry [15], which is always coupled with the TI method. The inevitable consequence of this drift problem is that the angular-velocity estimates become inaccurate after less than a second, limiting the applicability of the TI method to short experiments.

The PR method also comes with its share of problems. The quadratic forms of the angular-velocity components, which are used in this method, are invariant to a sign change of the angular-velocity vector. When the values of these quadratic forms are known, this leaves ambiguity in the solution for the angular velocity, which is generally overcome by a comparison with the TI estimate. Moreover, the PR estimates are very sensitive to errors at low angular velocities, as shown in [2]. This lack of robustness at low angular velocities seems to restrict the use of the PR method to applications with high angular rates. An example of such an application is the guidance of autonomous naval support rounds [37, 10]. However, for most applications, its high sensitivity to errors at low angular velocities makes the PR method unreliable.

Improvement of existing methods

From the portrait above, it is apparent that the drawbacks of the PR and TI methods are different, which suggests that their combination would be beneficial. Parsa et al. [38] proposed such a combination, which is described in Appendix A. However, in their method, the relative accuracies of the TI and PR estimates are not taken into account. Alike the method proposed in our project, the one by Parsa et al. provides an estimate $\widetilde{\boldsymbol{\omega}}$ of the angular velocity from both tangential and centripetal acceleration measurements. However, the underlying idea is different, as it consists in using a first-order Taylor series approximation to correct the TI estimate by using the quadratic products of the angular velocity that come from the PR estimate. Cardou and Angeles [39] also proposed a combination based on extended Kalman filtering, which minimizes the variance of the errors. However, this method was only devised for planar motions, and, to our knowledge, it has not yet been extended to the spatial case. Perhaps more importantly, this method was found to be difficult to use in practice. Indeed, it models the angular jerk of the accelerometer array as white noise, and requires prior knowledge of the variance of this “noise”. In other words, with this method, the user needs to have some knowledge of the trajectories to be measured beforehand, which is generally not convenient. Therefore, a simpler combination is considered in this project.

It is worth noting that because of the quadratic form of PR method and also the disproportionate importance of the PR and TI method, there will be large errors at low angular velocities, and often sign errors. Therefore, a better combination of PR and TI is necessary to ensure that PR estimate should contribute more at high angular velocities but less at low ones, which will be discussed more in Section 2.3.3.

1.0.3 Estimating the orientation from the angular velocity

In this section, let us first review the main elementary methods that are available for computing the rigid-body orientation from its angular velocity. We then review more advanced methods combining the elementary ones. Finally, from the overall analysis, we find their advantages and, particularly, disadvantages so that our proposed method is necessary.

Existing basic methods

In inertial navigation systems (INS), the attitude of the moving vehicle is usually obtained by time integrating the angular velocity, which is measured directly by gyroscopes in an inertial reference frame [40, 41] or estimated from accelerometer measurements. With this method, however, small errors in the measurement of acceleration and angular velocity progressively accumulate into large errors with time passing. Thus, when used alone, all INS suffer from integration drift [42], which is why they are usually combined with other measurement systems.

Therefore, instead of being used as stand alone systems, inertial navigation systems are usually combined with other types of navigation systems. A common complementary sensor system is the GPS (Global Positioning System), which provides estimates of the object position from the triangulation of Earth-orbiting satellites. In this case, the GPS provides error-bounded low-frequency measurements to correct the unbounded-drift high-frequency measurements of the INS [43], while an INS can compensate for a momentary loss of the GPS signal. The GPS can provide location and time information in any type of weather, anywhere there is an unobstructed line of sight to four or more GPS satellites. For example, Edwan et al. (2009) [34] reported a scheme for the integration of gyro-free inertial-measurement units (IMUs) with the GPS. The GPS [44] does not work indoors, however, which prohibits its use in game controllers motion capture and other indoors applications. It is those applications that are targeted in this project.

For these applications where the GPS is inaccessible, the gravitational method or tilt-sensor method is often used to estimate the rigid-body orientation. Because, for some inertial sensors, typically like accelerometers, there exists the influence of a constant downward gravitational force, as we are on earth. By this approach, the gravity force is picked up by an accelerometer as an upward acceleration of $1g$ [45]. When the inertial acceleration is sufficiantly small, an accelerometer can be considered as a tilt sensor, and provides an estimate of the vertical direction [46, 47, 48, 49, 50, 51]. When a tilt sensor is submitted to high inertial accelerations, its output thus becomes contaminated with errors.

Notice, however, that using only gyroscope and accelerometers does not provide any information regarding the orientation of the game controller about a vertical axis. An extra sensor is needed to provide this piece of information. In the case of the Wii remote, the angle about the vertical, or azimuth angle, is estimated from two infrared LEDs that are fixed to the ground. The infrared light from these LEDs is measured by an infrared camera onboard the game controller. This constrains the Wii remote to maintain a line of sight between the camera and the LEDs.

Improvement of existing basic methods

Accordingly, neither the tilt-sensor method nor the time-integration method is recommended being used alone. Therefore, a combination of these two methods is necessary. Sabatini (2006) [52] developed a quaternion-based extended Kalman filter for determining the orientation of a rigid body from the outputs of inertial sensors. This method, however, cannot distinguish the gravitational and inertial acceleration, as it assumes that the inertial acceleration is negligible. Therefore, when there exists inertial acceleration, the variance of accelerometer noise is set to extremely high values, so as to force the filter to rely on the magnetic information. After that, Yun et al. (2008) [53] tried to estimate the orientation from Earth's gravity and magnetic field measurements, but it still suffered from motion disturbance (inertial acceleration) which means that a user had to move slowly. Huyghe et al. (2009) [54] presented the design of an orientation tracking algorithm based on 3D accelerometer and magnetometer measurements. This algorithm uses the unscented Kalman filter and is complemented with extra filters and parameters to reduce the influence of motion disturbance on the sensor signals. Although this is a good improvement, there is no systematic approach of choosing the most accurate values of filter parameters for different motion disturbances or with uncertainty for the case of large inertial acceleration. Hence, further research is still needed.

Some of other methods work very well but on the basis of one pivot point being fixed. Liu et al. (2009) [55] proposed a new method using an accelerometer array for analyzing the relative angles of the leg. However, their method is useful only under the assumption that the pivot point of the hip is fixed in the global frame or only the translational acceleration on the hip joint is considered. After that, Liu et al. (2011) [56] presented an original approach based on accelerometers and magnetometers for ambulatory estimation and analysis of 3D knee-joint kinematics, but under the same assumption, likewise, for Liu et al. (2011) [57]. Similarly, O'Donovan et al. (2007) [58] presented us a method of combining three sensors—a gyroscope, an accelerometer and a magnetometer—to estimate the joint angle of the leg, but the trunk was required to be immobile as well. In addition, Vikas et al. (2011) [59] gave us a novel approach to the dynamic, non-contact measurement of the inclination parameters of the rigid links of a serial chain using the Vestibular Dynamic Inclinometer (VDI), where there are two symmetrically-placed biaxial linear accelerometers and one single-axis gyroscope [60, 61]. This is of huge help for manipulators connected to a fixed point on the ground, but not for those rigid bodies moving freely in the space.

Regarding the game-controllers, on which our research is focused, human motion received much attention, many methods being suggested for human-limb tracking. For example, Marins et al. (2001) [62] used Gauss-Newton iteration algorithm to find the quaternion that best relates the measurements of linear accelerations and earth magnetic field. The optimal quaternion and the angular rate from the gyroscope are combined with a Kalman filter. This method is compared in Chapter 4 to those that are proposed there. Similarly, Yun et al. (2006) [63] suggested that extended Kalman filter (EKF) algorithm be combined with the Quaternion Estimator (QUEST) algorithm [64, 65], as the latter fuses the accelerometer and magnetometer measurements to produce an estimate of the rigid-body orientation represented by a quaternion. The quaternion is then merged with the gyroscope measurements with the EKF. This method can only work at low accelerations of the rigid body, however, as the QUEST algorithm assumes all measured accelerations are gravitational, thereby neglecting inertial accelerations. In addition, Ren et al. (2012) [43] used the MARG sensor to develop a prototype inertial measurement unit for tracking hand-held surgical instruments and proposed VecQua algorithm that integrates vectorized attitude mapping and EKF filtering based on quaternion state variables. The articles mentioned above all propose algorithms which have two stages. Thus, there is room for simplification, if we were able to trim the method to a single Kalman filter applied on a state-space system. Moreover, the proposed methods do not explicitly distinguish the inertial and gravitational accelerations, making them inaccurate at high accelerations. Last but not least, as the final objective is to obtain the rotation matrix, but not the quaternions nor the Euler angles, it is better to avoid these intermediate rotation parameters.

Given these problems of estimating rigid-body displacements in inertial navigation and game controllers, we set out to investigate a new way of estimating motion using an array of accelerometers.

1.0.4 Calibration of accelerometers and gyroscopes

As a part of an inertial navigation system (INS), a typical inertial measurement unit(IMU) [66] consists of accelerometers and gyroscopes which measure the linear accelerations and angular velocities, respectively. Generally, these output signals have to be numerically integrated to obtain either the position estimates, the orientation estimates or both [67]. As this integration process is inherently unstable, the IMU is very sensitive to measurement errors, which can be decreased by means of calibration.

As a matter of fact, most of accelerometers and gyroscopes are factory calibrated, allowing the user to avoid any further calibration for most of the applications now present in the market [68]. However, nonidealities such as scale factors, cross coupling, bias, and other higher-order nonlinearities affect the output of these inertial sensors, giving rise to unwanted errors. Therefore, in order to reach the higher accuracy requirement, further calibration procedures or algorithms are generated by some researchers.

The calibration of inertial sensors, especially for the gyroscope, mostly needs some velocity standards, like a rate table [69]. An accurate rate table, however, is somewhat expensive, more than 10000$, and even more for those that have three degree of freedom (DOF). It is evident that, for the comsumer-grade accelerometers, this is not suitable. Schopp et al. (2010) [70] designed well a gyroscope-free inertial measurement unit (GF-IMU) that only comprises linear accelerometers in order to directly measure the transversal acceleration as well as the angular acceleration and velocity. The accuracy improvements are well demonstrated after the calibration. The reference motion data and the measurements, however, are collected during a calibration run on a 3D rotation table, which is not accepted in our case. Similarly, Batista et al. (2011) [71] presented us a calibration technique for a tri-axial accelerometer and a novel dynamic filtering solution for the bias, which also accounts explicitly for the estimation of the gravity in body-fixed coordinates. In the tests, however, a 3D rate table (Model 2103HT from Ideal Aerosmith) was employed that provided ground truth signals for performance evaluation purposes, both for offline calibration and dynamic bias and gravity estimation. Similar costly equipment was also found in [72].

Certainly, some researches on calibration are done on one DOF rate table or less expensive equipment [73]. However, there are some defective points for some experimental designs. Because, as a matter of fact, the calibration algorithms are mostly from the same basic equation, which is a linear relationship between the sensor output and reference values. The principal problem is finally on the experimental design. However, many tests are static or quasi-static tests [74], which can not reach the full scale range of the accelerometers. Lee et al. (2011) [75] developed a method, which consists of a novel dynamic testbed with a combination of a single-axis rate table and an attitude change mount, to calibrate a low-quality, MEMS-based Inertial Measurement Unit (IMU). A Fourier Transform method is proposed, compared to the commonly used Recursive Least Squares method. However, the rate table is rotated at a constant angular velocity of only 10°/s. This indicates that the associated centripetal acceleration becomes approximately 60 μg, which can be ignored. This is quasi-static, while the sensing range of the accelerometer is ±4g. The similar phenomenon can be seen in [76], in which the multi-position static tests were performed to extract accelerometer scale factor error, misalignment and bias. Bachmann et al. (2003) [77] conducted two types of tests, in which both the purposes of the rotations was to allow the individual accelerometers to sense the gravity vector in all attitudes. Thus, the sensors remained quasi-static or static. For a human-machine interface, the object of this thesis, the accelerations often range beyond 1g. Hence, we prefer to use the full range of the accelerometers. For sensors with a range exceeding 1g, the static calibration relies on extrapolation, which is uncertain and at a risk of producing erroneous results.

Some other researchers have solved the range problem. For example, Nieminen et al. (2010) [67]

used the whole dynamic range of the accelerometers and well enhanced the standard multi-position calibration method using a one DOF rate table. The method is based on averaging the collected data, while some defective points over a long period of time may be taken into account when computing their mean value.

In addition to the calibration of accelerometers [78], the gyroscopes were calibrated as well, which can be seen in [79, 80, 81]. Hwangbo et al. (2008) [81] used a calibration apparatus called a universal right-angle iron as a rate table, but rotated by hand. The true angular velocity, however, was unknown. Therefore, additional rotations of known amplitudes were performed to determine the scale factor. Glueck et al. (2013) [80] reported only on offset calibration of a gyroscope exclusively based on the redundant information of one three-axis accelerometer. It is based on the knowledge that the output of any gyroscope equals zero if the inertial measurement unit is in a static position.

Therefore, in order to obtain the more accurate sensor parameters, we would like to propose new methods to calibrate the consumer-grade inertial sensors (two triaxial accelerometers and one triaxial gyroscope) installed in game controller. Meanwhile, given the factors mentioned above, like scale range of sensors, rate table cost, static or dynamic tests and so on, a less expensive rate table (one DOF) was used for the dynamic calibration.

Conclusion

This chapter firstly gives a general introduction on human-machine interface and its main applications on game-controllers. In most of these applications, inertial sensors play a vital and indispensable role. However, they still entail some diffculties and disadvantages. In order to solve these problems, the objectives of our project are established, as well as several milestone leading to their fulfillment. At each step, many references are reviewed. They are discussed and compared and the main disadvantages are mentioned so as to prepare for the presentation of our own proposed algorithms. In the following chapter, we begin this presentation with a method for estimating the angular velocity from accelerometer measurements.

Chapter 2

Estimating the angular velocity

Alle Wege führen nach Rom.

The objective of the thesis is to obtain the orientation of a game controller moving in space. Generally, for the inertial navigation system (INS), inertial sensors, i.e., accelerometers and gyroscopes, are used. The output of these sensors are point acceleration and angular velocity, which are the second and first order derivative of position and orientation, respectively. Moreover, estimating the angular velocity of a rigid body finds application in biomechanics, robotics and military systems [2]. Thus, the estimation of angular velocity is preferred as the first step. On the other hand, as mentioned in the chapter above, given the higher price, more power consumption and unsatisfactory performance at high accelerations, e.g., in crashworthiness and guided missile projectile [82], gyroscope is replaced by using accelerometer-array [2].

In this chapter, we investigate the problem of estimating the angular velocity from the measurements of m accelerometers, and propose a new method, which has already been published in [82]. In order to avoid the disadvantages and preserve the advantages of the methods mentioned in the chapter above, we combine the TI and PR estimates, under the assumption that the measurement errors consist in independently identically distributed zero-mean Gaussian white noise, as in ref. [39]. Unlike what was proposed in ref. [39], the two angular-velocity estimates are combined through a weighted sum that minimizes the variance of the resulting estimate. This blend preserves the robustness of the PR estimates at high angular velocities and that of the TI estimates at low angular velocities. Furthermore, it does not require prior knowledge of the estimated trajectory.

2.1 Accelerometer-array model

An accelerometer array is characterized by the positions and orientations of its m accelerometers. Fig. 2.1 represents the model of an accelerometer array, in which a rigid body moving in space is equipped with m accelerometers. $\mathbf{e}_i$ is a unit column vector representing the sensitive direction of the i^{th} accelerometer, while $\mathbf{r}_i$ represents its position with respect to an arbitrary rigid-body reference point B. Also, $\mathbf{b}_k$ gives the position of point B at the k^{th} time step, and $\mathbf{p}_{i,k}$ represents the position of the i^{th} accelerometer, both with respect to O.

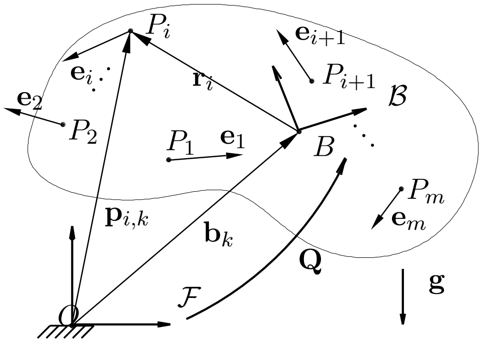


Figure 2.1: A rigid body equipped with m accelerometers moving in space

Let us consider a single accelerometer located at P_i. The measurement $\widehat{a}_{i,k}$ of the i^{th} accelerometer can be modeled as

$$\widehat{a}_{i,k} = a_{i,k} + \delta a_{i,k}, \tag{2.1}$$

where $a_{i,k}$ is the true acceleration component along the accelerometer sensitive direction, and $\delta a_{i,k}$ is the measurement error. In the case of perfect measurement, the acceleration is the combination of an inertial and a gravitational component, since, per Einstein's equivalence principle [83], the two cannot be distinguished. Formally, this gives

$$a_{i,k} = \mathbf{e}_i^T \ddot{\mathbf{p}}_{i,k} - \mathbf{e}_i^T \mathbf{Q}^T \mathbf{g} \tag{2.2}$$

where, $\mathbf{Q}$ is the rotation matrix from the fixed frame $\mathcal{F}$ to the moving frame $\mathcal{B}$. Firstly, rotate the rigid body around Z axis clockwise by an angle ϕ_Z. Then, rotate it around Y axis by an angle ϕ_Y, and finally, rotate it around X by an angle ϕ_X. Therefore, each rotation matrix is

$$\mathbf{Q}_Z = \begin{bmatrix} \cos\phi_Z & -\sin\phi_Z & 0 \\ \sin\phi_Z & \cos\phi_Z & 0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2.3}$$

$$\mathbf{Q}_Y = \begin{bmatrix} \cos\phi_Y & 0 & \sin\phi_Y \\ 0 & 1 & 0 \\ -\sin\phi_Y & 0 & \cos\phi_Y \end{bmatrix}, \tag{2.4}$$

$$\mathbf{Q}_X = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi_X & -\sin\phi_X \\ 0 & \sin\phi_X & \cos\phi_X \end{bmatrix}. \tag{2.5}$$

Then,

$$\mathbf{Q} = \mathbf{Q}_Z\mathbf{Q}_Y\mathbf{Q}_X$$
$$= \begin{bmatrix} \cos\phi_Y\cos\phi_Z & -\sin\phi_Z\cos\phi_X + \cos\phi_Z\sin\phi_Y\sin\phi_X & \sin\phi_Z\sin\phi_X + \cos\phi_Z\sin\phi_Y\cos\phi_X \\ \sin\phi_Z\cos\phi_Y & \cos\phi_Z\cos\phi_X + \sin\phi_Z\sin\phi_Y\sin\phi_X & -\cos\phi_Z\sin\phi_X + \sin\phi_Z\sin\phi_Y\cos\phi_X \\ -\sin\phi_Y & \cos\phi_Y\sin\phi_X & \cos\phi_Y\cos\phi_X \end{bmatrix}, \tag{2.6}$$

and $\mathbf{g}$ is the vector of gravitational acceleration

$$\mathbf{g} = [0 \; 0 \; -g]^T, \tag{2.7}$$

considering that if one directs the accelerometer sensitive axis downwards, it will yield a minus one g signal.

We assume that the measurement errors are zero-mean, identically, independently distributed Gaussian white noise $\delta\mathbf{a}$, i.e.,

$$\delta\mathbf{a}_k \sim \mathcal{N}(\mathbf{0}_m, \sigma_a^2\mathbf{1}_{m\times m}), \tag{2.8}$$

where, $\delta\mathbf{a}_k \equiv [\delta a_{1,k} \quad \cdots \quad \delta a_{m,k}]^T$, $\mathcal{N}$ represents the normal distribution, $\mathbf{0}_m$ is the m-dimensional zero vector, σ_a^2 is the measurement-error variance of a single accelerometer, and $\mathbf{1}_{m\times m}$ represents the $m \times m$ identity matrix.

The validity of this error model was assessed by measuring the outputs of twelve accelerometers subjected to no other accelerations than gravity. These twelve accelerometers are installed on a rigid body made by Pultex series 1525, which is presented in detail in Section 2.3. In order to verify whether the material is rigid and resistant enough, a finite element analysis was performed by two former members in robotics laboratory. We apply the results here without rewriting the process. A pressure of 0.8 MPa can only cause a deformation of 0.1 mm. Because of the acceleration limit from human hand and the mass of the rigid body, it gives far less than 0.8 MPa. Therefore, in our case, this material is rigid enough.

Then, three steps of tests were conducted for the mean value, autocorrelation and distribution, respectively. Firstly, we let accelerometers immoblie on the table for more than 24 hours, while the sampling time is 1 s. The RT-Lab is used to receive the outputs of accelerometers. Fig. 2.2 shows the error of one of the accelerometers, which is representative of all others. The fitted line ascends from -0.1478m/s^2 to -0.1450m/s^2 during this long time period, which indicates that the drift can be ignored. Meanwhile, it is evident that the mean value is small enough compared to $1g$. Thence, we can coarsely consider that the mean of accelerometer noise is null.

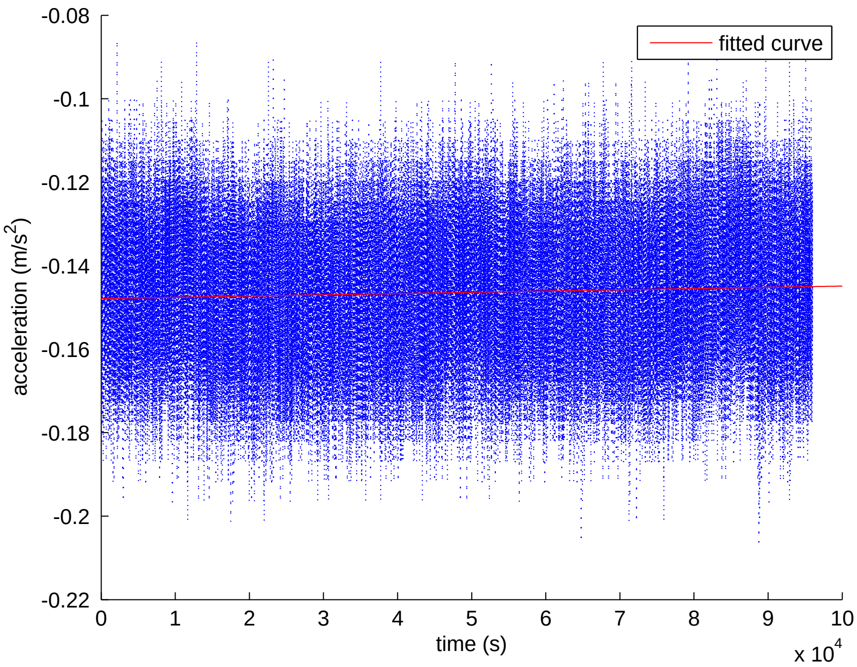
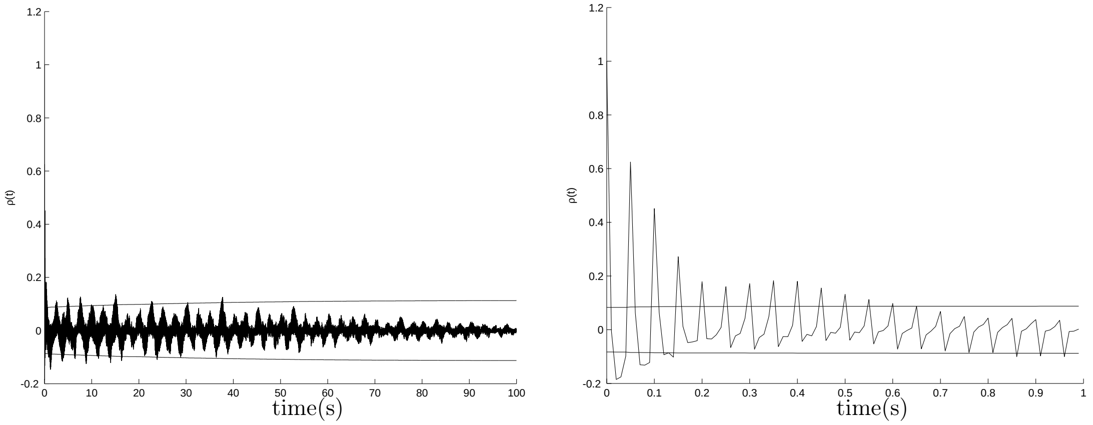


Figure 2.2: Bias of one of the accelerometers

Secondly, the self correlation of the measurement noise has been tested by tracing its autocorrelogram, which is presented in Fig. 2.3 (Fig. 2.3(b) is an enlarged map of Fig. 2.3(a)). For an ideal random time series, lagged values of the series are completely uncorrelated and the correlation coefficient $\rho(t) = 0$. We should observe that the lagged values should fall within the confidence limits. This is consistent with Fig. 2.3. In addition, there is a strong correlation at the beginning, which is normal because of correlation of "present on present". Combining the results from the first two steps, we come to the conclusion that the accelerometer noise is white noise.

Thirdly, the distribution of the accelerometer noise was also obtained. The noise was seperated into several intervals, and the probability of noise lying in each part was obtained, seen in Fig. 2.4. In this figure, the probability density curves are close to the symmetric bell curves. Therefore, we approximately consider that they are distributed normally, although many other distributions are bell-shaped (such as Cauchy's, Student's, and logistic). Finally, we assume that the noise of the accelerometers are Gaussian white noise.

From the rigid-body kinematics equations, the position, velocity and acceleration of the ac-



(a) Autocorrelogram over 100 s

(b) Autocorrelogram over 1 s

Figure 2.3: Autocorrelogram of one accelerometer (sampling period: Δt=0.01 s)

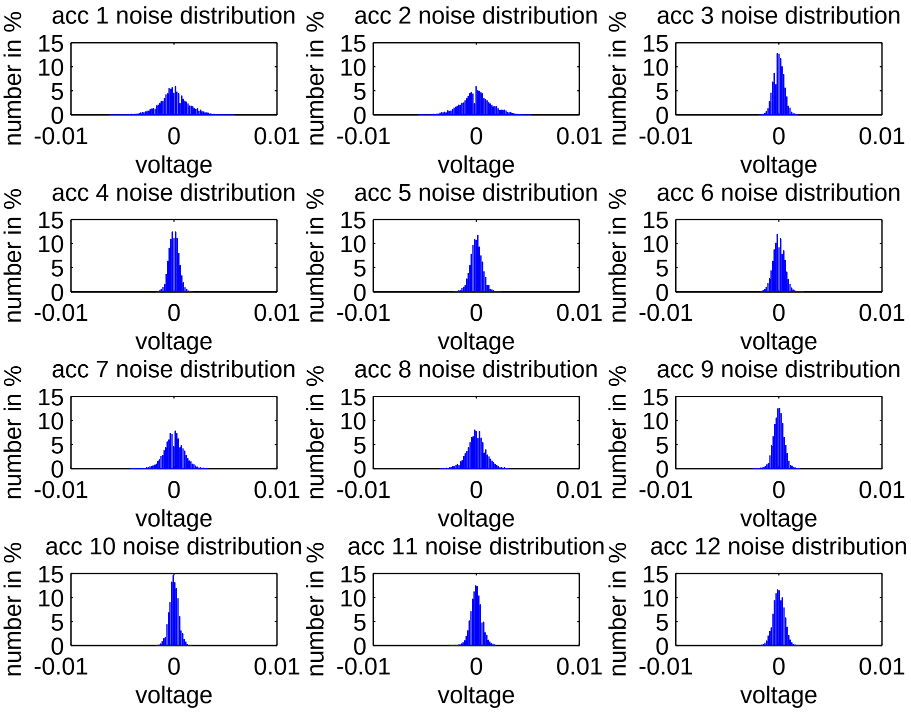


Figure 2.4: Noise distribution of accelerometers

celerometer P_i at time-step k are

$$\mathbf{p}_{i,k} = \mathbf{b}_k + \mathbf{r}_k, \tag{2.9}$$

$$\dot{\mathbf{p}}_{i,k} = \dot{\mathbf{b}}_k + \boldsymbol{\omega}_k \times \mathbf{r}_i, \tag{2.10}$$

$$\ddot{\mathbf{p}}_{i,k} = \ddot{\mathbf{b}}_k + \dot{\boldsymbol{\omega}}_k \times \mathbf{r}_i + \boldsymbol{\omega}_k \times (\boldsymbol{\omega}_k \times \mathbf{r}_i), \tag{2.11}$$

respectively, where $\boldsymbol{\omega}_k$ is the rigid-body angular-velocity vector at time step k. In the latter equation, $\dot{\boldsymbol{\omega}}_k \times \mathbf{r}_i = -\mathbf{r}_i \times \dot{\boldsymbol{\omega}}_k = \mathbf{R}_i^T \dot{\boldsymbol{\omega}}_k$ are the tangential accelerations, where $\mathbf{R}_i$ is the cross-product matrix of $\mathbf{r}_i$. $\boldsymbol{\omega}_k \times (\boldsymbol{\omega}_k \times \mathbf{r}_i) = \boldsymbol{\Sigma}_i^T \boldsymbol{\xi}_k$ are the centripetal accelerations, where $\boldsymbol{\xi}_k \equiv [\,\omega_{x,k}^2\ \omega_{y,k}^2\ \omega_{z,k}^2\ \omega_{y,k}\omega_{z,k}\ \omega_{z,k}\omega_{x,k}\ \omega_{x,k}\omega_{y,k}\,]^T$ and $\boldsymbol{\Sigma}_i^T \equiv \left[\mathrm{diag}(\mathbf{r}_i) - \mathbf{r}_i \mathbf{1}_3^T \quad \mathrm{sm}(\mathbf{r}_i)\right]^T$, where $\mathrm{sm}([x_1 \quad x_2 \quad x_3]^T) \equiv (\mathbf{1}_3^T \mathbf{x})(\mathbf{1}_3 \mathbf{1}_3^T - \mathbf{1}_{3\times 3}) + 2\mathrm{diag}(\mathbf{x}) - \mathbf{1}_3 \mathbf{x}^T - \mathbf{x}\mathbf{1}_3^T = \begin{bmatrix} 0 & x_3 & x_2 \\ x_3 & 0 & x_1 \\ x_2 & x_1 & 0 \end{bmatrix}$, sm stands for "special matrix", and $\mathbf{1}_n \equiv [1 \cdots 1]^T \in \mathbb{R}^n$. Therefore, upon substituting eq. (2.11) in eq. (2.2), we obtain

$$a_{i,k} = \mathbf{e}_i^T \ddot{\mathbf{b}}_k + \mathbf{e}_i^T (\mathbf{R}_i^T \dot{\boldsymbol{\omega}}_k) + \mathbf{e}_i^T \boldsymbol{\Sigma}_i^T \boldsymbol{\xi}_k - \mathbf{e}_i^T \mathbf{Q}_k^T \mathbf{g}, \tag{2.12}$$

which is the input-output equation of one accelerometer.

Upon stacking eq. (2.12), $i = 1, ..., m$, in an array, we obtain the input-output equations of an m-accelerometer array, i.e.,

$$\mathbf{A}\mathbf{z}_k = \mathbf{a}_k, \tag{2.13}$$

where $\mathbf{A} \equiv [\mathbf{A}_P\ \mathbf{A}_T\ \mathbf{A}_C] \in \mathbb{R}^{m\times 12}$, $\mathbf{A}_P \equiv [\mathbf{e}_1 \cdots \mathbf{e}_m]^T \in \mathbb{R}^{m\times 3}$, $\mathbf{A}_T \equiv [\mathbf{R}_1\mathbf{e}_1 \cdots \mathbf{R}_m\mathbf{e}_m]^T \in \mathbb{R}^{m\times 3}$ and $\mathbf{A}_C \equiv [\boldsymbol{\Sigma}_1 \mathbf{e}_1 \cdots \boldsymbol{\Sigma}_m \mathbf{e}_m]^T \in \mathbb{R}^{m\times 6}$, $\mathbf{z}_k \equiv \left[(\ddot{\mathbf{b}}_k - \mathbf{Q}_k^T \mathbf{g})^T\ \boldsymbol{\alpha}_k^T\ \boldsymbol{\xi}_k^T\right]^T \in \mathbb{R}^{12}$, $\boldsymbol{\alpha}_k = \dot{\boldsymbol{\omega}}_k$ is the rigid-body angular acceleration vector at time step k, and $\mathbf{a}_k \in \mathbb{R}^m$. Here, $\mathbf{A}_T$ stands for the tangential acceleration matrix, and $\mathbf{A}_C$ stands for the centripetal acceleration matrix, and $\mathbf{A}_P$ stands for the point-acceleration matrix.

2.2 The proposed estimation method

We wish to compute the most robust estimate $\widehat{\boldsymbol{\omega}}_k$ of $\boldsymbol{\omega}_k$ under the assumptions of Section 2.1. Since both the tangential and centripetal components of the rigid-body acceleration field contain information regarding $\boldsymbol{\omega}_k$, the strategy is to linearly combine the two estimates they provide in a way such that the variance of the resulting estimate is minimized. Hence, we must have

$$\widehat{\boldsymbol{\omega}}_k = \mathrm{diag}(\mathbf{w}_k)\widehat{\boldsymbol{\omega}}_{TI,k} + (\mathbf{1}_{3\times 3} - \mathrm{diag}(\mathbf{w}_k))\widehat{\boldsymbol{\omega}}_{PR,k}, \tag{2.14}$$

where, $\mathrm{diag}(\mathbf{w}_k)$ is the diagonal matrix of the weighting factors $\mathbf{w}_k \in \mathbb{R}^3$ and $0 \leqslant w_{i,k} \leqslant 1$. $\widehat{\boldsymbol{\omega}}_{TI,k}$ and $\widehat{\boldsymbol{\omega}}_{PR,k}$, which will be described in eq. (2.15) and eq. (2.16), are estimates from methods of time-integration and polynomial-roots, respectively. In order to minimize the covariance matrix $\boldsymbol{\Sigma}_{\omega,k}$ of $\widehat{\boldsymbol{\omega}}_k$, which amounts to minimizing the error of the estimate, we must first determine how it relates to the error of the accelerometer-array measurements $\delta\mathbf{a}_k$. This is the goal of Section 2.2.1.

2.2.1 The variance of $\widehat{\boldsymbol{\omega}}_k$

Let us first compute the variances of the estimate $\widehat{\boldsymbol{\omega}}_k$.

In eq. (2.14), the time-integration estimate is computed as

$$\widehat{\boldsymbol{\omega}}_{TI,k} = \widehat{\boldsymbol{\omega}}_{k-1} + \frac{1}{2}\Delta t(\widehat{\boldsymbol{\alpha}}_{k-1} + \widehat{\boldsymbol{\alpha}}_k), \tag{2.15}$$

whereas the polynomial-roots estimate is a complicated function of $\boldsymbol{\xi}_k$ alone, i.e.,

$$\widehat{\boldsymbol{\omega}}_{PR,k} = f(\boldsymbol{\xi}_k), \tag{2.16}$$

which was obtained in [2]. This function yields the solution to the equation

$$(\widehat{\mathbf{W}}_{s,k} - \text{tr}(\widehat{\mathbf{W}}_{s,k})\mathbf{1}_{3\times 3} - 2\|\widehat{\boldsymbol{\omega}}_{PR,k}\|_2^2\mathbf{1}_{3\times 3})\widehat{\boldsymbol{\omega}}_{PR,k} = \mathbf{0}_3, \tag{2.17}$$

where matrix $\widehat{\mathbf{W}}_{s,k}$ may be expressed as a linear function of $\widehat{\boldsymbol{\xi}}_k$, i.e.,

$$\widehat{\mathbf{W}}_{s,k} \equiv \text{sm}(\widehat{\boldsymbol{\xi}}_{2,k}) + \text{diag}(\widehat{\boldsymbol{\xi}}_{1,k}) - (\mathbf{1}_3^T\widehat{\boldsymbol{\xi}}_{1,k})\mathbf{1}_{3\times 3}. \tag{2.18}$$

From equations (2.14,2.15,2.16), we see that the angular-velocity estimate $\widehat{\boldsymbol{\omega}}_k$ is a function of $\widehat{\mathbf{y}}_k$, where

$$\widehat{\mathbf{y}}_k \equiv \begin{bmatrix} \widehat{\boldsymbol{\omega}}_{k-1}^T & \widehat{\boldsymbol{\alpha}}_{k-1}^T & \widehat{\boldsymbol{\alpha}}_k^T & \widehat{\boldsymbol{\xi}}_k^T \end{bmatrix}^T \in \mathbb{R}^{15}. \tag{2.19}$$

In order to estimate the covariance matrix $\boldsymbol{\Sigma}_{\omega,k}$ of $\widehat{\boldsymbol{\omega}}_k$, we must first estimate the statistical distribution of $\widehat{\mathbf{y}}_k$. This statistical distribution is assumed to be Gaussian, i.e.,

$$\widehat{\mathbf{y}}_k = \mathbf{y}_k + \delta\mathbf{y}_k, \tag{2.20}$$

where

$$\delta\mathbf{y}_k \sim \mathcal{N}(\mathbf{0}_{15}, \boldsymbol{\Sigma}_{y,k}). \tag{2.21}$$

Matrix $\boldsymbol{\Sigma}_{y,k}$ is computed as

$$\begin{aligned}\boldsymbol{\Sigma}_{y,k} &\equiv E(\delta\mathbf{y}_k\delta\mathbf{y}_k^T) \qquad (2.22)\\ &= \begin{bmatrix} E(\delta\boldsymbol{\omega}_{k-1}\delta\boldsymbol{\omega}_{k-1}^T) & E(\delta\boldsymbol{\omega}_{k-1}\delta\boldsymbol{\alpha}_{k-1}^T) & E(\delta\boldsymbol{\omega}_{k-1}\delta\boldsymbol{\alpha}_k^T) & E(\delta\boldsymbol{\omega}_{k-1}\delta\boldsymbol{\xi}_k^T) \\ E(\delta\boldsymbol{\alpha}_{k-1}\delta\boldsymbol{\omega}_{k-1}^T) & E(\delta\boldsymbol{\alpha}_{k-1}\delta\boldsymbol{\alpha}_{k-1}^T) & E(\delta\boldsymbol{\alpha}_{k-1}\delta\boldsymbol{\alpha}_k^T) & E(\delta\boldsymbol{\alpha}_{k-1}\delta\boldsymbol{\xi}_k^T) \\ E(\delta\boldsymbol{\alpha}_k\delta\boldsymbol{\omega}_{k-1}^T) & E(\delta\boldsymbol{\alpha}_k\delta\boldsymbol{\alpha}_{k-1}^T) & E(\delta\boldsymbol{\alpha}_k\delta\boldsymbol{\alpha}_k^T) & E(\delta\boldsymbol{\alpha}_k\delta\boldsymbol{\xi}_k^T) \\ E(\delta\boldsymbol{\xi}_k\delta\boldsymbol{\omega}_{k-1}^T) & E(\delta\boldsymbol{\xi}_k\delta\boldsymbol{\alpha}_{k-1}^T) & E(\delta\boldsymbol{\xi}_k\delta\boldsymbol{\alpha}_k^T) & E(\delta\boldsymbol{\xi}_k\delta\boldsymbol{\xi}_k^T) \end{bmatrix},\end{aligned}$$

where $E(\cdot)$ is the expectation of its random-variable argument. Because of the independence or noncorrelation, some components of this matrix may be null. Let us start from this.

Recall that $E(\mathbf{x}\mathbf{y}^T) = \mathbf{0}_{n\times n}$ if $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^n$ are independent. Since samples $\mathbf{a}_{k-1}$ and $\mathbf{a}_k$ are assumed to be independent, we readily have

$$\begin{aligned} E(\delta\boldsymbol{\alpha}_k \delta\boldsymbol{\omega}_{k-1}^T) &= E(\delta\boldsymbol{\alpha}_k \delta\boldsymbol{\alpha}_{k-1}^T) = \mathbf{0}_{3\times 3}, \\ E(\delta\boldsymbol{\xi}_k \delta\boldsymbol{\omega}_{k-1}^T) &= E(\delta\boldsymbol{\xi}_k \delta\boldsymbol{\alpha}_{k-1}^T) = \mathbf{0}_{6\times 3}, \end{aligned} \tag{2.23}$$

Moreover, we define

$$E(\delta\boldsymbol{\omega}_{k-1} \delta\boldsymbol{\omega}_{k-1}^T) \equiv \boldsymbol{\Sigma}_{\omega,k-1}, \tag{2.24}$$

which is assumed to be known from the previous time step.

On the other hand, eq. (2.15) and eq. (2.14) are the equations on the estimates of angular velocities. They can be also applied to the true angular velocities $\boldsymbol{\omega}_k$ and their errors $\delta\boldsymbol{\omega}_k$. Therefore, the relation between $\delta\boldsymbol{\omega}_{k-1}$ and $\delta\boldsymbol{\alpha}_{k-1}$ may be obtained after substituting eq. (2.15) into eq. (2.14), which gives

$$\begin{aligned} \delta\boldsymbol{\omega}_{k-1} =& \mathrm{diag}(\mathbf{w}_{k-1})(\delta\boldsymbol{\omega}_{k-2} + \frac{1}{2}\Delta t(\delta\boldsymbol{\alpha}_{k-2} + \delta\boldsymbol{\alpha}_{k-1})) + \\ & (\mathbf{1}_{3\times 3} - \mathrm{diag}(\mathbf{w}_{k-1}))\delta\boldsymbol{\omega}_{PR,k}. \end{aligned} \tag{2.25}$$

From this equation, we obtain

$$\begin{aligned} \delta\boldsymbol{\omega}_{k-1}\delta\boldsymbol{\alpha}_{k-1}^T =& \mathrm{diag}(\mathbf{w}_{k-1})(\delta\boldsymbol{\omega}_{k-2} + \frac{1}{2}\Delta t(\delta\boldsymbol{\alpha}_{k-2} + \delta\boldsymbol{\alpha}_{k-1}))\delta\boldsymbol{\alpha}_{k-1}^T + \\ & (\mathbf{1}_{3\times 3} - \mathrm{diag}(\mathbf{w}_{k-1}))\delta\boldsymbol{\omega}_{PR,k}\delta\boldsymbol{\alpha}_{k-1}^T, \end{aligned} \tag{2.26}$$

and

$$\begin{aligned} E(\delta\boldsymbol{\omega}_{k-1}\delta\boldsymbol{\alpha}_{k-1}^T) &= E((1/2)\Delta t \mathrm{diag}(\mathbf{w}_{k-1})\delta\boldsymbol{\alpha}_{k-1}\delta\boldsymbol{\alpha}_{k-1}^T), \\ &= \frac{1}{2}\Delta t \mathrm{diag}(\mathbf{w}_{k-1})E(\delta\boldsymbol{\alpha}_{k-1}\delta\boldsymbol{\alpha}_{k-1}^T) \end{aligned} \tag{2.27}$$

In turn, from eq. (2.13), $\mathbf{z}_{k-1}$ is computed as

$$\mathbf{z}_{k-1} = \mathbf{A}^{\dagger}\mathbf{a}_{k-1}, \tag{2.28}$$

where $\mathbf{A}^{\dagger} \equiv (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T$ is the left Moore-Penrose pseudoinverse of $\mathbf{A}$. Thus, we have

$$\begin{aligned} E(\delta\mathbf{z}_k\delta\mathbf{z}_k^T) &= E(\mathbf{A}^{\dagger}\delta\mathbf{a}_k\delta\mathbf{a}_k^T\mathbf{A}^{\dagger T}), \\ &= \mathbf{A}^{\dagger}E(\delta\mathbf{a}_k\delta\mathbf{a}_k^T)\mathbf{A}^{\dagger T}, \\ &= \sigma_a^2\mathbf{A}^{\dagger}\mathbf{A}^{\dagger T}, \\ &= \sigma_a^2(\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{A}(\mathbf{A}^T\mathbf{A})^{-1}, \\ &= \sigma_a^2(\mathbf{A}^T\mathbf{A})^{-1}. \end{aligned} \tag{2.29}$$

Let us define

$$\mathbf{B} \equiv (\mathbf{A}^T\mathbf{A})^{-1}. \tag{2.30}$$

Then, we partition $\mathbf{B}$ as

$$\mathbf{B} \equiv \begin{bmatrix} \mathbf{B}_P & \mathbf{B}_{PT} & \mathbf{B}_{PC} \\ \mathbf{B}_{PT}^T & \mathbf{B}_T & \mathbf{B}_{TC} \\ \mathbf{B}_{PC}^T & \mathbf{B}_{TC}^T & \mathbf{B}_C \end{bmatrix},$$

so that

$$\begin{aligned} E(\delta\mathbf{z}_k\delta\mathbf{z}_k^T) &= E\left(\begin{bmatrix} \delta\ddot{\mathbf{b}}_k\delta\ddot{\mathbf{b}}_k^T & \delta\ddot{\mathbf{b}}_k\delta\boldsymbol{\alpha}_k^T & \delta\ddot{\mathbf{b}}_k\delta\boldsymbol{\xi}_k^T \\ \delta\boldsymbol{\alpha}_k\delta\ddot{\mathbf{b}}_k^T & \delta\boldsymbol{\alpha}_k\delta\boldsymbol{\alpha}_k^T & \delta\boldsymbol{\alpha}_k\delta\boldsymbol{\xi}_k^T \\ \delta\boldsymbol{\xi}_k\delta\ddot{\mathbf{b}}_k^T & \delta\boldsymbol{\xi}_k\delta\boldsymbol{\alpha}_k^T & \delta\boldsymbol{\xi}_k\delta\boldsymbol{\xi}_k^T \end{bmatrix}\right) \\ &= \begin{bmatrix} \mathbf{B}_P & \mathbf{B}_{PT} & \mathbf{B}_{PC} \\ \mathbf{B}_{PT}^T & \mathbf{B}_T & \mathbf{B}_{TC} \\ \mathbf{B}_{PC}^T & \mathbf{B}_{TC}^T & \mathbf{B}_C \end{bmatrix}\sigma_a^2. \end{aligned} \tag{2.31}$$

We substitute the relation $E(\delta\boldsymbol{\alpha}_{k-1}\delta\boldsymbol{\alpha}_{k-1}^T) = \mathbf{B}_T\sigma_a^2$ in eq. (2.27), which gives

$$E(\delta\boldsymbol{\omega}_{k-1}\delta\boldsymbol{\alpha}_{k-1}^T) = \frac{1}{2}\Delta t\,\mathrm{diag}(\mathbf{w}_{k-1})\mathbf{B}_T\sigma_a^2. \tag{2.32}$$

We proceed with the balance of the blocks of $\mathbf{\Sigma}_{y,k}$ as given in eq. (2.22), which yields

$$\begin{aligned} E(\delta\boldsymbol{\alpha}_{k-1}\delta\boldsymbol{\alpha}_{k-1}^T) = E(\delta\boldsymbol{\alpha}_k\delta\boldsymbol{\alpha}_k^T) &= \mathbf{B}_T\sigma_a^2, \\ E(\delta\boldsymbol{\alpha}_k\delta\boldsymbol{\xi}_k^T) &= \mathbf{B}_{TC}\sigma_a^2, \\ E(\delta\boldsymbol{\xi}_k\delta\boldsymbol{\xi}_k^T) &= \mathbf{B}_C\sigma_a^2 \end{aligned} \tag{2.33}$$

Hence, matrix $\mathbf{\Sigma}_{y,k}$ is completely defined by the blocks of eqs. (2.23,2.24,2.32,2.33), namely,

$$\mathbf{\Sigma}_{y,k} = \begin{bmatrix} \mathbf{\Sigma}_{\omega,k-1} & \frac{1}{2}\Delta t\,\mathrm{diag}(\mathbf{w}_{k-1})\mathbf{B}_T\sigma_a^2 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times6} \\ \frac{1}{2}\Delta t\mathbf{B}_T\mathrm{diag}(\mathbf{w}_{k-1})\sigma_a^2 & \mathbf{B}_T\sigma_a^2 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times6} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{B}_T\sigma_a^2 & \mathbf{B}_{TC}\sigma_a^2 \\ \mathbf{0}_{6\times3} & \mathbf{0}_{6\times3} & \mathbf{B}_{TC}^T\sigma_a^2 & \mathbf{B}_C\sigma_a^2 \end{bmatrix}. \tag{2.34}$$

Having computed the variance of $\widehat{\mathbf{y}}_k$, we set out to compute the variance of the error on the angular-velocity estimate $\boldsymbol{\omega}_k$,

$$\mathbf{\Sigma}_{\omega,k} = E(\delta\boldsymbol{\omega}_k\delta\boldsymbol{\omega}_k^T). \tag{2.35}$$

From eq. (2.14), for small errors $\delta\mathbf{y}_k$, $\delta\boldsymbol{\omega}_k$ may be estimated as

$$\delta\boldsymbol{\omega}_k \approx \frac{\partial\widehat{\boldsymbol{\omega}}_k}{\partial\widehat{\mathbf{y}}_k}\cdot\delta\mathbf{y}_k = \operatorname{diag}(\mathbf{w}_k)\frac{\partial\widehat{\boldsymbol{\omega}}_{TI,k}}{\partial\widehat{\mathbf{y}}_k}\delta\mathbf{y}_k + (\mathbf{1}_{3\times 3} - \operatorname{diag}(\mathbf{w}_k))\frac{\partial\widehat{\boldsymbol{\omega}}_{PR,k}}{\partial\widehat{\mathbf{y}}_k}\delta\mathbf{y}_k. \tag{2.36}$$

In turn, from eq. (2.15), we have

$$\frac{\partial\widehat{\boldsymbol{\omega}}_{TI,k}}{\partial\widehat{\mathbf{y}}_k} = \begin{bmatrix}\mathbf{1}_{3\times 3} & \frac{1}{2}\Delta t\mathbf{1}_{3\times 3} & \frac{1}{2}\Delta t\mathbf{1}_{3\times 3} & \mathbf{0}_{3\times 6}\end{bmatrix}, \tag{2.37}$$

whereas computing $\partial\widehat{\boldsymbol{\omega}}_{PR,k}/\partial\widehat{\mathbf{y}}_k$ is more difficult. However, first, notice that $\widehat{\boldsymbol{\omega}}_{PR,k}$ only depends on $\widehat{\boldsymbol{\xi}}_k$, so that

$$\frac{\partial\widehat{\boldsymbol{\omega}}_{PR,k}}{\partial\widehat{\mathbf{y}}_k} = \begin{bmatrix}\mathbf{0}_{3\times 9} & \frac{\partial\widehat{\boldsymbol{\omega}}_{PR,k}}{\partial\widehat{\boldsymbol{\xi}}_k}\end{bmatrix} \in \mathbb{R}^{3\times 15}. \tag{2.38}$$

The problem now amounts to computing $\partial\widehat{\boldsymbol{\omega}}_{PR,k}/\partial\widehat{\boldsymbol{\xi}}_k$. To this end, we define $\widehat{\boldsymbol{\xi}}_k \equiv \begin{bmatrix}\widehat{\boldsymbol{\xi}}_{1,k}^T & \widehat{\boldsymbol{\xi}}_{2,k}^T\end{bmatrix}^T$, where $\widehat{\boldsymbol{\xi}}_{1,k} \equiv \begin{bmatrix}\widehat{\omega}_{x,k}^2 & \widehat{\omega}_{y,k}^2 & \widehat{\omega}_{z,k}^2\end{bmatrix}^T$ and $\widehat{\boldsymbol{\xi}}_{2,k} \equiv [\widehat{\omega}_{y,k}\widehat{\omega}_{z,k} \;\; \widehat{\omega}_{z,k}\widehat{\omega}_{x,k} \;\; \widehat{\omega}_{x,k}\widehat{\omega}_{y,k}]^T$.

From [2], the optimum angular-velocity estimate $\widehat{\boldsymbol{\omega}}_{PR,k}$ is obtained by solving the set of equations

$$(\widehat{\mathbf{W}}_{s,k} - \operatorname{tr}(\widehat{\mathbf{W}}_{s,k})\mathbf{1}_{3\times 3} - 2\|\widehat{\boldsymbol{\omega}}_{PR,k}\|_2^2\mathbf{1}_{3\times 3})\widehat{\boldsymbol{\omega}}_{PR,k} = \mathbf{0}_3.$$

Upon substituting eq. (2.18) into eq. (2.17), and differentiating both sides with respect to $\widehat{\boldsymbol{\xi}}_k$ by means of computer-assisted symbolic calculations, we obtain

$$\widehat{\mathbf{Z}}_k\frac{\partial\widehat{\boldsymbol{\omega}}_{PR,k}}{\partial\widehat{\boldsymbol{\xi}}_k} = \widehat{\boldsymbol{\Psi}}_k, \tag{2.39}$$

where

$$\widehat{\mathbf{Z}}_k \equiv -3(\operatorname{sm}(\widehat{\boldsymbol{\xi}}_{2,k}) + \operatorname{diag}(\widehat{\boldsymbol{\xi}}_{1,k})) - \left(\widehat{\boldsymbol{\xi}}_{1,k}^T\mathbf{1}_3\right)\mathbf{1}_{3\times 3}, \tag{2.40}$$

$$\widehat{\boldsymbol{\Psi}}_k \equiv \begin{bmatrix}\widehat{\boldsymbol{\omega}}_{TI,k}\mathbf{1}_3^T + \operatorname{diag}(\widehat{\boldsymbol{\omega}}_{TI,k}) & \operatorname{sm}(\widehat{\boldsymbol{\omega}}_{PR,k})\end{bmatrix}. \tag{2.41}$$

We compute $\partial\widehat{\boldsymbol{\omega}}_{PR,k}/\partial\widehat{\boldsymbol{\xi}}_k$ by solving eq. (2.39) as

$$\begin{aligned}\frac{\partial\widehat{\omega}_{PR,k}}{\partial\widehat{\boldsymbol{\xi}}_k} &= \widehat{\mathbf{Z}}_k^{-1}\widehat{\boldsymbol{\Psi}}_k \\ &= \left(\frac{3}{4}\frac{1}{\|\widehat{\boldsymbol{\xi}}_{1,k}\|_2^2 + 2\|\widehat{\boldsymbol{\xi}}_{2,k}\|_2^2}(\operatorname{sm}(\widehat{\boldsymbol{\xi}}_{2,k}) + \operatorname{diag}(\widehat{\boldsymbol{\xi}}_{1,k}))- \right. \\ &\quad \left.\frac{\widehat{\boldsymbol{\xi}}_{1,k}^T\mathbf{1}_3}{\|\widehat{\boldsymbol{\xi}}_{1,k}\|_2^2 + 2\|\widehat{\boldsymbol{\xi}}_{2,k}\|_2^2}\mathbf{1}_{3\times 3}\right)\widehat{\boldsymbol{\Psi}}_k.\end{aligned} \tag{2.42}$$

Upon resubstituting eq. (2.42) into eq. (2.38), we obtain

$$\frac{\partial \widehat{\boldsymbol{\omega}}_k}{\partial \widehat{\mathbf{y}}_k} \equiv \left[\mathrm{diag}(\mathbf{w}_k) \quad \frac{1}{2}\Delta t \mathrm{diag}(\mathbf{w}_k) \quad \frac{1}{2}\Delta t \mathrm{diag}(\mathbf{w}_k) \quad (\mathbf{1}_{3\times 3} - \mathrm{diag}(\mathbf{w}_k))\widehat{\mathbf{Z}}_k^{-1}\widehat{\boldsymbol{\Psi}}_k\right]. \tag{2.43}$$

We substitute this last equation into eq. (2.36), and, in turn, into eq. (2.35), which yields

$$\begin{aligned}
\boldsymbol{\Sigma}_{\omega,k} &= E\left(\frac{\partial \widehat{\boldsymbol{\omega}}_k}{\partial \widehat{\mathbf{y}}_k}\delta\widehat{\mathbf{y}}_k \; \delta\widehat{\mathbf{y}}_k^T(\frac{\partial \widehat{\boldsymbol{\omega}}_k}{\partial \widehat{\mathbf{y}}_k})^T\right), \\
&= \frac{\partial \widehat{\boldsymbol{\omega}}_k}{\partial \widehat{\mathbf{y}}_k} E(\delta\widehat{\mathbf{y}}_k\delta\widehat{\mathbf{y}}_k^T)\left(\frac{\partial \widehat{\boldsymbol{\omega}}_k}{\partial \widehat{\mathbf{y}}_k}\right)^T, \\
&= \frac{\partial \widehat{\boldsymbol{\omega}}_k}{\partial \widehat{\mathbf{y}}_k}\boldsymbol{\Sigma}_{y,k}\left(\frac{\partial \widehat{\boldsymbol{\omega}}_k}{\partial \widehat{\mathbf{y}}_k}\right)^T, \\
&= \mathrm{diag}(\mathbf{w}_k)\boldsymbol{\Sigma}_{\omega,k-1}\mathrm{diag}(\mathbf{w}_k) \\
&\quad + \sigma_a^2\frac{1}{4}\Delta t^2 \mathrm{diag}(\mathbf{w}_k)\mathrm{diag}(\mathbf{w}_{k-1})\mathbf{B}_T\mathrm{diag}(\mathbf{w}_k) \\
&\quad + \sigma_a^2\frac{1}{4}\Delta t^2 \mathrm{diag}(\mathbf{w}_k)\mathbf{B}_T\mathrm{diag}(\mathbf{w}_{k-1})\mathrm{diag}(\mathbf{w}_k) \\
&\quad + \sigma_a^2\frac{1}{2}\Delta t^2 \mathrm{diag}(\mathbf{w}_k)\mathbf{B}_T\mathrm{diag}(\mathbf{w}_k) \\
&\quad + \frac{1}{2}\Delta t\sigma_a^2(\mathbf{1}_{3\times 3} - \mathrm{diag}(\mathbf{w}_k))\widehat{\mathbf{Z}}_k^{-1}\widehat{\boldsymbol{\Psi}}_k\mathbf{B}_{TC}^T\mathrm{diag}(\mathbf{w}_k) \\
&\quad + \frac{1}{2}\Delta t\sigma_a^2\mathrm{diag}(\mathbf{w}_k)\mathbf{B}_{TC}\widehat{\boldsymbol{\Psi}}_k^T\widehat{\mathbf{Z}}_k^{-T}(\mathbf{1}_{3\times 3} - \mathrm{diag}(\mathbf{w}_k)) \\
&\quad + \sigma_a^2(\mathbf{1}_{3\times 3} - \mathrm{diag}(\mathbf{w}_k))\widehat{\mathbf{Z}}_k^{-1}\widehat{\boldsymbol{\Psi}}_k\mathbf{B}_C\widehat{\boldsymbol{\Psi}}_k^T\widehat{\mathbf{Z}}_k^{-T} \\
&\quad (\mathbf{1}_{3\times 3} - \mathrm{diag}(\mathbf{w}_k)),
\end{aligned} \tag{2.44}$$

where, $\frac{\partial \widehat{\boldsymbol{\omega}}_k}{\partial \widehat{\mathbf{y}}_k}$ is given by eq. (2.43) and $\boldsymbol{\Sigma}_{y,k}$, by eq. (2.34).

The resulting equation is recursive in that it requires knowledge of $\boldsymbol{\Sigma}_{\omega,k}$ from the previous time step. In fact, eq. (2.44) is a type of Riccati equation. With the covariance matrix expressed as a function of the accelerometer-array parameters, we are set to find best estimate $\widehat{\boldsymbol{\omega}}_k$ of $\boldsymbol{\omega}_k$.

2.2.2 The optimum weights $\mathbf{w}_k$

As mentioned above, we are to choose the weighting factors $\mathbf{w}_k$ that minimize the variance of $\widehat{\boldsymbol{\omega}}_k$. By the variance of $\widehat{\boldsymbol{\omega}}_k$, we refer to the sum of the variances of the entries of $\widehat{\boldsymbol{\omega}}_k$. This quantity is directly given by the trace of the covariance matrix $\boldsymbol{\Sigma}_{\omega,k}$. Notice that $\mathrm{tr}(\mathrm{diag}(\mathbf{w}_k)\mathbf{D}\mathrm{diag}(\mathbf{w}_k)) = \mathbf{w}_k^T\mathrm{diag}(\mathrm{diag}(\mathbf{D}))\mathbf{w}_k$, where $\mathbf{D}$ is a square matrix, $\mathrm{diag}(\mathrm{diag}(\mathbf{D}))$ is a diagonal matrix whose diagonal entries are the diagonal entries of $\mathbf{D}$, and $\mathrm{tr}(\mathbf{D}\mathrm{diag}(\mathbf{w}_k)) =$

$\mathbf{w}_k^T \mathrm{diag}(\mathbf{D})$. Hence, from eq. (2.44), we obtain

$$\begin{aligned}\mathrm{tr}(\boldsymbol{\Sigma}_{\omega,k}) = {} & \mathbf{w}_k^T \mathrm{diag}(\mathrm{diag}(\boldsymbol{\Sigma}_{\omega,k-1} + \frac{1}{4}\Delta t^2 \sigma_a^2 (\mathbf{B}_T \mathrm{diag}(\mathbf{w}_{k-1}) \\ & + \mathrm{diag}(\mathbf{w}_{k-1})\mathbf{B}_T + 2\mathbf{B}_T)))\mathbf{w}_k \\ & + \frac{1}{2}\Delta t \sigma_a^2 \mathrm{tr}((\mathbf{1}_{3\times 3} - \mathrm{diag}(\mathbf{w}_k))\widehat{\mathbf{Z}}_k^{-1}\widehat{\boldsymbol{\Psi}}_k \mathbf{B}_{TC}^T \mathrm{diag}(\mathbf{w}_k) \\ & + \mathrm{diag}(\mathbf{w}_k)\mathbf{B}_{TC}\widehat{\boldsymbol{\Psi}}_k^T \widehat{\mathbf{Z}}_k^{-T}(\mathbf{1}_{3\times 3} - \mathrm{diag}(\mathbf{w}_k))) \\ & + \sigma_a^2 (\mathbf{1}_3 - \mathbf{w}_k)^T \mathrm{diag}(\mathrm{diag}(\widehat{\mathbf{Z}}_k^{-1}\widehat{\boldsymbol{\Psi}}_k \mathbf{B}_C \widehat{\boldsymbol{\Psi}}_k^T \widehat{\mathbf{Z}}_k^{-T})) \\ & (\mathbf{1}_3 - \mathbf{w}_k), \\ = {} & \mathbf{w}_k^T \mathrm{diag}(\mathrm{diag}(\boldsymbol{\Sigma}_{\omega,k-1}))\mathbf{w}_k \\ & + \mathbf{w}_k^T \mathrm{diag}(\mathrm{diag}(\mathbf{T}_k))\mathbf{w}_k \\ & + (\mathbf{w}_k^T \mathrm{diag}(\mathbf{M}_k) - \mathbf{w}_k^T \mathrm{diag}(\mathrm{diag}(\mathbf{M}_k))\mathbf{w}_k) \\ & + (\mathbf{1}_3 - \mathbf{w}_k)^T \mathrm{diag}(\mathrm{diag}(\mathbf{C}_k))(\mathbf{1}_3 - \mathbf{w}_k),\end{aligned} \tag{2.45}$$

where $\mathbf{T}_k \equiv \frac{1}{4}\Delta t^2 \sigma_a^2 (2\mathbf{B}_T + \mathbf{B}_T \mathrm{diag}(\mathbf{w}_{k-1}) + \mathrm{diag}(\mathbf{w}_{k-1})\mathbf{B}_T)$, $\mathbf{C}_k \equiv \widehat{\mathbf{Z}}_k^{-1}\widehat{\boldsymbol{\Psi}}_k \mathbf{B}_C \widehat{\boldsymbol{\Psi}}_k^T \widehat{\mathbf{Z}}_k^{-T}\sigma_a^2$ and $\mathbf{M}_k \equiv \Delta t \sigma_a^2 \widehat{\mathbf{Z}}_k^{-1}\widehat{\boldsymbol{\Psi}}_k \mathbf{B}_{TC}^T$.

The variance of $\widehat{\boldsymbol{\omega}}_k$ is minimized whenever

$$\partial\left(\mathrm{tr}(\boldsymbol{\Sigma}_{\omega,k})\right)/\partial \mathbf{w}_k = \mathbf{0}_3, \tag{2.46}$$

that is, when

$$\begin{aligned}\partial\left(\mathrm{tr}(\boldsymbol{\Sigma}_{\omega,k})\right)/\partial \mathbf{w}_k = {} & 2\mathrm{diag}(\mathrm{diag}(\boldsymbol{\Sigma}_{\omega,k-1}))\mathbf{w}_k \\ & + 2\mathrm{diag}(\mathrm{diag}(\mathbf{T}_k))\mathbf{w}_k + \mathrm{diag}(\mathbf{M}_k) \\ & - 2\mathbf{w}_k \mathrm{diag}(\mathrm{diag}(\mathbf{M}_k)) \\ & - 2\mathrm{diag}(\mathrm{diag}(\mathbf{C}_k))(\mathbf{1}_3 - \mathbf{w}_k), \\ = {} & \mathbf{0}_3,\end{aligned} \tag{2.47}$$

or

$$\begin{aligned}\mathbf{w}_k = {} & (2\mathrm{diag}(\mathrm{diag}(\boldsymbol{\Sigma}_{\omega,k-1} + \mathbf{T}_k - \mathbf{M}_k \\ & + \mathbf{C}_k)))^{-1}(2\mathrm{diag}(\mathrm{diag}(\mathbf{C}_k))\mathbf{1}_3 \\ & - \mathrm{diag}(\mathbf{M}_k)).\end{aligned} \tag{2.48}$$

From eq. (2.48), it is not clear whether the resulting weights should always remain within the interval $[0, 1]$ or not. As we will see, experiments tend to confirm this assertion, as all the computed weights were between 0 and 1. Demonstrating that this is the case for any accelerometer array and any set of accelerometer measurements remains an open question, however.

Nevertheless, the resulting expression of $\mathbf{w}_k$ minimizes the sum of the variances of the entries of the estimate $\widehat{\boldsymbol{\omega}}_k$ with respect to the true value $\boldsymbol{\omega}_k$, which is the goal of this chapter. Equation (2.48) is then substituted back in eq. (2.14), which yields the desired estimate $\widehat{\boldsymbol{\omega}}_k$ of $\boldsymbol{\omega}_k$.

2.2.3 Summary of the proposed method

The ensuing algorithm is summarized below. $\widehat{\boldsymbol{\omega}}_0$, $\widehat{\boldsymbol{\alpha}}_0$ and $\boldsymbol{\Sigma}_{\omega,0}$ are assumed to be known, and we set $\mathbf{w}_0 = \mathbf{1}_3$, i.e., we rely exclusively on the TI estimate for the first time step. Notice that the computation of $\mathbf{w}_k$ requires knowledge of the covariance matrix from the previous time step. Hence, this matrix should be updated in real time, which is done just after computing $\widehat{\boldsymbol{\omega}}_k$, through the Riccati equation appearing as eq. (2.14).

```
for $k = 1, ..., n$,
    compute $\widehat{\mathbf{z}}_k$ from eq. (2.28) and $\widehat{\mathbf{a}}_k$,
    compute $\widehat{\boldsymbol{\omega}}_{TI,k}$ from eq. (2.15),
    compute $\widehat{\boldsymbol{\omega}}_{PR,k}$ from the PR algorithm proposed in [2],
    compute $\widehat{\mathbf{Z}}_k^{-1}$ from eq. (2.40) and $\widehat{\boldsymbol{\xi}}_k$,
    compute $\widehat{\boldsymbol{\Psi}}_k$ from eq. (2.41) and $\widehat{\boldsymbol{\omega}}_{TI,k}$,
    compute $\mathbf{w}_k$ from eq. (2.48),
    compute $\widehat{\boldsymbol{\omega}}_k$ from eq. (2.14),
    compute $\widehat{\boldsymbol{\Sigma}}_{\omega,k}$ from eq. (2.44).
end.
```

2.3 Experimental validation

We validated the proposed method by performing an experiment that reproduces the conditions in which the algorithm would work in real life. While this type of validation has the merit of testing the algorithm close to its operating condition, it also presents the drawback of assessing its performance only in this one situation. An alternative validation approach would consist in testing the method over a wide spectrum of trajectories, either through simulation or experiments. This task was not achieved here, however, as the goal of this chapter is to demonstrate the usefulness of this method in its operating conditions.

2.3.1 Experimental testbed

In the experimental validation, six biaxial ADXL320 accelerometers [84] from *Analog Devices* are mounted on a rigid body to form an accelerometer array called the Octahedral Constellation of Twelve Accelerometers (OCTA), as shown in Fig. 2.5. We chose these low-end accelerometers to illustrate the robustness of our method. This OCTA was used first because it was available, having been designed by two former members in the robotics laboratory, in Université Laval, for testing the accelerometer-array. Therefore, we used this OCTA for estimating angular velocity firstly using this accelerometer-array. Secondly, OCTA is too large and heavy to be used as a game controller, in our case. We can use it, however, to test the effectiveness of the accelerometer-array. If it is feasible and acceptable, we can go further in the research. If not, we may need to find other ways for estimating the orientation. Moreover, OCTA can be used as comparison with the designed game controller, seen in Chapter 3 and 4, on estimating results.

A crucial aspect of these devices is their bias stability, as it directly impacts the drift rate of the TI estimate. In the case of the ADXL320, this bias stability is in the orders of 1 mg [84]. In comparison, the bias stability of accelerometers used in cruise missiles can be as low as 0.01 mg, whereas that of accelerometers in strategic missiles is below 1 μg [85]. Moreover, the noise density of the ADXL320 is rated at 250 m$g/\sqrt{\text{Hz}}$, whereas it is in the orders of 0.01 m$g/\sqrt{\text{Hz}}$ for military-grade accelerometers. A TrackSTAR tracking sensor from Ascension Technology Corporation [86], shown in Fig. 2.6, is also attached on OCTA, and will serve as a reference in the experiments. Its static angular accuracy is 0.5° RMS averaged over the translational range. Because the errors that are expected from the inertial sensor measurements are larger than this figure, this sensor is deemed sufficiently accurate for our needs. Displacements are produced by shaking OCTA by hand.

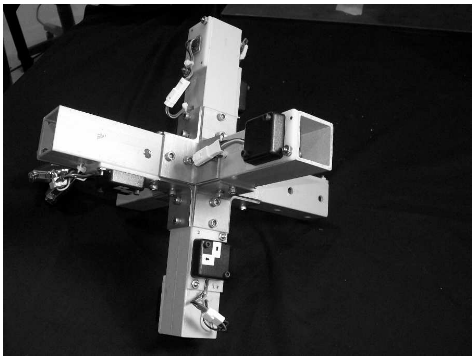


Figure 2.5: Photograph of OCTA

The geometry of OCTA is shown in Fig. 2.7. Its six pairs of accelerometers are located close

Figure 2.6: TrakSTAR from Ascension Technology Corporation

to the vertices of a regular octahedron. Each pair of accelerometers is directly screwed onto the 2 in × 2 in square tubing that composes the accelerometer array structure. Also they are attached with dowel pins in an effort to reduce misalignments between the sensors.

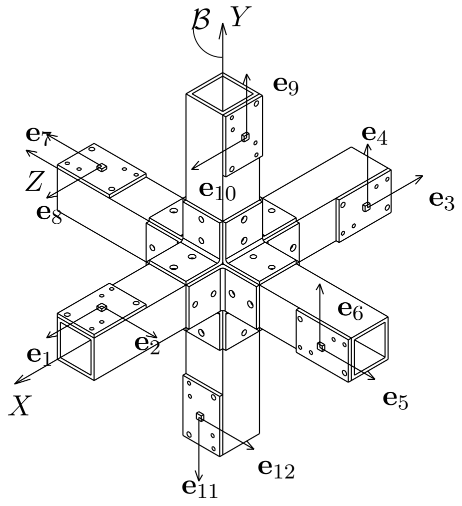


Figure 2.7: CAD drawing of OCTA

We have the nominal sensitive directions and positions

$$[\mathbf{e}_1 \quad \mathbf{e}_2 \cdots \mathbf{e}_{12}] = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & -1 \end{bmatrix},$$

and

$$\begin{bmatrix} \mathbf{r}_1^T \\ \mathbf{r}_2^T \\ \cdot \\ \cdot \\ \mathbf{r}_{12}^T \end{bmatrix} = \begin{bmatrix} 0.1970 & 0.0010 & 0.0060 \\ 0.1970 & 0.0010 & 0.0060 \\ -0.0450 & -0.0390 & -0.0300 \\ -0.0450 & -0.0390 & -0.0300 \\ 0.1120 & -0.0330 & -0.1210 \\ 0.1120 & -0.0330 & -0.1210 \\ 0.0670 & 0 & 0.1260 \\ 0.0670 & 0 & 0.1260 \\ 0.0730 & 0.0900 & -0.0280 \\ 0.0730 & 0.0900 & -0.0280 \\ 0.1120 & -0.1640 & 0.0040 \\ 0.1120 & -0.1640 & 0.0040 \end{bmatrix} \text{m},$$

Prior to this work, OCTA was calibrated [87], which allowed to identify the actual sensitive directions and positions of its accelerometers, as well as their biases and scale factors.

2.3.2 Preprocessing the measurements from OCTA and the TrackSTAR

The main objective is to estimate the accuracy of the angular-velocity estimates computed from the proposed method. In this vein, the angular-velocity measurements provided by the TrackSTAR are regarded as a reference.

First, the output voltages $v_k, k = 1, \ldots, n$, from the twelve accelerometers, are saved. From [87], we compute the corresponding accelerations

$$a_{i,k} = \mu_i v_{i,k} + \eta_i, \tag{2.49}$$

where $a_{i,k}$ is the acceleration of the accelerometer i along direction $\mathbf{e}_i$ at time t_k, μ_i and η_i are the scale factor and bias for this accelerometer, and $v_{i,k}$ is the voltage given by this accelerometer at t_k. Upon applying the algorithm proposed in Section 2.2.3, summarized in Fig. 2.8, the angular velocity $\widehat{\boldsymbol{\omega}}_k$ of OCTA is obtained.

On the other hand, from the TrackSTAR displacement sensor, the orientation of OCTA was measured every 0.01 s, for periods of 100 s. These orientations are returned in the form of rotation matrices $\mathbf{Q}_k$ that take the fixed frame onto a sensor frame. In order to compute the corresponding angular velocity $\boldsymbol{\omega}_k$, a first-order low-pass filter was first applied over the entries of $\mathbf{Q}_k$. Also, to avoid the time shifts caused by the filter, we filtered the data both in forward and backward time, and retained the average. After filtering the data, a central difference approximation of the first derivative was used to compute $\dot{\mathbf{Q}}_k$, namely,

$$\dot{\mathbf{Q}}_k = (\mathbf{Q}_{k+1} - \mathbf{Q}_{k-1})/(2\Delta t). \tag{2.50}$$

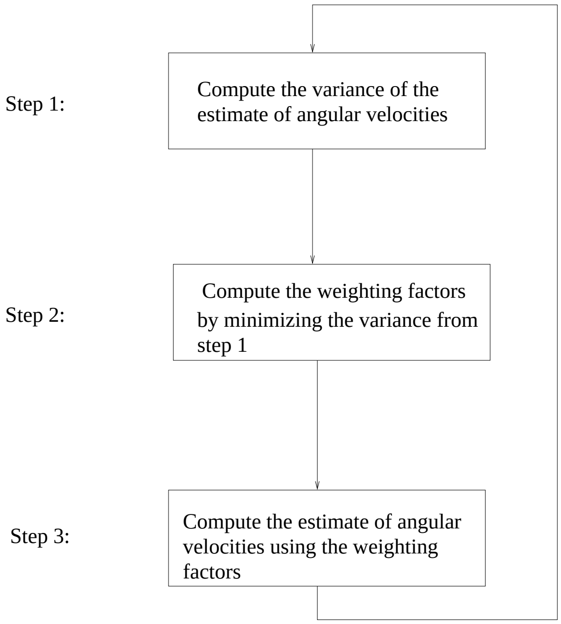

Step 1:
Compute the variance of the estimate of angular velocities
Step 2:
Compute the weighting factors by minimizing the variance from step 1
Step 3:
Compute the estimate of angular velocities using the weighting factors

Figure 2.8: The flowchart of the proposed method for estimating the angular velocities

The cross-product matrix $\mathbf{\Omega}_k$ of the angular velocity $\boldsymbol{\omega}_k$ is computed as (see [88])

$$\mathbf{\Omega}_k = \mathbf{Q}_k \dot{\mathbf{Q}}_k^T. \tag{2.51}$$

Because of measurement errors, this matrix is generally not skew-symmetric as it should be. We keep only the skew-symmetric component of $\mathbf{\Omega}_k$ by resorting instead to the relation

$$\mathbf{\Omega}_k = (\mathbf{Q}_k \dot{\mathbf{Q}}_k^T - (\mathbf{Q}_k \dot{\mathbf{Q}}_k^T)^T)/2. \tag{2.52}$$

2.3.3 Experimental results

OCTA was shaken by hand so as to produce arbitrary angular-velocity estimates. The range of angular velocities obtained by this method are thought to be representative of those obtained in many biomechanics experiments.

TI method [17]

The TI method [17], expressed in eq. (2.15), relies on angular-acceleration measurements. Because of the time-integration, any error in the angular acceleration estimate is accumulated, inevitably causing a drift of the estimate $\widehat{\boldsymbol{\omega}}_{TI}$ from its exact value. This problem has not prevented the TI method from being used successfully in biomechanics experiments lasting a fraction of a second, e.g. in crashworthiness and sports analysis. However, this method cannot be used alone for longer periods of time.

This can be verified on Fig. 2.9, which shows the x-component of the angular velocity estimates. These estimates were obtained from the TI method, the proposed method and the one by Parsa et al. [38]. Clearly, the TI estimates are completely useless after just a few instants, as they drift away from the TrackSTAR estimates.

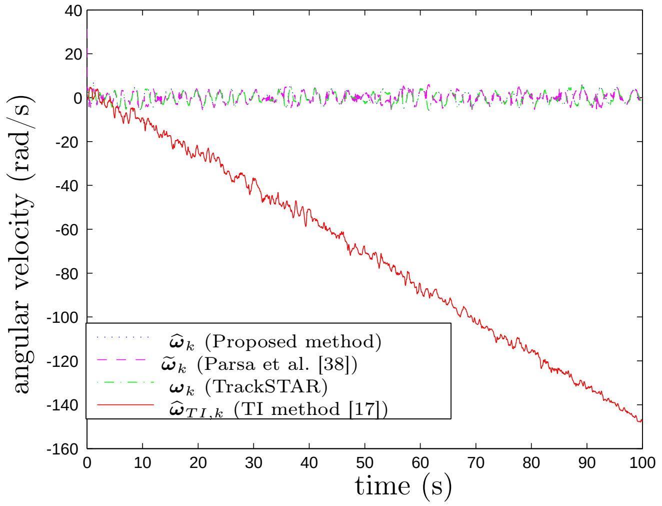


Figure 2.9: The first component of the angular-velocity estimates over 100 seconds

Method of Cardou et al. [2]

The PR method is best understood by its application to planar rigid-body motions. In this case, the quadratic form is simply represented by

$$\zeta \equiv \omega^2. \tag{2.53}$$

ζ is easily estimated, as it is linearly related to the accelerometer measurements. ζ is algebraically related to ω so that the PR method provides a stable estimate of the angular velocity:

$$\omega = \pm\sqrt{\zeta}. \tag{2.54}$$

Although this method is stable through time, notice that ζ only provides information regarding the absolute value of ω, none on its sign. Because of this sign ambiguity, it is generally not advisable to use the PR method alone. This can result in sign errors, as can be seen in Fig. 2.10, where the PR algorithm proposed in [2] picks up the wrong root between 17 s and 20 s.

Moreover, the square-root function has an infinite slope as $\zeta \to 0$, which generally results in a large error amplification whenever $\omega \to 0$. In fact, the slope is inversely proportional to the angular velocity itself, which seriously hinders the method in low angular-velocity applications.

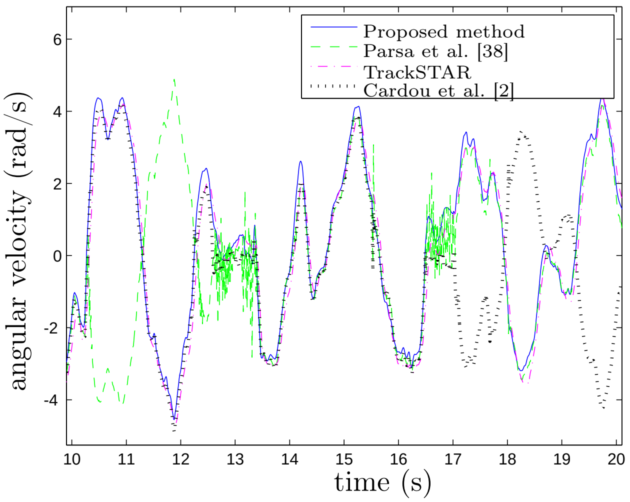


Figure 2.10: The first component of the angular-velocity estimates over ten seconds

From Fig. 2.10, the PR algorithm proposed in [2] does not seem to be affected by this problem over a part of the generated trajectories. As shown in [2], this specific PR method becomes singular only when all angular-velocity components are close to zero, whereas Fig. 2.10 only reports the behavior of the first component. This component approaching zero does not generally imply that the whole angular-velocity vector is null, which explains why the method seems unaffected by some zero crossings, while it encounters problems in others.

Method of Parsa et al. [38]

These drawbacks explain why a combination of the TI and PR methods should be done with care. In particular, the PR estimates should contribute less at low angular velocities, and more at high angular velocities. This is not what is done in the method by Parsa et al. [38], which is summarized in the Appendix A, for quick reference. In this method, the proportions of the TI and PR estimates in the final estimate are quite arbitrary.

This gives rise to important errors from the PR estimates at low angular velocities. This phenomenon can be observed on Fig. 2.10, which is a close-up view of the graph of Fig. 2.9. In Fig. 2.10, the estimates from the method of Parsa et al. are seen to oscillate when the angular velocity is close to zero. This undesired behavior, which was not observed even in the PR estimates of [2], is due to the method of combining the two angular-velocity estimates. This combination is done through the least-squares solution of an overdetermined linear system of equations, as shown in eq. (A.4). This linear system, however, becomes rank-deficient whenever any of the angular-velocity components goes to zero. This can be verified by computing

$$\det\left(\left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\omega}}\right)^T\left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\omega}}\right)\right) = 20\omega_x^2\omega_y^2\omega_z^2, \tag{2.55}$$

which is null whenever any of the angular-velocity components is zero. Notice that this is quite different from the PR method of [2], which only becomes singular when all angular-velocity components are zero. When approaching zero, the full rank of the linear system solved in eq. (A.4) is preserved, but it generally becomes ill-conditioned. Its solution is then very sensitive to measurement errors, which results in the oscillations of Fig. 2.10. Worse, this sometimes results in sign errors that last until the angular velocity reaches zero again.

In due fairness, we should mention, however, that this sign problem only occurs sporadically in the experiments. To illustrate this, we compute the Euclidean norm of the errors,

$$\delta\omega_k = \|\widehat{\boldsymbol{\omega}}_k - \boldsymbol{\omega}_k\|_2, \tag{2.56}$$

where $\widehat{\boldsymbol{\omega}}_k$ is the angular-velocity of OCTA and $\boldsymbol{\omega}_k$ is the angular-velocity obtained from the TrackSTAR. The resulting graphs are shown in Fig. 2.11. To these curves, we add $\sqrt{\mathrm{tr}(\boldsymbol{\Sigma}_{\omega,k})}$, the square root of the variance of $\boldsymbol{\omega}$ estimated through eq. (2.44).

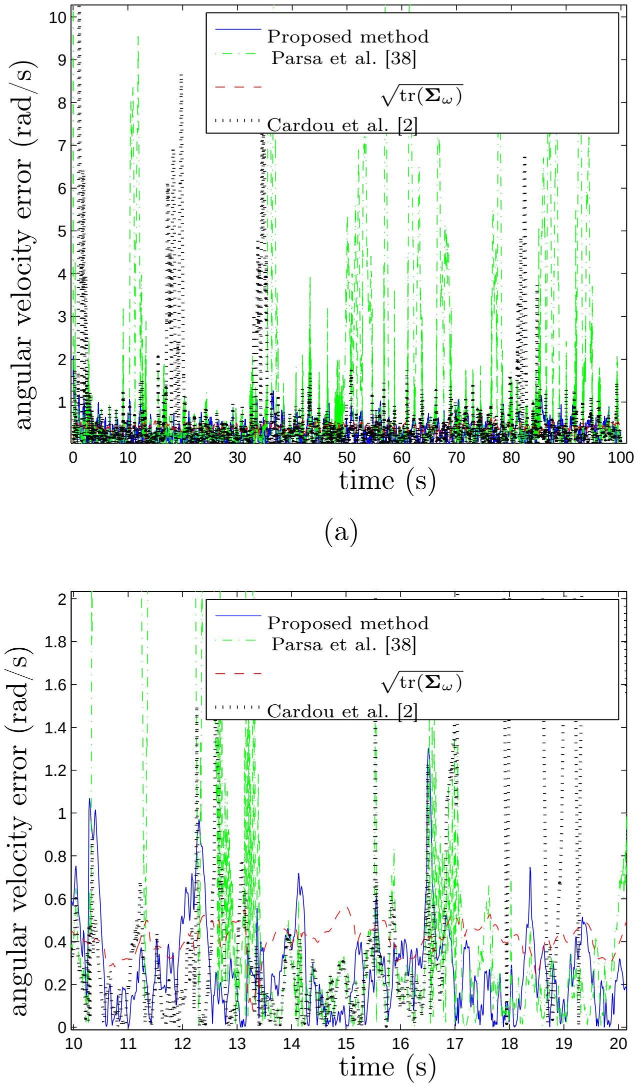
Apparently, from Fig. 2.11(a), the proposed method, that of Parsa et al. [38], and that of Cardou et al. [2] all provide stable results, as their error magnitudes do not grow over time. In this graph, the large peaks coming from the methods of [38] and [2] are explained by the sign problems described above. A closer look at the errors in Fig. 2.11(b) on the angular-velocity vectors reveals that the proposed method is slightly more robust than those of Parsa et al. and Cardou et al.. The time-scale of Fig. 2.11(b) corresponds to that of Fig. 2.10, which shows that the oscillations observed in the former correspond to the peak errors obtained in Fig. 2.11. On the other hand, the proposed method does not exhibit this undesired behavior, mainly because it reduces the contribution of the PR estimates at low angular velocities.

Additional experiments

In order to further validate the repeatability of the reported results, five other trajectories of 100 s each were performed. The corresponding RMS values of the errors of the proposed method and the one by Parsa et al. are reported in Table 2.1. These values were computed as

$$\delta\omega_{rms} = \sqrt{\frac{1}{n}\sum_{k=1}^{n}\|\widehat{\boldsymbol{\omega}}_k - \boldsymbol{\omega}_k\|_2^2}, \tag{2.57}$$

where $n = 10{,}000$ is the number of samples acquired in the experiment. The observed behavior was similar to that reported in Figs. 2.9, 2.10, 2.11, which is corroborated by the computed RMS values. Clearly, the proposed method is more robust than the existing ones.



(a)

(b)

Figure 2.11: Errors on the angular-velocity estimates

2.3.4 Analysis of the proposed method

The main novelty in this proposed method consists in weighting the TI and PR estimates according to a stochastic error model. This error model takes into account factors such as the acquisition frequency and the geometry of the accelerometer array.

Fig. 2.12 shows the time evolution of weighting factor $\mathbf{w}_k$, which reveals that $\mathbf{w}_k$ is closer to one than zero, and that it varies rapidly as the amplitude of $\boldsymbol{\omega}_k$ changes. Because $\mathbf{w}_k$ is generally close to one, from eq. (2.14), we conclude that $\widehat{\boldsymbol{\omega}}_{TI,k}$ predominates, while the small contribution from $\widehat{\boldsymbol{\omega}}_{PR,k}$ is enough to stabilize $\widehat{\boldsymbol{\omega}}_k$.

Table 2.1: RMS values of the errors

i	$\delta\omega_{rms}$(rad/s)		
	Proposed Method	Parsa et al. [38]	Cardou et al. [2]
1	0.4195	2.7579	1.3577
2	0.6350	2.3780	3.2347
3	0.6781	3.1036	2.9325
4	0.6456	4.3302	3.7612
5	0.6714	5.5671	5.1620
6	0.6274	4.8877	4.8051

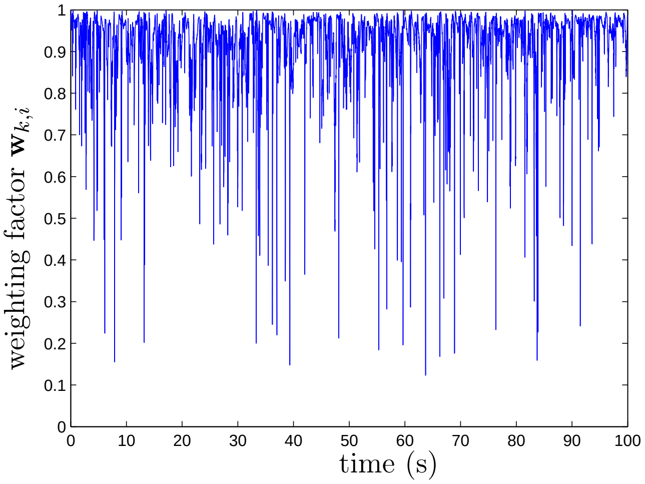


Figure 2.12: Weighting factor $\mathbf{w}$

In order to gain better insight on the relationship between $\mathbf{w}_k$ and $\widehat{\boldsymbol{\omega}}_k$, we trace all the three components of the weighting factors against the corresponding components of $\boldsymbol{\omega}_k$ in Fig. 2.13. Apparently, the weighting factor $\mathbf{w}_k$ remains closer to one under high angular velocities.

One criticism that may be addressed at the method proposed here is that it requires knowledge of σ_a^2, the variance of the accelerometer measurement errors. This parameter is not used by Parsa et al. [38] nor by Cardou et al. [2], for instance, which is certainly one of their advantages. It may be shown, however, that the method proposed here is independent from σ_a^2 under mild conditions. When the initial angular velocity of an experiment is known exactly, we choose $\mathbf{\Sigma}_{\omega,0} = \mathbf{0}_{3\times 3}$ and $\mathbf{w}_0 = \mathbf{1}_3$. In this case, one may verify that σ_a^2 cancels out in the Riccati eq. (2.44), so that $\mathbf{w}_k$ and $\widehat{\boldsymbol{\omega}}_k$ remain independent from this parameter.

In summary, the weighting factor $\mathbf{w}_k$ is used to balance the $\widehat{\boldsymbol{\omega}}_{PR}$ and $\widehat{\boldsymbol{\omega}}_{TI}$, to preserve the

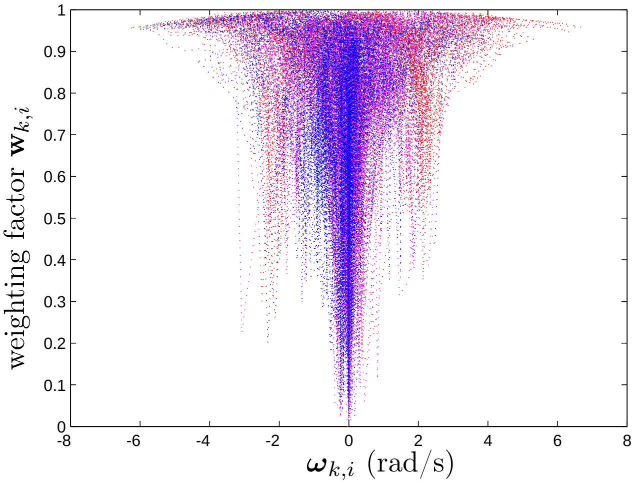


Figure 2.13: Relationship between the weighting factor $\mathbf{w}$ and angular velocity

advantages of both, while mitigating their defects. The relative accuracy of the method seems to improve at higher accelerations and higher angular velocities, as the proposed method tends to rely more on the PR estimates, which are stable.

2.4 Conclusion

Accelerometer arrays present some advantages over gyroscopes for estimating the angular velocity of a rigid body in space, such as better robustness under high accelerations. In this chapter, a new algorithm for the estimation of the angular velocity is proposed, based on a combination of the existing PR and TI methods. The main novelty of the proposed approach lies in the optimally weighted sum that is used to form the angular-velocity estimate. Through computation of the covariance of the angular velocity, a recursive form of the optimum weighting factor $\mathbf{w}_k$ is obtained.

The proposed approach is validated through experiments by manually shaking a custom-made accelerometer array called OCTA. The new method is shown to be stable, unlike the TI methods, and to behave well at low angular velocities, unlike the methods based on the polynomial roots of the angular-velocity terms. Furthermore, the proposed method consistently appears to be more robust than the methods avalilable in the literature. This was explained by the optimality of the proposed approach in terms of error propagation.

Meanwhile, we noted that the proposed method does not depend on the variance σ_a^2 of the accelerometer measurement errors whenever the initial angular velocity is known exactly. Hence, in this case, no prior knowledge of σ_a is required.

The proposed method will prove useful in applications where accelerometer arrays are already in use (e.g., crashworthiness and motion analysis in sports) by extending the period over which the angular velocity can be estimated accurately. It is hoped that the improved robustness will open new fields of application for accelerometer arrays such as human-robot collaboration and game controllers, for instance. In such applications, the error sensitivity of the accelerometer-based methods at low angular velocities could be mitigated by fusing the estimates with those of a low-cost gyroscope. In this fashion, one would rely more on gyroscope measurements over smooth trajectories, and more on accelerometers for quick, high angular-rate movements.

After obtaining the angular velocity, it is time to estimate the orientation. It is noted, however, that the angular velocity estimates we get in this chapter are not sufficiently accurate for a stable time integration to obtain the rigid-body orientation. Meanwhile, to avoid a two-step estimation method, we come to present a new way, in the next chapter, to estimate the orientation directly from inertial sensor measurements.

Chapter 3

Estimating the orientation in the vertical plane

La difficulté ou l'obscurité d'un sujet n'est pas une raison suffisante pour la négliger.
—Alexis Carrel

This chapter presents a novel method [89] for estimating the orientation of a rigid body in the vertical plane from point-acceleration measurements, by discerning its gravitational and inertial components. In this method, a simple stochastic model of the human-hand motions is used in order to distinguish between the two types of accelerations. Two mathematical models of the rigid-body motion are formulated as distinct state-space systems, each corresponding to a proposed method. In both cases, the output is a nonlinear function of the state, which calls for the application of the extended Kalman filter (EKF). The proposed filter is shown to work efficiently through two simulated trajectories, which are representative of human-hand motions. A comparison of the orientation estimates obtained from the proposed method shows that the filter offers more accuracy than a tilt sensor under high accelerations, and avoids the drift obtained by the time-integration of gyroscope measurements.

3.1 Accelerometer-array model

In order to estimate the orientation in 2D, a game controller similar to the Wii-mote is designed, where two biaxial accelerometers are installed at each of its ends. Fig. 3.1 represents the model of the accelerometer array mounted on a designed game controller, which is supposed to move in the vertical plane.

Similar to the accelerometer-array model described in Section 2.1, the measurement $\widehat{a}_{i,k}$ of the

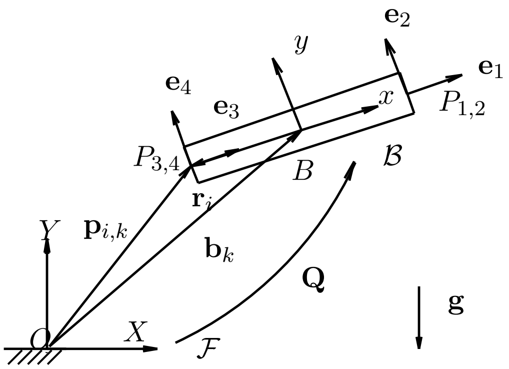


Figure 3.1: Sketch of the game controller with four accelerometers moving in a vertical plane

i^{th} accelerometer can be also modeled as

$$\widehat{a}_{i,k} = a_{i,k} + \delta a_{i,k}, \tag{3.1}$$

where

$$a_{i,k} = \mathbf{e}_i^T \ddot{\mathbf{p}}_{i,k} - \mathbf{e}_i^T \mathbf{Q}_k^T \mathbf{g} \tag{3.2}$$

is the true acceleration-component along the accelerometer sensitive direction, $\delta a_{i,k}$ is the measurement error, $\mathbf{Q}_k$ is the rotation matrix from the fixed frame $\mathcal{F}$ to the moving frame $\mathcal{B}$,

$$\mathbf{Q}_k = \begin{bmatrix} \cos\phi_k & -\sin\phi_k \\ \sin\phi_k & \cos\phi_k \end{bmatrix}, \tag{3.3}$$

where ϕ_k is the orientation in the vertical plane, and $\mathbf{g}$ is the vector of gravitational acceleration

$$\mathbf{g} = [0 \quad -g]^T. \tag{3.4}$$

The only difference is that it is in the vertical plane, where two biaxial accelerometers are needed, which indicates that $\widehat{\mathbf{a}}_k \in \mathbb{R}^4$.

As in eq. (2.8), the accelerometer measurement errors at the k^{th} time step are assumed to be zero-mean, identically, independently distributed Gaussian white noise $\delta\mathbf{a}_k$, i.e.,

$$\delta\mathbf{a}_k \sim \mathcal{N}(\mathbf{0}_m, \sigma_a^2 \mathbf{1}_{m\times m}). \tag{3.5}$$

Meanwhile, the position, velocity and acceleration of the accelerometer P_i in frame $\mathcal{B}$ are

$$\mathbf{p}_{i,k} = \mathbf{b}_k + \mathbf{r}_i, \tag{3.6}$$

$$\dot{\mathbf{p}}_{i,k} = \dot{\mathbf{b}}_k + \omega_k \mathbf{E}\mathbf{r}_i, \tag{3.7}$$

$$\ddot{\mathbf{p}}_{i,k} = \ddot{\mathbf{b}}_k + \dot{\omega}_k \mathbf{E}\mathbf{r}_i - \omega_k^2 \mathbf{r}_i, \tag{3.8}$$

where $\boldsymbol{\omega}_k$ is the rigid-body angular-velocity at time step k, and $\mathbf{E} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$. Because, for planar motion, we can write $\dot{\boldsymbol{\omega}}_k \times \mathbf{r}_i = r_i \dot{\omega}_k = r_i \alpha_k$ and $\boldsymbol{\omega}_k \times (\boldsymbol{\omega}_k \times \mathbf{r}_i) = r_i \omega_k^2$. Therefore, eq. (2.12) becomes

$$a_{i,k} = \mathbf{e}_i^T \left(\ddot{\mathbf{b}}_k - \mathbf{Q}_k^T \mathbf{g} \right) + \mathbf{e}_i^T (\mathbf{r}_i \dot{\omega}_k) + \mathbf{e}_i^T \mathbf{E} \mathbf{r}_i \omega_k^2. \tag{3.9}$$

Finally, eq. (2.13), $i = 1, \ldots, m$, simplifies to

$$\mathbf{A}\mathbf{z}_k = \mathbf{a}_k, \tag{3.10}$$

where $\mathbf{A} \equiv [\mathbf{A}_P \ \mathbf{A}_T \ \mathbf{A}_C] \in \mathbb{R}^{m\times 4}$, $\mathbf{A}_P \equiv [\mathbf{e}_1 \ \mathbf{e}_2 \ \mathbf{e}_3 \ \mathbf{e}_4]^T \in \mathbb{R}^{m\times 2}$, $\mathbf{A}_T \equiv [\mathbf{r}_1^T \mathbf{e}_1 \ \cdots \ \mathbf{r}_m^T \mathbf{e}_m]^T \in \mathbb{R}^{m\times 1}$ and $\mathbf{A}_C \equiv [\mathbf{r}_1^T \mathbf{E}^T \mathbf{e}_1 \ \cdots \ \mathbf{r}_m^T \mathbf{E}^T \mathbf{e}_m]^T \in \mathbb{R}^{m\times 1}$, $\mathbf{z}_k \equiv \left[(\ddot{\mathbf{b}}_k - \mathbf{g}^T \mathbf{Q}_k) \ \alpha_k \ \omega_k^2 \right]^T \in \mathbb{R}^4$, and $\mathbf{a}_k \in \mathbb{R}^m$. Furthermore, in the case of the proposed game controller, we have the nominal sensitive directions and positions

$$[\mathbf{e}_1 \ \mathbf{e}_2 \ \mathbf{e}_3 \ \mathbf{e}_4] = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}, \tag{3.11}$$

and

$$[\mathbf{r}_1 \ \mathbf{r}_2 \ \mathbf{r}_3 \ \mathbf{r}_4] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ r & r & -r & -r \end{bmatrix}. \tag{3.12}$$

Provided that the accelerometer array is fully determined [90], the application of eq. (3.10) to the contaminated measurements yields the least-squares estimate $\widehat{\mathbf{z}}$ of $\mathbf{z}$:

$$\begin{aligned} \widehat{\mathbf{z}}_k &= (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \widehat{\mathbf{a}}_k \\ &= \mathbf{A}^\dagger \widehat{\mathbf{a}}_k \\ &= \mathbf{A}^\dagger \mathbf{a}_k + \mathbf{A}^\dagger \delta \mathbf{a}_k \\ &= \mathbf{z}_k + \delta \mathbf{z}_k. \end{aligned} \tag{3.13}$$

Because the accelerometer measurements are assumed to be independent, as mentioned in eq. (3.5), we have

$$\delta \mathbf{z}_k \sim \mathcal{N}(\mathbf{0}_m, \sigma_a^2 (\mathbf{A}^T \mathbf{A})^{-1}). \tag{3.14}$$

Having defined the model for accelerometer-array planar motion, the main steps of the research can be outlined as follows:

a) Create various trajectories in Matlab for simulation.

b) Then, present some existing algorithms for estimating the orientation from multiple accelerometer or gyroscope measurements, and analyse the advantages and disadvantages.

c) Finally, propose a new method and have some tests in Matlab to justify and verify the algorithm proposed, compared with the existing methods mentioned.

3.2 Stochastic model of human-hand motion

In order to provide a minimum of additional information on the human-hand motion to the game controller, a coarse stochastic model is proposed for representing hand displacements. Let us define γ_k, the angular jerk [91] or the time rate of the angular acceleration α_k, and $\boldsymbol{\beta}_k$, the jerk or the time rate of the inertial acceleration $\ddot{\mathbf{b}}_k$, which is expressed in the fixed frame. We assume γ_k and $\boldsymbol{\beta}_k$ to be independently, identically distributed according to the Gaussian distribution, i.e.,

$$\gamma_k \sim \mathcal{N}(0, \sigma_\gamma^2), \tag{3.15}$$

$$\boldsymbol{\beta}_k \sim \mathcal{N}(\mathbf{0}_2, \sigma_\beta^2 \mathbf{1}_{2\times 2}). \tag{3.16}$$

Notice that this assumption is equivalent to stating that the angular and point accelerations undergo random walk, a.k.a. drunkard's walk. Indeed, we have

$$\alpha_{k+1} = \alpha_k + \Delta t \gamma_k, \tag{3.17}$$

$$\ddot{\mathbf{b}}_{k+1} = \ddot{\mathbf{b}}_k + \Delta t \boldsymbol{\beta}_k, \tag{3.18}$$

where Δt is the sample time. Through the 2^{nd} law of Newton, this model of the human-hand motions may be viewed as that by which the user applies moments and forces following random-walk variations on the game controller.

3.3 Algorithms for estimating the rigid-body orientation

As mentioned in the Chapter "Introduction", there are several existing methods for estimating the rigid-body orientation. In this section, we introduce the detailed existing basic algorithms, which are time-integration method and tilt-sensor method respectively, and our proposed methods, which are mean square error method and a Kalman filter method.

3.3.1 Time integration method

For time-integration method, as the name suggests, it is to integrate the angular velocity through time. Therefore, the orientation computed from this method may be written as

$$\widetilde{\phi}_k = \widetilde{\phi}_{k-1} + \frac{1}{2}(\omega_{k-1} + \omega_k)\Delta t, \tag{3.19}$$

where, $\widetilde{\phi}_k$ is the estimate of orientation from time-integration method, ω_k is the angular velocity and Δt is the sampling time.

3.3.2 Tilt-sensor method

For the tilt-sensor method, we assume that the orientation estimate can be computed

$$\overline{\phi}_k = \arctan\left(\frac{\ddot{b}_{x,k} + g\sin\phi_k}{\ddot{b}_{y,k} + g\cos\phi_k}\right). \tag{3.20}$$

where, the values of the entire expressions $\ddot{b}_{x,k} + g\sin\phi_k$ and $\ddot{b}_{y,k} + g\cos\phi_k$ can be computed from eq. (3.10).

Because of the drift from the time-integration method and the error caused by the inertial acceleration from the tilt-sensor method, it is not recommended to use any of these methods alone. These two methods are complementary, however, and an appropriate combination may yield robust estimates of ϕ. This is the idea behind the following proposed methods.

3.3.3 Method of the mean square error (MSE)

We propose to compute the orientation as a weighted average of the integration and tilt-sensor estimates, namely,

$$\widehat{\phi}_k = \rho_k\widetilde{\phi}_k + (1-\rho_k)\overline{\phi}_k = \rho_k\left(\widehat{\phi}_{k-1} + \frac{1}{2}(\omega_{k-1}+\omega_k)\Delta t\right) + (1-\rho_k)\arctan\left(\frac{\ddot{b}_{x,k} + g\sin\phi_k}{\ddot{b}_{y,k} + g\cos\phi_k}\right), \tag{3.21}$$

where ρ_k is the weighting factor that minimizes the MSE of the estimate $\widehat{\phi}_k$ with respect to the true value ϕ_k.

The MSE $\epsilon^2_{\phi,k}$ of $\widehat{\phi}_k$ may be written as

$$\epsilon^2_{\phi,k} \equiv E\left((\widehat{\phi}_k - \phi_k)^2\right) = E\left((\delta\phi_k)^2\right). \tag{3.22}$$

The goal is to choose the weighting factor ρ_k so as to minimize $\epsilon^2_{\phi,k}$ at time step k. To this end, let the derivative of the MSE be zero, i.e.,

$$\frac{d\left(\epsilon^2_{\phi,k}\right)}{d\rho_k} = 0, \tag{3.23}$$

and solve for the weighting factor ρ_k. Finally, substitute the resulting value of ρ_k in eq. (3.21) to obtain the estimate sought, $\widehat{\phi}_k$.

3.3.4 Method of the Kalman filter (KF)

The Kalman filter (KF) is an efficient recursive filter that estimates the internal state of a linear dynamic system from a series of noisy measurements [92]. The application of the KF requires writing the model of the accelerometer array as a state space system, namely

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{e}_k, \tag{3.24}$$

$$\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{n}_k, \tag{3.25}$$

where, $\mathbf{x}_k$ is the state vector, $\mathbf{y}_k$ is the measurement vector, $\mathbf{A}$ is the state matrix, $\mathbf{B}$ is the input matrix, $\mathbf{C}$ is the output matrix, $\mathbf{e}_k$ is the process noise, $\mathbf{n}_k$ is the measurement noise, and both are assumed to be zero-mean, identically, independently distributed Gaussian white noise,

$$\mathbf{e}_k \sim \mathcal{N}(\mathbf{0}_m, \mathbf{V}_k), \tag{3.26}$$

$$\mathbf{n}_k \sim \mathcal{N}(\mathbf{0}_m, \mathbf{R}_k). \tag{3.27}$$

In order to find the appropriate state space system, we define the time-integration of the angular velocity as the state equation and the inertial estimate as the output, namely

$$\phi_{k+1} = \phi_k + \frac{1}{2}\Delta t(\widehat{\omega}_{k+1} + \widehat{\omega}_k) + e_k, \tag{3.28}$$

$$\overline{\phi}_k = \phi_k + n_k, \tag{3.29}$$

where $\widehat{\omega}_k$ and $\widehat{\omega}_{k+1}$ are the input, $\overline{\phi}_k$ from eq. (3.20) is the output, e_k is the process noise whose covariance can be gained from the algorithm estimating the angular velocity $\widehat{\omega}_k$, and n_k is the measurement noise.

A knotty problem is that of obtaining the covariance of measurement noise n_k [93, 94, 95]. Therefore, different noise covariance matrices need to be tried through simulation before reaching a reasonable accuracy. As a result, the filter is very sensitive to variations in its parameters and may turn out unreliable in practice.

3.3.5 Method of the extended Kalman filter (EKF)

In this method, two mathematical models (orientation-only and complete pose) of the rigid-body motion are formulated as distinct state-space systems, each corresponding to a proposed method. In both cases, the output is a nonlinear function of the state, which calls for the application of the extended Kalman filter (EKF).

Let us begin with the case where we only estimate the orientation. We first write a state-space system, where the time-integration of the angular velocity is the state equation and the acceleration from the measurements is the output equation

$$\phi_{k+1} = \phi_k + \frac{1}{2}\Delta t(\widehat{\omega}_{k+1} + \widehat{\omega}_k) + e_k, \tag{3.30}$$

$$\ddot{\mathbf{c}}_k = g \begin{bmatrix} \sin\phi_k \\ \cos\phi_k \end{bmatrix} + \ddot{\mathbf{b}}_k + \delta\ddot{\mathbf{c}}_k, \tag{3.31}$$

where ϕ_k is the state variable, $\widehat{\omega}_k$ is an input measured here by a gyroscope, g is the gravitational acceleration, $\ddot{\mathbf{c}}_k$ is the combined inertial and gravitational acceleration which is computed directly from eq. (3.10), and e_k and $\delta\ddot{\mathbf{c}}_k$ are the process and measurement noise, respectively.

The state-space system of eq. (3.30) and eq. (3.31) cannot be solved by the regular KF, since the output equation is nonlinear in the system state. The EKF consists in using the first-order Taylor expansion of the nonlinear functions to linearize the state and output equations about the current mean and covariance.

In eq. (3.31), $\ddot{\mathbf{c}}_k$ is computed directly from eq. (3.10), but the inertial acceleration $\ddot{\mathbf{b}}_k$ is unknown. However, it is possible to consider it as the additional noise whose covariance could be easily known. Although we have no deterministic way of predicting the path taken by the rigid body, we may estimate, however coarsely, the probability of moving from one pose to the next within one time step. This probability may be assessed by taking into account that the rigid body, in our application, is being acted upon by a human. In particular, we may attempt to quantify the likely accelerations of the human hand in given applications. This additional noise $\ddot{\mathbf{b}}_k$ added with the initial measurement noise $\delta\ddot{\mathbf{c}}_k$ will be considered as the new measurement noise in the proposed state-space system.

Therefore, the covariance matrices of the measurement and process noise, respectively, $\mathbf{R}_k$ and v_k can be obtained as follows. From eq. (3.14), we have the covariance matrix

$$E(\delta\mathbf{z}_k\delta\mathbf{z}_k^T) = (\mathbf{A}^T\mathbf{A})^{-1}\sigma_a^2, \tag{3.32}$$

where $\delta\mathbf{z}_k = \begin{bmatrix}\delta\ddot{\mathbf{c}}_k^T \ \delta\alpha_k \ \delta\omega_k^2\end{bmatrix}^T \in \mathbb{R}^4$. Therefore, $E(\delta\ddot{\mathbf{c}}_k\delta\ddot{\mathbf{c}}_k^T)$ can be obtained from eq. (3.32). Finally, the covariance matrix of the measurement noise becomes

$$\mathbf{R}_k = E(\ddot{\mathbf{b}}_k\ddot{\mathbf{b}}_k^T) + E(\delta\ddot{\mathbf{c}}_k\delta\ddot{\mathbf{c}}_k^T). \tag{3.33}$$

The covariance matrix v_k of the process noise is obtained from the gyroscope properties, which yields

$$v_k = \Delta t^2\sigma_\omega^2, \tag{3.34}$$

where σ_ω^2 is the covariance of the error of the angular velocity estimate in the case of a planar motion.

Finally, the orientation estimate can be obtained via a classical EKF. Additionally, the application of an Extended Kalman Filter (EKF) requires computing the Jacobian matrix of the output function, namely,

$$\mathbf{H}(\mathbf{x}) = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\phi_k} = g \begin{bmatrix} \cos\phi_k \\ \text{-}\sin\phi_k \end{bmatrix}. \tag{3.35}$$

Let us now turn our attention towards the second proposed method, by which we estimate the complete rigid-body pose in the vertical plane. To this end, the rigid body is yet again instrumented with m accelerometers whose outputs at time step k form the vector $\mathbf{y}_k \in \mathbb{R}^m$. These measurements are related to five parameters of the rigid-body motion, namely, the acceleration of a reference point $\mathbf{a}$, the orientation ϕ with respect to gravity, the angular acceleration α and the angular velocity squared ω^2. Therefore, the state-space system is

$$\mathbf{x}_{k+1} = \mathbf{F}\mathbf{x}_k + \mathbf{G}\mathbf{u}_k, \tag{3.36}$$

$$\widehat{\mathbf{y}}_k = \mathbf{h}(\mathbf{x}_k) + \delta\mathbf{y}_k, \tag{3.37}$$

where $\mathbf{x}_k \equiv \left[\phi_k \; \omega_k \; \alpha_k \; \mathbf{p}_k \; \mathbf{v}_k \; \ddot{\mathbf{b}}_k\right]^T$, $\mathbf{p}_k$ and $\mathbf{v}_k$ are the position and velocity of the reference point, $\mathbf{u}_k \equiv [\gamma_k \; \boldsymbol{\beta}_k]^T$ (see eq. (3.15) and eq. (3.16)) is the system input and is assumed to be Gaussian white noise, namely

$$\mathbf{u}_k \sim \mathcal{N}(\mathbf{0}_3, \boldsymbol{\Sigma}_u^2), \tag{3.38}$$

which implies a covariance matrix of the form

$$\boldsymbol{\Sigma}_{\mathbf{u}}^2 = \begin{bmatrix} \sigma_\gamma^2 & \mathbf{0}_2^T \\ \mathbf{0}_2 & \sigma_\beta^2 \mathbf{1}_{2\times 2} \end{bmatrix}. \tag{3.39}$$

We also have

$$\mathbf{F} = \begin{bmatrix} 1 & \Delta t & (1/2)\Delta t^2 & \mathbf{0}_2^T & \mathbf{0}_2^T & \mathbf{0}_2^T \\ 0 & 1 & \Delta t & \mathbf{0}_2^T & \mathbf{0}_2^T & \mathbf{0}_2^T \\ 0 & 0 & 1 & \mathbf{0}_2^T & \mathbf{0}_2^T & \mathbf{0}_2^T \\ \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{1}_{2\times 2} & \Delta t \mathbf{1}_{2\times 2} & (1/2)\Delta t^2 \mathbf{1}_{2\times 2} \\ \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{0}_{2\times 2} & \mathbf{1}_{2\times 2} & \Delta t \mathbf{1}_{2\times 2} \\ \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{0}_2 & \mathbf{0}_{2\times 2} & \mathbf{0}_{2\times 2} & \mathbf{1}_{2\times 2} \end{bmatrix}, \tag{3.40}$$

$$
\mathbf{G} = \begin{bmatrix} (1/6)\Delta t^3 & 0 & 0 \\ (1/2)\Delta t^2 & 0 & 0 \\ \Delta t & 0 & 0 \\ \mathbf{0}_2 & \mathbf{0}_2 & (1/6)\Delta t^3 \mathbf{1}_{2\times 2} \\ \mathbf{0}_2 & \mathbf{0}_2 & (1/2)\Delta t^2 \mathbf{1}_{2\times 2} \\ \mathbf{0}_2 & \mathbf{0}_2 & \Delta t \mathbf{1}_{2\times 2} \end{bmatrix}, \tag{3.41}
$$

$$
\mathbf{h}(\mathbf{x}_k) = \mathbf{A}_P \mathbf{Q}_k^T \ddot{\mathbf{b}}_k - \mathbf{A}_P \mathbf{g} \cos\phi_k + \mathbf{A}_P \mathbf{E} \mathbf{g} \sin\phi_k + \mathbf{A}_T \alpha_k + \mathbf{A}_C \omega_k^2 \tag{3.42}
$$

where $\mathbf{A}_P$, $\mathbf{A}_T$ and $\mathbf{A}_C$ from eq. (3.10) are constant coefficients, $\mathbf{g} \in \mathbb{R}^2$ is the gravity vector, and $\ddot{\mathbf{b}}_k$ is the inertial acceleration in the fixed frame.

The covariance $\mathbf{V}_k$ of the process error and the covariance $\mathbf{R}_k$ of the measurement error are, respectively,

$$
\mathbf{V}_k = \mathbf{G} \boldsymbol{\Sigma}_u^2 \mathbf{G}^T, \tag{3.43}
$$

$$
\mathbf{R}_k = \sigma_a^2 (\mathbf{A}^T \mathbf{A})^{-1}. \tag{3.44}
$$

The extended Kalman filter requires that we compute the output-function Jacobian matrix as well, namely,

$$
\begin{aligned}
\mathbf{H}(\mathbf{x}) &= \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}_k} \\
&= \begin{bmatrix} \mathbf{A}_P \boldsymbol{\Omega}_k^T \ddot{\mathbf{b}}_k + \mathbf{A}_P \mathbf{g} \sin\,\phi_k + \mathbf{A}_P \mathbf{E} \mathbf{g} \cos\phi_k & 2\mathbf{A}_C \omega_k & \mathbf{A}_T & \mathbf{0}_{m\times 2} & \mathbf{0}_{m\times 2} & \mathbf{A}_P \end{bmatrix}.
\end{aligned} \tag{3.45}
$$

where $\boldsymbol{\Omega}_k^T = \begin{bmatrix} -\sin\phi_k & \cos\phi_k \\ -\cos\phi_k & -\sin\phi_k \end{bmatrix}$ is the derivative of rotation matrix $\mathbf{Q}_k$ with respect to the orientation ϕ_k. Finally, using the method of the EKF, the state vector $\mathbf{x}_k$ can be estimated.

3.3.6 Method of the unscented Kalman filter (UKF)

Generally, the extended Kalman filter (EKF) is popular, as it is quite simple to use the first-order Taylor expansion of the nonlinear functions for linearizing about the current mean and covariance. However, this introduces errors in the a posteriori mean and covariance and thus could make the estimate diverge if the process were modeled incorrectly or if the initial estimate were unreliable. For this reason, we decided to resort instead to the unscented Kalman filter (UKF) for the state-space system from eq. (3.30) and eq. (3.31).

At the core of the UKF is the Unscented Transformation (UT), by which one calculates the nonlinear information of a random variable [96, 97, 98, 99]. This approximation uses a set of sample points called sigma points (SP), which represent the mean and covariance of the random variables. Therefore, the first step is to determine these sigma points, the second step is to apply the UT for each SP, and the last step is to compute the mean and covariance from the transformed SP.

Let us assume a nonlinear discrete state-space system

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k, \mathbf{u}_k) + \mathbf{e}_k, \tag{3.46}$$

$$\mathbf{y}_k = h(\mathbf{x}_k) + \mathbf{n}_k, \tag{3.47}$$

where $\mathbf{x}_k$ is the system state vector, $\mathbf{u}_k$ is the input vector, $\mathbf{y}_k$ is the system output or measurement vector, and $\mathbf{e}_k$ and $\mathbf{n}_k$ are process noise and measurement noise respectively.

Assume that $\overline{\mathbf{x}}_k$ and $\mathbf{P}_k$ are the mean and the error covariance matrices of state $\mathbf{x}$, and that $\widehat{\mathbf{x}}_k$ is the state estimate. The SP $\mathcal{X}_k$ is defined as follows:

$$\begin{aligned} &\mathcal{X}_{0,k} = \overline{\mathbf{x}}_k, \\ &\mathcal{X}_{i,k} = \overline{\mathbf{x}}_k + \left(\sqrt{(L+\lambda)\mathbf{P}_k}\right)_i, \quad i = 1, 2, \ldots, L \\ &\mathcal{X}_{i,k} = \overline{\mathbf{x}}_k - \left(\sqrt{(L+\lambda)\mathbf{P}_k}\right)_{i-L}, \quad i = L+1, L+2, \ldots, 2L \end{aligned} \tag{3.48}$$

where L is the dimension of vector $\mathbf{x}_k$, $\lambda = \widetilde{\alpha}^2(L+\kappa) - L$ is a scaling parameter, $\widetilde{\alpha}$ determines the spread of the PS around $\overline{\mathbf{x}}$ and is usually given a very small positive value between 0.0001 and 1, $\kappa = 3 - L$ is a secondary scaling parameter and $\left(\sqrt{(L+\lambda)\mathbf{P}_k}\right)_i$ is the ith column of the matrix square root.

For the covariance matrices of process and measurement noise, they are the same as eq. (3.33) and eq. (3.34). As a matter of fact, when the state transition and observation models—that is, the predict and update functions f and h—are highly non-linear, the extended Kalman filter can give particularly poor performance. This is because the covariance is propagated through linearization of the underlying non-linear model. However, in our case, from eq. (3.31), it is evident that the observation model is not highly non-linear, which could indicate that the EKF is as accurate as UKF.

3.4 Validation through simulation

3.4.1 Simulated game controller

In order to estimate the orientation in the plane, a game controller similar to the Wii-mote is designed. In this version, two biaxial accelerometers are assumed to be at each of its ends, forming an array of 4 accelerometers, and a gyroscope is at its midpoint. Fig. 3.2 shows the

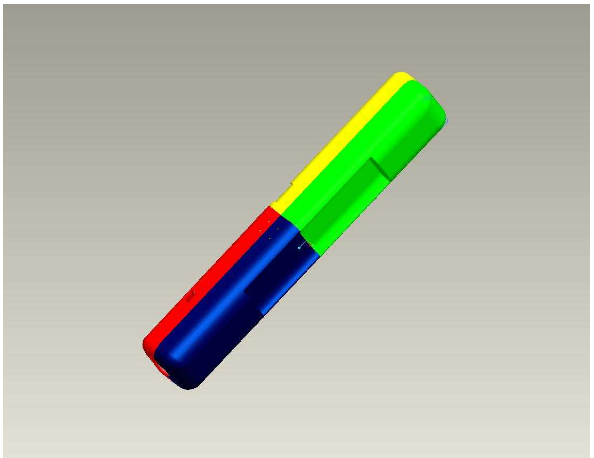


Figure 3.2: ProE drawing of game controlller

ProE rendering of the designed game controller. Let us introduce the details of this game controller.

The length of the game controller is $2r = 0.3$ m, where r is the half-length dimension appearing in matrix $\mathbf{A}$ in eq. (3.10). For the simulation, all the geometric parameters used are the same as those of this game controller. The sampling period it uses is $\Delta t = 0.01$ s. Some other simulation parameters are gravitational acceleration $g = 9.81$ m/s^2, the standard deviation of position $\sigma_b = 0.3$ m, the standard deviation of accelerometer noise $\sigma_a = 0.3$ m/s^2, the standard deviation of angular velocity from gyroscope $\sigma_\omega = 0.6$ rad/s, where the standard deviation values are chosen to make sure that they are less than 5% of the virtual sensors' scales in simulation, the initial state (orientation) $\phi_0 = 0^o$ and its initial estimate error covariance $\mathbf{P}_0 = 10^3$.

3.4.2 Simulated random walk of the trajectory

In order to generate arbitrary trajectory of hand motions, we first use a random-walk model on its pose, to which we applied a first-order low-pass filter, so as to smoothen the trajectory.

For the random walk motion, we have

$$\mathbf{b}_k = \mathbf{b}_{k-1} + \Delta\mathbf{b}_k, \tag{3.49}$$

$$\phi_k = \phi_{k-1} + \Delta\phi_k, \tag{3.50}$$

where $\mathbf{b}_k = [x \quad y]^T$ represents the position, ϕ_k represents the orientation and $\Delta\mathbf{b}_k$ and $\Delta\phi_k$ are assumed to be zero-mean, identically, independently distributed Gaussian white noise, namely,

$$\Delta\mathbf{b}_k \sim \mathcal{N}(\mathbf{0}_2, \sigma_b^2 \mathbf{1}_{2\times 2}), \tag{3.51}$$

$$\Delta\phi_k \sim \mathcal{N}(0, \sigma_\phi^2). \tag{3.52}$$

In order to better model the dynamic capabilities of the hand, we add a filter $G(s) = 1/(\tau s+1)$ to the noise in eq. (3.50), where $\tau = 1$ s is the associated time constant. At first sight, this time constant may appear larger than that of a typical human-hand movement. This value of τ was adjusted to produce accelerations in the range 3g, approximately, which is similar to those of the human hand. The resulting accelerations are traced over time in Fig. 3.4. It is evident that this acceleration is not obtained from the Newton's second law of motion, but rather from eq. (3.49). Hence, the value of τ was chosen a posteriori, from the generated accelerations. As the filter is expressed in frequency-domain, we need to convert equations (3.49) and (3.50) from time-domain to frequency-domain. Thus, from eq. (3.50), we have

$$\dot{\phi}_k = \frac{\phi_k - \phi_{k-1}}{\Delta t} = \frac{1}{\Delta t}\Delta\phi_k. \tag{3.53}$$

Then, we convert eq. (3.53) to a continuous-time expression, which yields, after a Laplace transformation,

$$s\Phi(s) = \frac{1}{\Delta t}\Delta\phi. \tag{3.54}$$

After this, we combine the first-order filter and the random-walk model, to obtain

$$\Phi(s) = \frac{1}{\tau s+1}\frac{1}{s\Delta t}\Delta\phi = \frac{1}{\tau s^2 + s}\frac{\Delta\phi}{\Delta t}. \tag{3.55}$$

Reverting back to the discrete-time domain, we obtain

$$\phi_k = \left(\frac{\Delta t}{\tau} - 1\right)\phi_{k-2} + \left(2 - \frac{\Delta t}{\tau}\right)\phi_{k-1} + \frac{\Delta t}{\tau}\Delta\phi_k. \tag{3.56}$$

A similar development yields the recursive law of the trajectory of point B, namely,

$$\mathbf{b}_k = \left(\frac{\Delta t}{\tau} - 1\right)\mathbf{b}_{k-2} + \left(2 - \frac{\Delta t}{\tau}\right)\mathbf{b}_{k-1} + \frac{\Delta t}{\tau}\Delta\mathbf{b}_k. \tag{3.57}$$

A typical trajectory resulting from this random walk model is shown in Fig. 3.3.

In order to verify the algorithms of EKF and UKF mentioned above in Section 3.3, we need to compute the covariance of inertial acceleration for the random walk model. From eq. (3.57), we

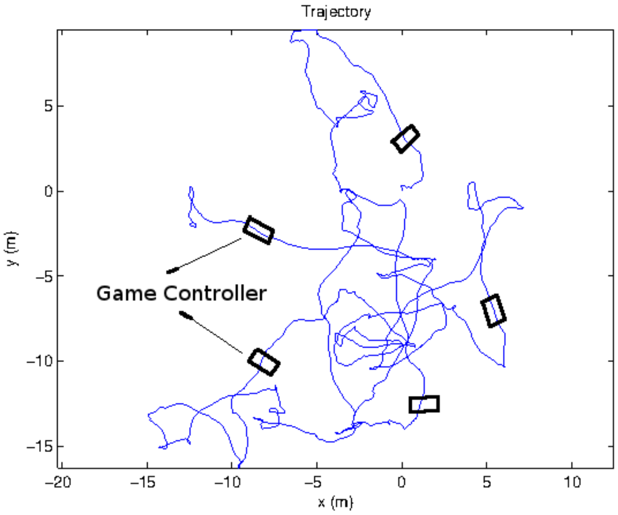


Figure 3.3: Trajectory of random walk model

have its velocity expression $\dot{\mathbf{b}}_k = (\mathbf{b}_k - \mathbf{b}_{k-1})/\Delta t$ and its variance $E(\dot{\mathbf{b}}_k \dot{\mathbf{b}}_k^T)$. Then the inertial acceleration in the fixed frame is $\ddot{\mathbf{b}}_k = (\dot{\mathbf{b}}_k - \dot{\mathbf{b}}_{k-1})/\Delta t$, and finally, we have its covariance,

$$E(\ddot{\mathbf{b}}_k \ddot{\mathbf{b}}_k^T) = \frac{3 - \Delta t/\tau}{\tau^2 \Delta t^2 (2 - \Delta t/\tau)} \sigma_b^2 \mathbf{1}_{2\times 2}. \tag{3.58}$$

One component of the true and error-stained simulated accelerometer measurements resulting from this simulation are shown in Fig. 3.4. As a result, we present the final comparison of the orientation estimates described here in Fig. 3.5, and their corresponding errors appear in Fig. 3.6.

In these two figures, it is shown that the time-integration estimate drifts over time, by as much as 50^o over 100 s. Clearly, the estimate from the tilt sensor is not reliable over the simulated range of inertial accelerations. This is explained from eq. (3.20), which is exact only when $\ddot{\mathbf{b}}_k = \mathbf{0}_2$. When, on the contrary, $\ddot{\mathbf{b}}_k$ is far from being null, the error on $\overline{\phi}_k$ is correspondingly large. For the estimate from the EKF for orientation-only method, provided that the covariance of noises is chosen properly, we observe that the result matches well the true orientation. However, the estimate from the EKF for complete-pose method has an error that is somewhat larger.

3.4.3 Simulated ∞-shaped trajectory

In order to test the proposed methods, another regular trajectory model having the shape of the ∞ symbol is used in simulation. To follow such a trajectory, we use a Lissajous curve,

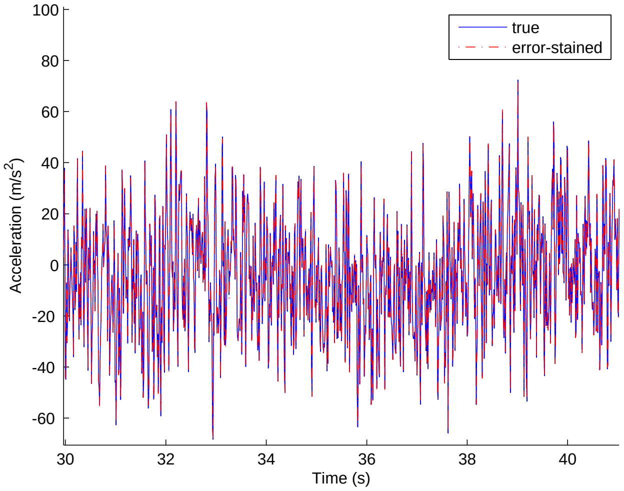


Figure 3.4: The true and error-stained accelerometer measurements

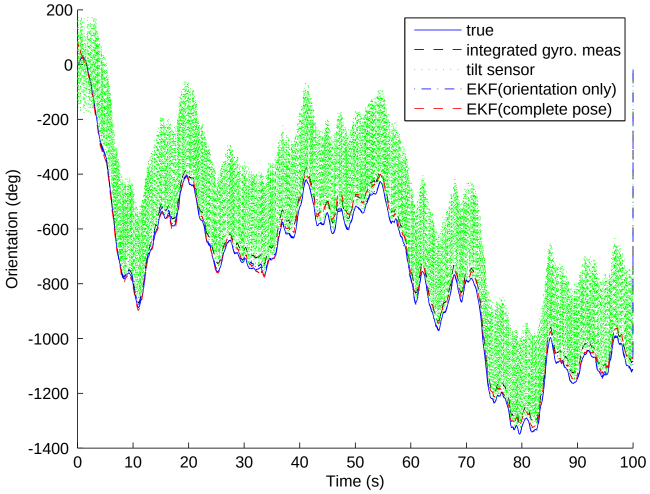


Figure 3.5: The proposed and already-existing orientation estimates

namely,

$$x = a\sin(ct + \delta), \tag{3.59}$$

$$y = b\sin(dt). \tag{3.60}$$

where, $a = 0.5$ m, $b = 0.25$ m, $c = 2\pi/\tau$, $\tau = 2$ s, $d = 2c$, $\delta = 0$, t is the time and $\dot{x}$ and $\dot{y}$ are time derivative of x and y respectively. Moreover the game-controller is to remain parallel to the velocity vector of point B, that is,

$$\phi = \arctan(\dot{y}/\dot{x}). \tag{3.61}$$

The corresponding trajectory is shown in Fig. 3.7.

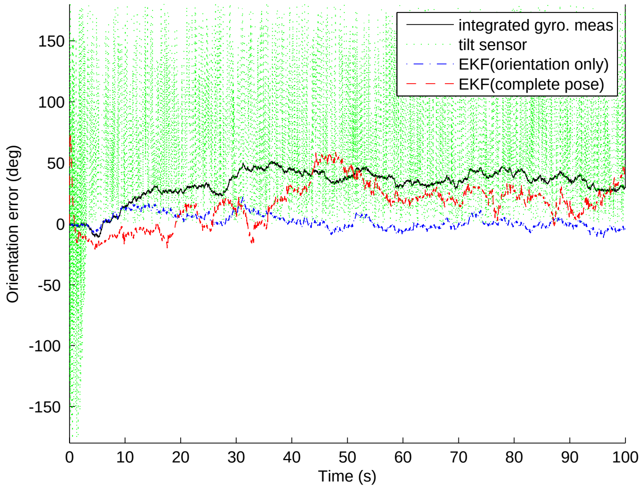


Figure 3.6: Errors of the proposed and existing orientation estimates

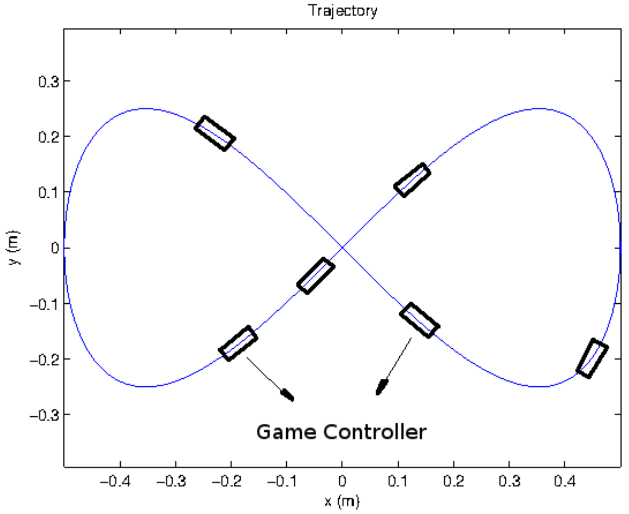


Figure 3.7: Trajectory of ∞-shaped model

As for the simulated ∞-shaped trajectory, Fig. 3.8 presents the accelerometer measurements, and Fig. 3.9 and Fig. 3.10 present the orientation estimates and their errors. It is seen that the time-integration method and the tilt-sensor method exhibit the same problems as before, namely, drift and sensitivity to inertial accelerations. Similarly, the orientation estimate from the complete-pose method again has more error than the one from the orientation-only method. The cause of this phenomenon is left to be explained in the discussion.

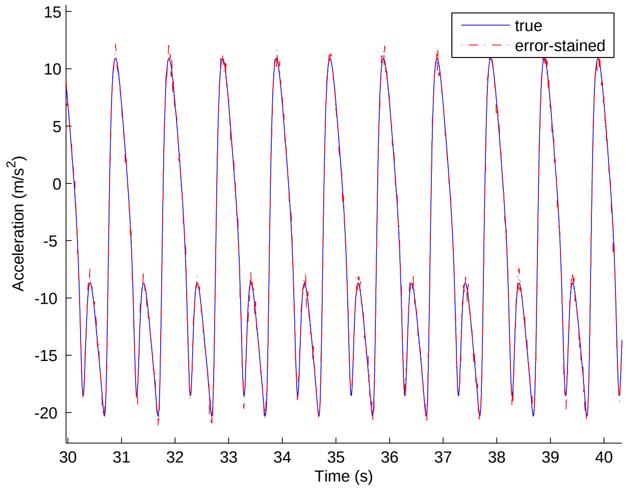


Figure 3.8: The true and error-stained accelerometer measurements

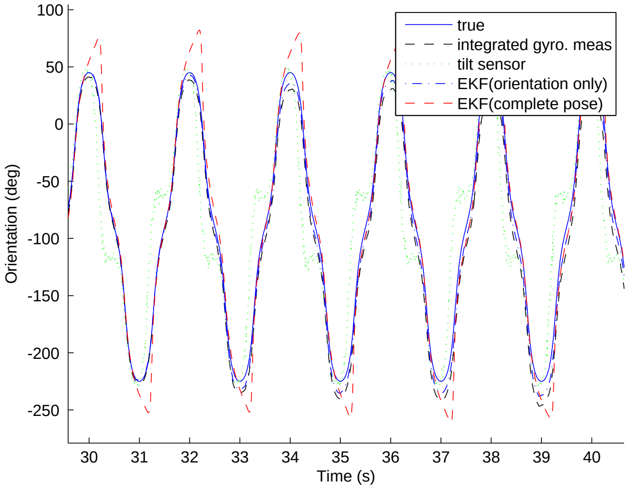


Figure 3.9: Comparison of orientation

3.5 Discussion

The time-integration method consists in integrating the angular velocity measured by a gyroscope over time, thus accumulating errors in the process and, inevitably, causing a drift. This prevents anyone from using this method alone for long periods of time. However, it may prove useful in some applications where estimates are required over very short time periods. It is evident in Fig. 3.6 and Fig. 3.10 that during the first 5 seconds, the errors are small, while they are seen to steadily increase in the following seconds.

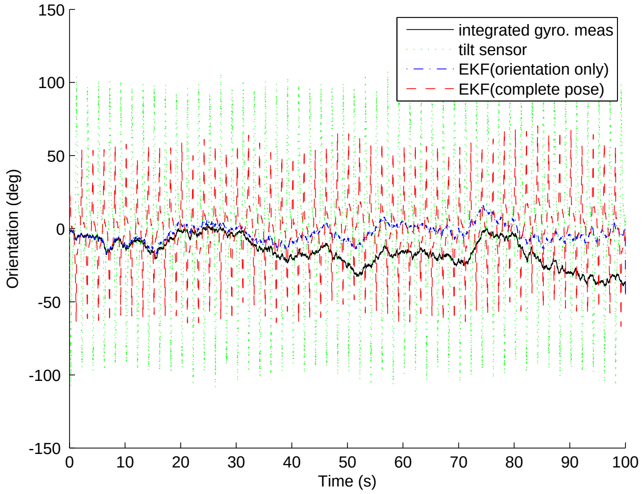


Figure 3.10: Errors of orientation

With the help of gravity, accelerometers are usually taken as tilt sensors for estimating the roll and pitch angles when the rigid body is immobile. For the game controller moving in the vertical plane, the estimate from this gravitational method, is severely affected by the inertial acceleration (see Fig. 3.6 and Fig. 3.10). Furthermore, it oscillates constantly between different values shown in Fig. 3.5, which can be explained by the argument of $arctan$ in eq. (3.20), only when both $(\ddot{b}_{y,k} + g\cos\phi_k) \rightarrow 0$ and $(\ddot{b}_{x,k} + g\sin\phi_k) \rightarrow 0$.

Meanwhile, In order to keep the clarity of Figures 3.5 and 3.9, we implement and analyse the MSE and KF methods in a different trajectory. Fig. 3.11 presents a new game-controller trajectory, and Fig. 3.12 shows the methods comparison. Fig. 3.12 indicates that, the final estimates of the MSE method are so close to those of the tilt-sensor method. In principle, it is impossible to distinguish the inertial acceleration and the gravitational one, and only their sum can be taken into the calculation. In the event that the inertial acceleration is small enough (smaller than $1g$), the accelerometer may be considered as a tilt sensor, and the MSE estimates as well as the gravitational estimates then become satisfactory.

As mentioned in the Section 3.3, the KF method proposed is a flawed method. The attempt to combine the time-integration and tilt-sensor estimates via a linear Kalman filter proved unsuccessful, as can be seen from Fig. 3.12. In this figure, the error of the KF method reaches 200°. This may be explained by the inability of the standard Kalman filter to capture the nonlinear relationship between the angle ϕ and the gravitational acceleration. This result spurred the application of nonlinear variants of the Kalman filter to our problem. Let us now turn our attention to these methods.

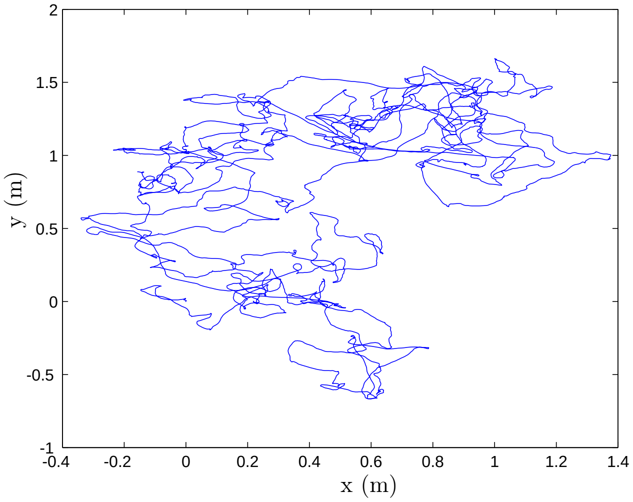
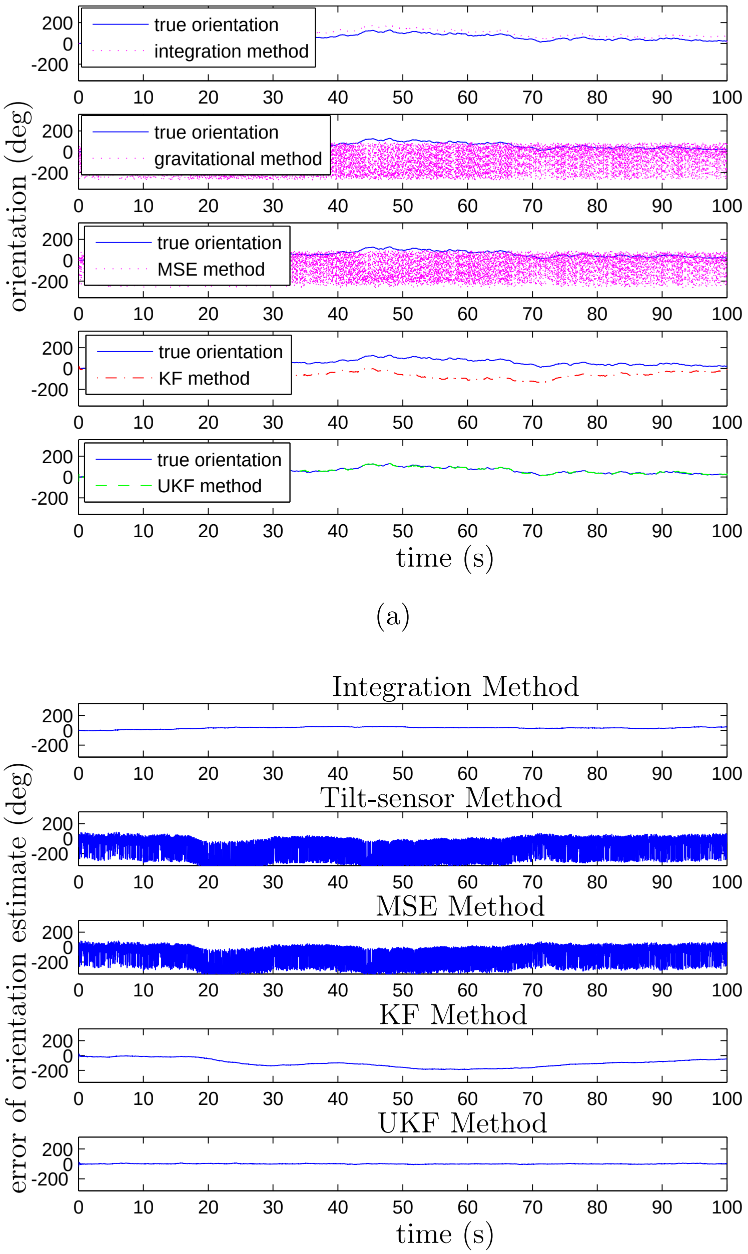


Figure 3.11: Game controller and remote trajectory in the plane

For the orientation-only method, the orientation is estimated on the assumption that the angular velocity has already been obtained from the gyroscope. With this angular velocity as the input and its error covariance, the estimate is accurate to within 10°, which is illustrated in Fig. 3.6 and Fig. 3.10. The advantage of this method is that the accuracy of the gyroscope is independent from the trajectory of the rigid body, which is not the case when one relies solely on the accelerometers. This can be appreciated from Fig. 3.13 (using data from Fig.)3.6.

Meanwhile, the orientation-only method works well at both high and low accelerations. This may be seen by inspecting eq. (3.31), where $\ddot{\mathbf{b}}_k$ is considered as a part of measurement noise, which implies that the accuracy of the estimates is independent from the values of the inertial acceleration. This can also be seen from Fig. 3.13. Observe as well that the filter performance may be assessed from the value of $\mathbf{R}_k$ in eq. (3.33), as a low $\mathbf{R}_k$ implies a high confidence in the measurements. This is easily understood that when there is no inertial acceleration, the orientation can totally be obtained from $\ddot{\mathbf{c}}_k$ which only includes the gravitational acceleration.

In the case of the complete pose method, the orientation estimate matches the true one only occasionally. We conjecture that this is due to the poor accuracy of the angular-velocity estimates, which are closely linked to the orientation estimates through the state equations. This may be assessed from Fig. 3.14 and Fig. 3.15 by comparing the angular velocity estimated from the complete-pose method. Contrary to the orientation-only method, the complete-pose method relies on a nonlinear relationship between the measurements and the angular veloc-



(a)

(b)

Figure 3.12: Orientation estimates and errors from different methods

ity. Although not explicitly stated, this relationship is a square-root operation, since the accelerometers measure the angular velocity squared. The square-root function has an infinite slope at zero, which explains the poor accuracy of the angular-velocity estimates whenever they are close to being null. The poor accuracy of the orientation estimates are only a consequence of this singularity of the complete-pose method.

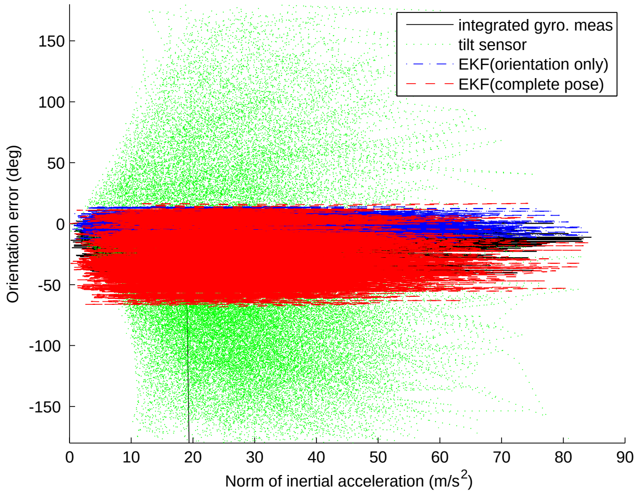


Figure 3.13: Errors of the proposed and existing orientation estimates

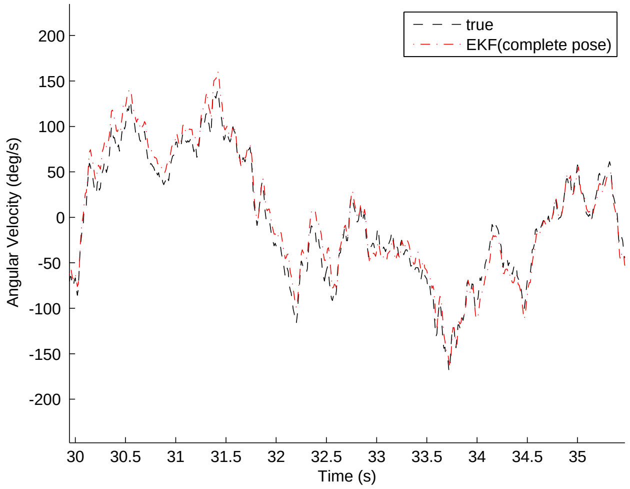


Figure 3.14: The angular velocity estimate and true one

Similarly, UKF is used in nonlinear applications as well. In this chapter, the UKF method has the same state-space system as the EKF. The difference is the way of linearization. We observe, in Fig. 3.12, that the UKF estimate matches well the true orientation as well, similar to a large extent to the orientation-only method, probably because of the good linearity of the system. Hence, the improvement of the UKF method over the EKF method appears to be insignificant.

As a matter of fact, in the UKF, the state distribution is also approximated by a Gaussian random variable, but is represented using a minimal set of carefully chosen sample points. The result is a filter which more accurately captures the true mean and covariance to the

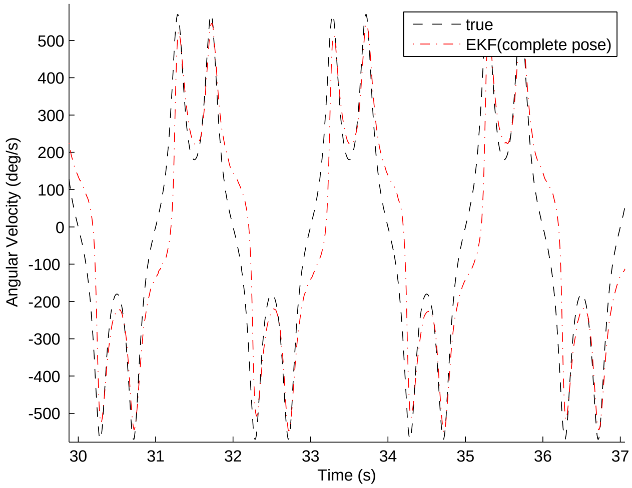


Figure 3.15: The angular velocity estimate and true one

3rd order (Taylor series expansion). The EKF, in contrast, only achieves first-order accuracy. In addition, the UKF removes the requirement to explicitly calculate Jacobians, which for complex functions can be a difficult task in itself.

From what has been discussed above, we come to the conclusion that the EKF (orientation only) and UKF (orientation only) algorithm should be preferred to the KF algorithm, mainly due to a better choice of state space model and the noise covariance matrices. Additonally, they are clearly more accurate than the MSE method, which appears to be very sensitive to errors in the inertial acceleration.

3.6 Conclusion

Estimating the orientation of a game controller moving in a vertical plane is of much importance in game playing. Its accuracy directly affects the game response. In this chapter, three new algorithms using extended Kalman filter (EKF) and unscented Kalman filter (UKF) to estimate the orientation are presented and compared with two existing methods, namely, the time-integration method and the tilt-sensor method. Because of the drift from the time-integration method and the error caused by the inertial acceleration from the tilt-sensor method, it is not recommended to use any of these methods alone. In the simulated examples, the orientation estimate from the complete-pose method matches the true one only occasionally because of a high sensitivity of the method at low angular velocities. On the other hand, the orientation-only method (EKF and UKF), which is shown to be stable, offers us an excellent orientation estimate. In addition, it works efficiently at both high and low angular velocities which denotes that the accuracy of estimate is independent from the trajectory. This will avoid the limitations of both the integration and tilt-sensor methods, and possibly open new applications in human-machine interaction.

Chapter 4

Calibration of the inertial and magnetic sensors

A beard well lathered is half shaved.

As mentioned before, the objective of this thesis is to improve the orientation estimation of game controllers like the Wii Remote with low-cost sensors. We thus design a game controller ourselves using accelerometers, gyroscopes and magnetometers. For almost all these MEMS (Micro-Electro-Mechanical System) sensors, however, the unit of initial output is voltage (for analog sensors) or LSB (Least Significant Bit) (for digital sensors), while their final units are m/s^2, °/s and Tesla, respectively. Therefore, the calibration for the sensors is necessary to determine their scale factors and offsets. As a matter of fact, most of the sensors are factory calibrated, allowing the user to avoid any further calibration for most of the applications now present in the market [68]. However, to reach a higher accuracy, it is still preferred to calibrate them, in practice. Moreover, it is unrealistic that we calibrate the sensor everytime before using a mass product. This chapter on calibration, however, is for preparation for validating or testing our proposed algorithms. We do not consider its necessity for mass product here, but may do more research in the furture work.

In this chapter, we firstly present the game controller design, in which all necessary sensors are installed. Then, these sensors are calibrated to determine their scale factors and offsets so as to obtain the final output and improve the performances of them. In addition, some tests are presented for accelerometers and gyroscopes to validate the sensitivity to the cross-axis acceleration.

4.1 Design of game-controller

In the game controller, two triaxial accelerometers are assumed to be installed at each end of it, a triaxial gyroscope and also a triaxial magnetometer are fixed in its center, as shown in Fig. 4.1. It is noted that Fig. 4.1 only presents the general information of game controller model, where the sensitive directions of accelerometers drawed there may not be consistent with the true ones. Fig. 4.2 and Fig. 3.2 show the photo and ProE drawing of the designed game-controller, respectively. Then, let us introduce the details of this game controller.

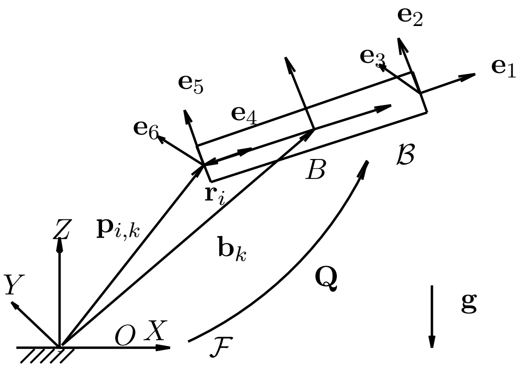


Figure 4.1: Model of the game controller with two triaxial accelerometers moving in space

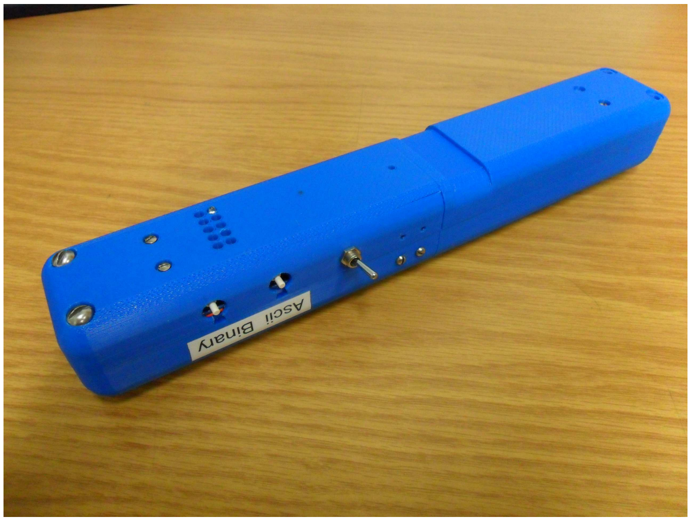


Figure 4.2: Photo of game controller

The length of the game controller is 0.3 m. The distance between the two triaxial accelerometers is $2r = 0.24$ m, where r is the half-length dimension appearing in matrix $\mathbf{A}$ in eq. (3.10). In addition to the two triaxial accelerometers, an electronic compass (magnetometer) and a gyroscope are mounted on the game controller. Bluetooth and USB are used for connecting the game controller with a PC and LEDs are used to indicate whether it is working.

a) Accelerometer array: This game controller uses the **ADXL345** accelerometers [100] from *Analog Devices*. Two such triaxial accelerometers are mounted at each end of the game controller, and thus form an accelerometer-array.

b) Electronic Compass: The **LSM303DLH** [68], a tilt compensated electronic compass, is mounted in the game-controller as well. This compass is a $5 \times 5 \times 1$ mm chip, which includes a digital triaxial accelerometer and a digital triaxial magnetic sensor. The triaxial accelerometer can help to obtain the tilt angles of pitch and roll for tilt compensation, and the magnetic sensor is capable of mesuring the earth's magnetic field, thus determining the yaw (heading) angle with respect to the magnetic north. We integrate the measurements from this sensor when generalizing the algorithm to spatial displacements.

However, the accelerometer of the electronic compass does not work very well in case of high accelerations or impacts. Because accelerometers cannot distinguish the gravitational acceleration and the inertial acceleration, high-acceleration motion causes pitch and roll calculation errors. Therefore the accuracy of the angle calculation in an electronic compass is affected by the smoothness of the game controller motion.

c) Gyroscope: For the robustness of the estimates, the **L3G4200D** [101], another MEMS motion sensor, is included. It is an ultra-stable triaxial digital output gyroscope. Mentioned in the introduction of this thesis, the triaxial gyroscope may be used in combination with the others for the estimation of spatial displacements.

d) Bluetooth: A Class 2 Bluetooth Module **RN-42** [102] is used to connect the game controller to computers. It is a small form factor, low power, highly economic wireless transmitter, which can deliver up to 3 Mbps data rate for distance to 20 m. This is quite enough for a game-controller, where a range of 5 m is sufficient in most game-playing applications.

4.2 Calibration of the accelerometers

After the accelerometers are installed in game controller, it is necessary to calibrate them, using the one degree-of-freedom rate table RT1112 [103] from ACTIDYN, in order to determine their scale factors and offsets. The rate table RT1112 accuracy on the angular velocity is reported to be within ±0.005% of its full scale range by its manufacturer. As for the positioning accuracy of the sensor, it was controlled by a custom aluminum jig built to fix the game controller on the rate table. Thus, we ignore any influence from the assembly on the measured noise. Both the rate table and the jig are shown in Fig. 4.3.

In this game controller, as mentioned in Section 4.1, there are two triaxial accelerometers. $\mathbf{e}_i$ is a unit vector representing the sensitive direction of the i^{th} accelerometer, while $\mathbf{r}_i$ represents its position with respect to the rate table center point.

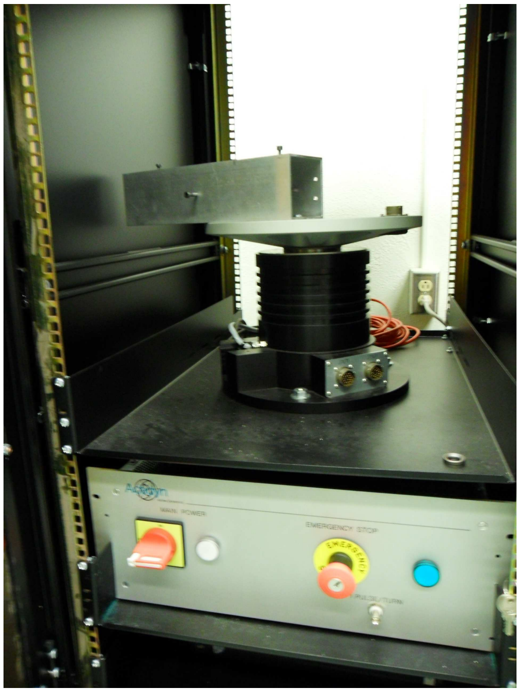


Figure 4.3: One DOF rate table

Then, we have

$$a_i = \mu_i v_{i,j} + \eta_i = \mathbf{e}_i^T \mathbf{\Omega}_j^2 \mathbf{r}_i + \mathbf{e}_i^T \mathbf{Q}_j \mathbf{g}, \qquad j = 1, \ldots, n, \tag{4.1}$$

where, ω_j is the angular velocity of step j, n is the number of samples acquired during the calibration run, $\mathbf{\Omega}_j^2 = \begin{bmatrix} -\omega_j^2 & 0 & 0 \\ 0 & -\omega_j^2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$, $\boldsymbol{\omega}_j = \omega_j \mathbf{k}$, μ_i is the scale factor, $v_{i,j}$ is the output of the accelerometer, η_i is the offset, $\mathbf{Q}_j$ is the proper orthogonal matrix that rotates the fixed frame onto the mobile one at time step j and $\mathbf{g} = [0 \ 0 \ -g]^T$ is the vector of gravitational acceleration with respect to the rate table reference. In the case of perfect installation, $\mathbf{e}_i$ is known. Finally, we have 5 unknowns μ_i, η_i and $\mathbf{r}_i$, where $i = 1, \ldots, 6$.

Eq. (4.1) can be written as

$$[v_{i,j}\ 1\ -\mathbf{e}_i^T\boldsymbol{\Omega}_j^2]\begin{bmatrix}\mu_i\\ \eta_i\\ \mathbf{r}_i\end{bmatrix}=\mathbf{e}_i^T\mathbf{Q}_j\mathbf{g}. \tag{4.2}$$

For the calibration, the game controller is fixed in different poses on the rate table, as seen in Fig. 4.4. Figure 4.5 presents the schematic drawing of the two triaxial accelerometers installed in the game controller which is fixed on the rate table. Therefore, for each triaxial accelerometer, one axis (a_{1y}, a_{2y} in Fig. 4.5) measures the centripetal acceleration, one (a_{1z}, a_{2z}) measures the gravitational acceleration and the other (a_{1x}, a_{2x}), the tangential acceleration. As the calibration is performed of constant angular velocities, the tangential acceleration is zero, and is not taken into account in the equations. Because the accelerometers are not at the center of rotation(i.e., $\mathbf{r}_i \neq \mathbf{0}_3$), a centripetal acceleration is observed by the single accelerometer which points towards the center of the rate table, or in the reverse direction. It is this accelerometer axis (e.g., a_{1y}, a_{2y} in Fig. 4.5) that is calibrated. Moreover, Fig. 4.5 informs us of the nominal sensitive directions and positions of the accelerometers:

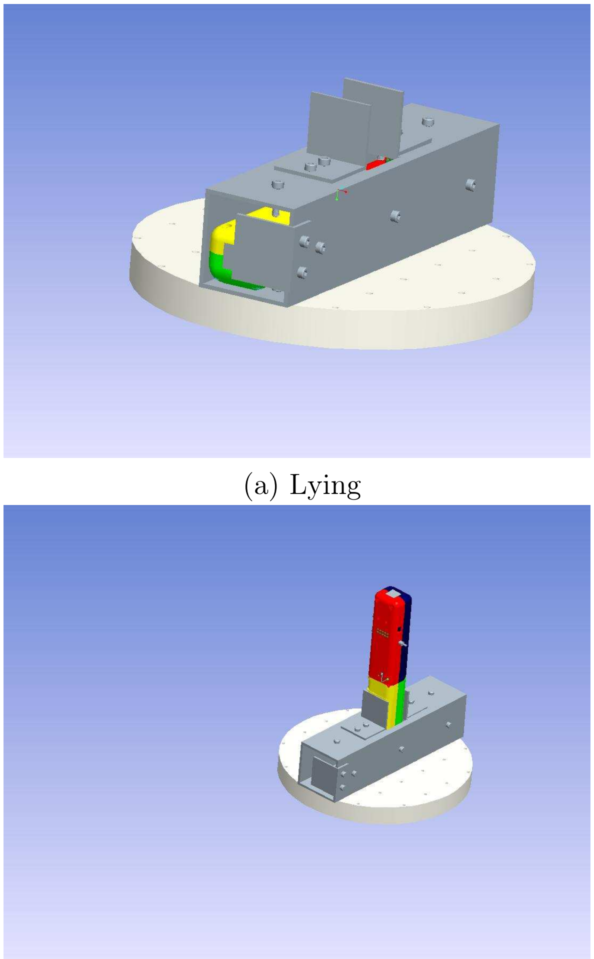
$$[\mathbf{e}_1\quad \mathbf{e}_2\cdots\mathbf{e}_6]=\begin{bmatrix}1&0&0&-1&0&0\\0&1&0&0&-1&0\\0&0&1&0&0&1\end{bmatrix},$$

$$[\mathbf{r}_1\quad \mathbf{r}_2\cdots\mathbf{r}_6]=r\begin{bmatrix}0&0&0&0&0&0\\1&1&1&-1&-1&-1\\0&0&0&0&0&0\end{bmatrix},$$

where $r = 0.15$ m.

In addition, for the horizontal accelerometer (when there are only tangential and centripetal accelerations), gravitational acceleration is not observed, which indicates that $\mathbf{e}_i^T\mathbf{Q}_j\mathbf{g} = 0$, in eq. (4.2). Because of the zero, this equation has an infinite number of solutions. We cannot distinguish between scale factor, offset and position. Thence, in order to obtain the estimated results, we need to change the game controller's direction to make the accelerometer calibrated vertical to the ground, which can observe the gravitational acceleration. Therefore, in order to calibrate each accelerometer axis, it is necessary to test two different orientations of the game controller on the rate table. Then, the new data is added in eq. (4.2), which gives

$$\begin{bmatrix}v_{i,j}&1&-\mathbf{e}_i^T\boldsymbol{\Omega}_j^2\\ v_{i,g,k}&1&-\mathbf{e}_i^T\boldsymbol{\Omega}_k^2\end{bmatrix}\begin{bmatrix}\mu_i\\ \eta_i\\ \mathbf{r}_i\end{bmatrix}=\begin{bmatrix}\mathbf{e}_i^T\mathbf{Q}_j\mathbf{g}\\ \mathbf{e}_i^T\mathbf{Q}_{g,k}\mathbf{g}\end{bmatrix}=\begin{bmatrix}0_j\\ g\cdot 1_k\end{bmatrix}, \tag{4.3}$$



(a) Lying

(b) Standing

Figure 4.4: Game controller on rate table

where $k = 1, \ldots, n$ is the time step as j, $\mathbf{Q}_{g,k}$ is the rotation matrix as $\mathbf{Q}_j$, $v_{i,g,k}$ is the output of the accelerometer that observes the gravitational acceleration at time step k. Eq. (4.3) can be written as

$$\mathbf{X}_i \boldsymbol{\beta}_i = \mathbf{y}_i, \tag{4.4}$$

where, $\mathbf{X}_i \equiv [\mathbf{x}_{i,1} \ldots \mathbf{x}_{i,n} \quad \mathbf{x}_{i,g,1} \ldots \mathbf{x}_{i,g,n}]^T$, $\mathbf{x}_{i,j} = \left[v_{i,j} \ 1 \ \ -\mathbf{e}_i^T \boldsymbol{\Omega}_j^2\right]^T$, $\mathbf{x}_{i,g,k} = \left[v_{i,g,k} \ 1 \ \ -\mathbf{e}_i^T \boldsymbol{\Omega}_k^2\right]^T$, $\boldsymbol{\beta}_i = [\mu_i \ \eta_i \ \mathbf{r}_i]^T$ and $\mathbf{y}_i = \left[\mathbf{0}_n^T \quad g \cdot \mathbf{1}_n^T\right]^T$. By using the least square method, eq. (4.4) is solved as

$$\boldsymbol{\beta}_i = \left(\mathbf{X}_i^T \mathbf{X}_i\right)^{-1} \mathbf{X}_i^T \mathbf{y}_i. \tag{4.5}$$

It is noted that keeping the accelerometer axis in one direction (accelerometer sensitive direction pointing to the center of rate table or opposite) allows us to cover only a half of its range, as the centripetal acceleration is always directed toward the rotation axis. In order to make

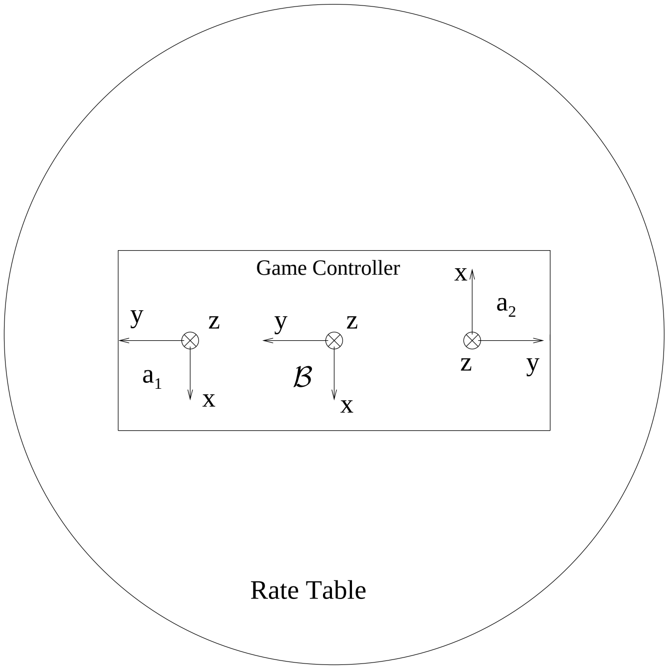


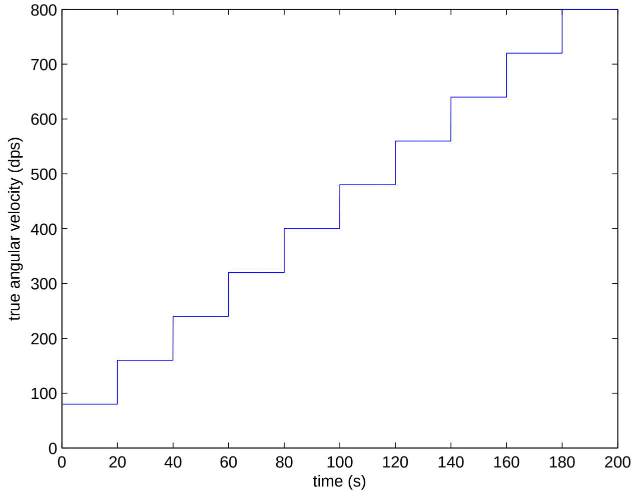
Figure 4.5: Schematic drawing of two triaxial accelerometers in game controller on rate table

the calibration results more accurate, we want to calibrate the whole range of accelerations of the sensor. Hence, the game controller is flipped in the opposite direction on the rate table as well. By adding the data from the opposite direction, eq. (4.4) is changed to

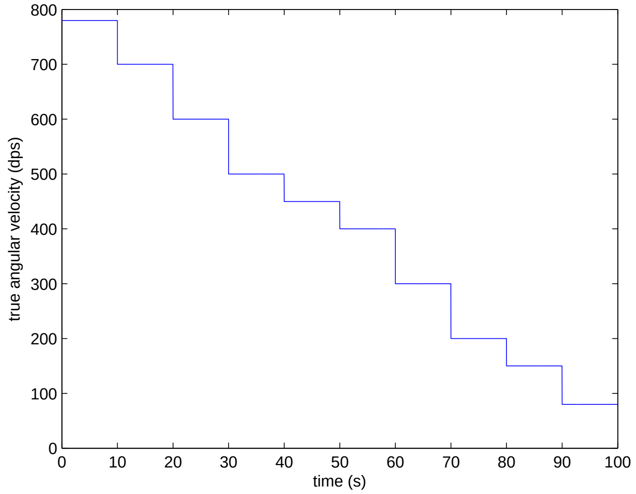
$$\begin{bmatrix} v_{i,j} & 1 & -\mathbf{e}_i^T\boldsymbol{\Omega}_j^2 & \mathbf{0}_3^T \\ v_{-i,m} & 1 & \mathbf{0}_3^T & \mathbf{e}_i^T\boldsymbol{\Omega}_m^2 \\ v_{i,g,k} & 1 & -\mathbf{e}_i^T\boldsymbol{\Omega}_k^2 & \mathbf{0}_3^T \\ v_{i,g,p} & 1 & \mathbf{0}_3^T & -\mathbf{e}_i^T\boldsymbol{\Omega}_m^2 \end{bmatrix} \begin{bmatrix} \mu_i \\ \eta_i \\ \mathbf{r}_{1,i} \\ \mathbf{r}_{2,i} \end{bmatrix} = \begin{bmatrix} \mathbf{e}_i^T\mathbf{Q}_j\mathbf{g} \\ -\mathbf{e}_i^T\mathbf{Q}_m\mathbf{g} \\ \mathbf{e}_g^T\mathbf{Q}_k\mathbf{g} \\ \mathbf{e}_g^T\mathbf{Q}_p\mathbf{g} \end{bmatrix} = \begin{bmatrix} 0_j \\ 0_m \\ g\cdot 1_k \\ g\cdot 1_p \end{bmatrix}, \tag{4.6}$$

where m and p are the time steps just as j, $\mathbf{r}_{1,i}$ is the position vector of sensitive accelerometer for the first direction and $\mathbf{r}_{2,i}$ is the position vector of sensitive accelerometer for the opposite direction.

For the two directions, several angular velocities are reached and each lasts for a period of time of 10 s or 20 s to even out the effects of noise and vibration and also avoid some inherent delays. Given the full range of accelerometers, which is $\pm 2g$ and their distance to the rotation axis, the angular velocities were chosen as a sequence increasing by 80°/s from 80°/s to 800°/s for direction 1, and a sequence decreasing by 100°/s from 800°/s to 80°/s for direction 2, where some random values were added occasionally in order to make the experimental data more convincing. Meanwhile, two directions of each accelerometer were chosen to cover both its positive and negative axes. Fig. 4.6 shows us the imposed angular velocities and Fig. 4.7 gives the corresponding output of accelerometer.



(a) Direction 1 (sensitive direction pointing to the center of rate table)



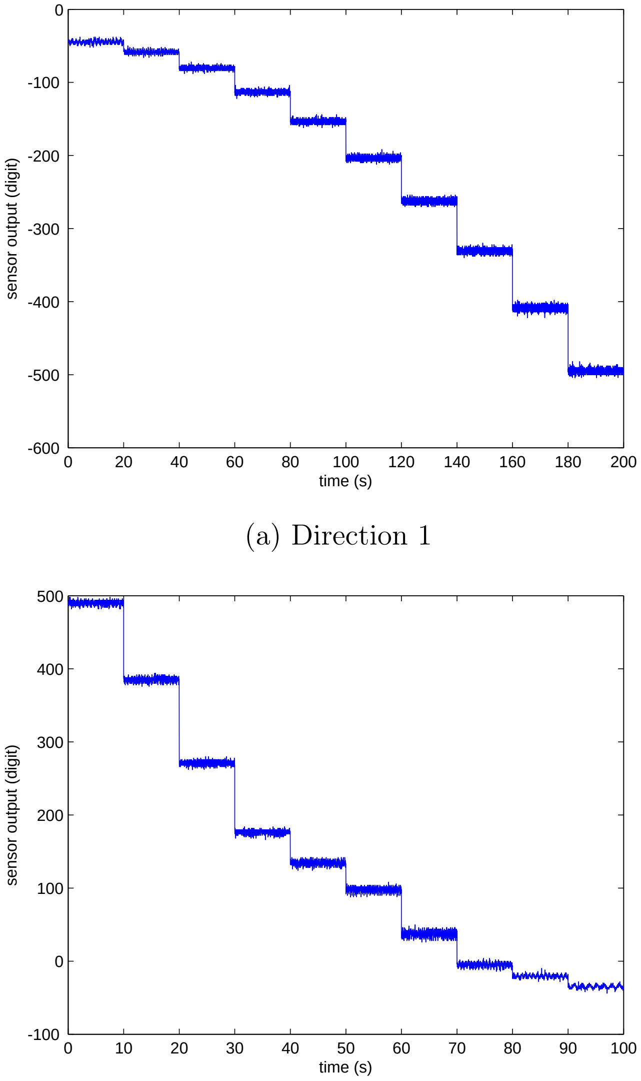
(b) Direction 2 (sensitive direction pointing in opposite direction)

Figure 4.6: Angular velocity steps

With the outputs of the accelerometers and the true angular velocities, the scale factor α_i, the offset β_i and the position parameters $\mathbf{r}_{1,i}$ and $\mathbf{r}_{2,i}$ can be estimated. Using the estimated r and the known angular velocity ω, centripetal acceleration is obtained by $a_c = r\omega^2$. Thus, combining Figs. 4.6 and 4.7, we obtain Fig. 4.8, which shows the sensor output, in counts, against the imposed acceleration.

Then, the scale factor and offset are computed from eq. (4.5), which gives the best fit line shown in Fig. 4.8. The corresponding calibration error is shown in Fig. 4.9. From this figure, the minimum and maximum calibration errors are -0.7 m/s^2 and 0.7 m/s^2, respectively.

Given the experimental results, we can compute and analyse the nonlinearity error. As we use



(a) Direction 1

(b) Direction 2

Figure 4.7: Output of an accelerometer

linear calibration laws to compute the sensor measurements from their signals, we consider any nonlinearity as a component of the error. Moreover, from the specifications of the sensors, these nonlinearities are more important than the noise, when taken over the full scale range. Similar analysis will be done for the calibration of the gyroscope in Section 4.3.

From the specifications of the ADXL345 [100] and the tests we have done, the full-scale range of the accelerometer is ±2g. Thus, the measured relative error is

$$\pm \frac{0.7 \text{ m/s}^2}{4\text{g}} \approx \pm 1.75\%. \tag{4.7}$$

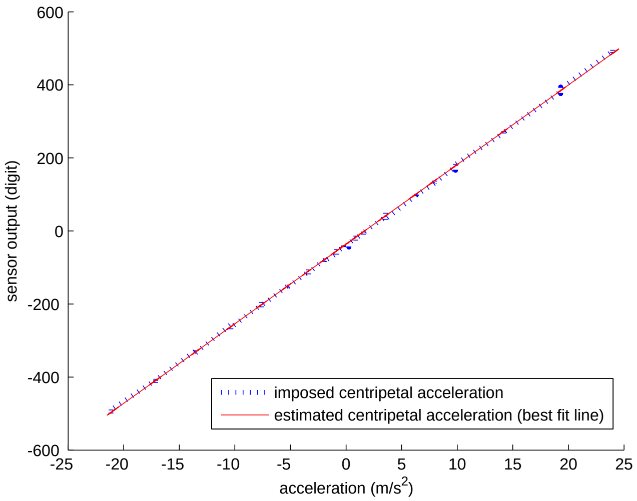


Figure 4.8: Accelerometer output versus the imposed acceleration

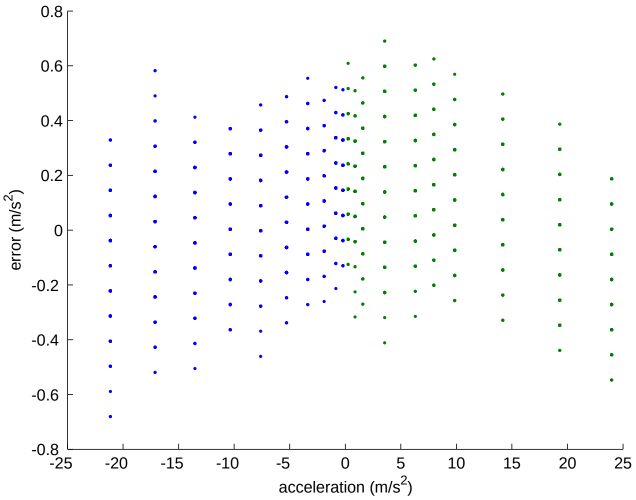


Figure 4.9: Regression error

From the accelerometer datasheet [100], the typical nonlinearity error is ±0.5%. A "typical" specification is not guaranteed, however. In addition, given the magnetic noise created by installation and the vibrations, this regression error percentage, which is somewhat larger than expected, is taken to be plausible, in our case.

We also note that the error distribution is fairly uniform over the range of the sensor. Thus, only minimal gains can be expected from the use of a higher-order calibration law. Moreover, the accelerometer outputs exhibit the same behaviour whether the imposed acceleration increases or decreases. Hence, there is no noticeable hysteresis in the response of this low-cost

Table 4.1: Scale factor and offset of the accelerometers

Acc.	Axis	scale factor (calibration) (mg/LSB)	scale factor (specification) (mg/LSB)	offset (calibration) (mg)	offset (specification) (mg)
1	x	4.24	3.5—4.3	5.3061	-150—150
1	y	4.17	3.5—4.3	34.8980	-150—150
1	z	4.23	3.5—4.3	169.6327	-250—250
2	x	3.63	3.5—4.3	10.8367	-150—150
2	y	4.00	3.5—4.3	65.8776	-150—150
2	z	4.24	3.5—4.3	149.1531	-250—250

sensor.

In summary, we calibrated both triaxial accelerometers, and obtained the scale factors and offsets of their six axes in total, which are shown in Table 4.1.

4.3 Calibration of the gyroscope

Similarly, the triaxial gyroscope is also calibrated on the one DOF rate table. We fix the gyroscope on the rate table for three different poses to make each of its sensitive directions parallel to the rotating axis of the table, shown in Fig. 4.10. In this case, the angular velocities of the rate table are considered as the gyroscope measurements. Thus, we have, for the i^{th} sensitive axis,

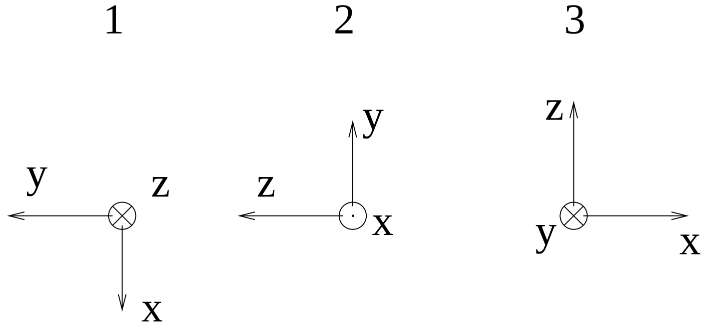


Figure 4.10: Three IMU positions shown by the sensitive axes of gyroscope

$$\widehat{\omega}_i = \mu_i v_{i,j} + \eta_i, \tag{4.8}$$

where, μ_i is the scale factor, $v_{i,j}$ is the output of gyroscope, η_i is the offset and ω_j the angular velocity of rate table.

From the datasheet [101], the scale range of the gyroscope is ±250°/s. In our tests, we found the value to be somewhat larger, at ±275°/s. In order to achieve the full range of the data, the rate table is performed from $-325°/s$ to $350°/s$, which is a arithmetic sequence with the common difference of 25°/s. This can be seen in Fig. 4.11. Then, the corresponding gyroscope output is obtained and shown in Fig. 4.12. Similar to Fig. 4.8, the relationship between the output of the gyroscope and the true angular velocity is found and shown in Fig. 4.13.

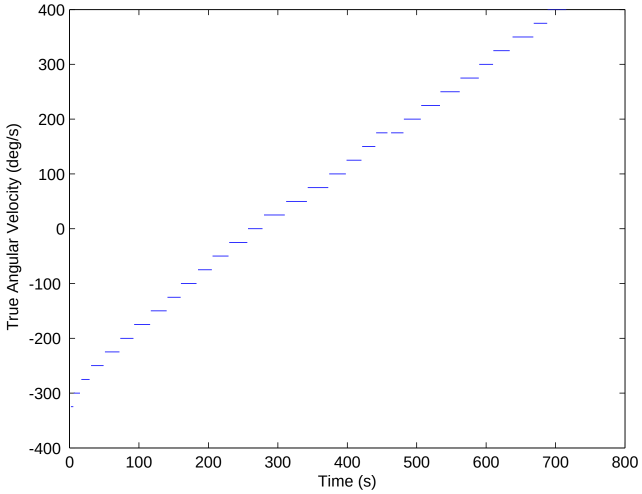


Figure 4.11: True angular velocity of rate table

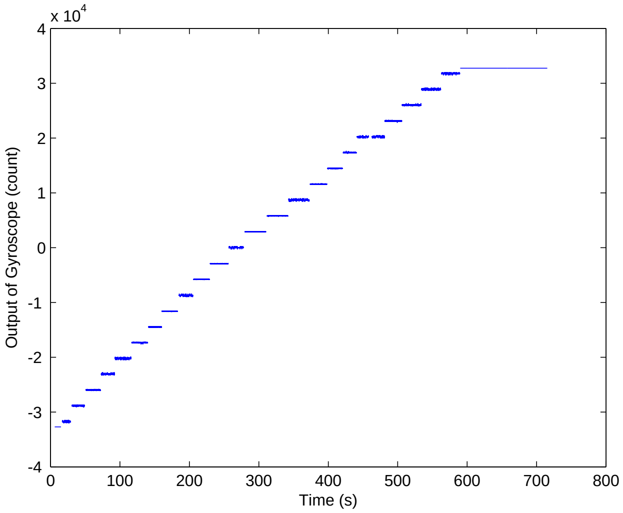


Figure 4.12: Output of gyroscope

The parameters μ_i and η_i are computed by using the linear least squares method. Therefore, the best fit line is given in Fig. 4.13 and its error is shown in Fig. 4.14. In Fig. 4.14, the

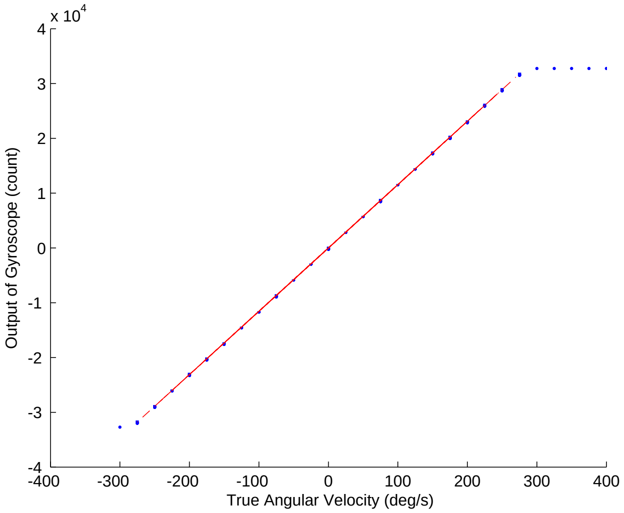


Figure 4.13: Best fit line

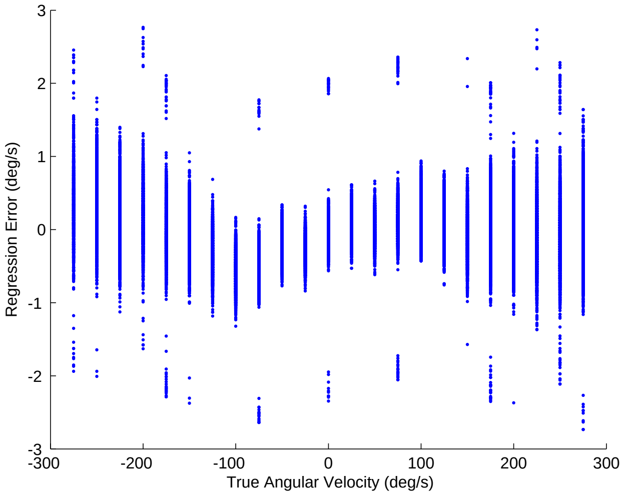


Figure 4.14: Regression error

unit of regression error is deg/s. Moreover, we find that the regression errors are disconnected from the rest at some angular velocities (e.g., 0°/s, 75°/s). This is because the output of the gyroscope is discretized. From Fig. 4.14, the maximum deviation is observed to be ±2.8°/s. The full scale range of the gyroscope being ±250°/s, we have a relative calibration of error $\pm\frac{2.8}{250 \times 2} = \pm 0.56\%$. From the gyroscope specifications [101], the nonlinearity is ±0.2%, which is typical specification and is not guaranteed. Then, similar to the analysis for accelerometer, the results we obtain are slightly less accurate than the advertised typical value, but they are still acceptable.

The error is not distributed evenly over the gyroscope range of sensitivity. From Fig. 4.14, we see a larger error in the measurements at high angular velocities, in both directions of rotation. This error seems evenly distributed on both sides of the linear regression, however, and a higher-order calibration law would not improve its accuracy significantly.

In the same manner, we estimate the scale factors and offsets for the two remaining axes of the gyroscope. Finally, the calibration parameters are shown in Table 4.2, where dps indicates degree per second, and mdps indicates milli degrees per second.

Table 4.2: Scale factor and offset of the gyroscope

Axis	Scale Factor (mdps/digit)	Offset (dps)	Correlation Coefficient
x	8.7700	-0.4978	0.999994
y	8.8805	-0.2228	0.999993
z	8.6502	-0.0655	0.999992
specification	8.75		

It is noted that the scale factor in the datasheet of the gyroscope, which is typical specification, is 8.75 mdps/digit, which is close to the estimated values.

One factor that we conjecture could influence the accuracy of the gyroscopes is the cross axis sensitivity to acceleration. MEMS gyroscopes measure the angular velocity through the Coriolis effect. When rotating the Coriolis force causes the displacement of an oscillating mass in the MEMS gyroscope, which is read from a capacitive sensing structure. The differential capacitance is proportional to the angular velocity and is then converted into output for gyroscope. Hence, the output of the gyroscopes is determined by this Coriolis force. An external transverse acceleration can also cause the same displacement, however. Although these gyroscopes are designed to be significantly more sensitive to angular velocity than point acceleration, this phenomenon still affects, to a degree, the precision of gyroscope. Therefore, two other tests are performed to evaluate the influence.

Firstly, we change the position of the game controller on the rate table and obtain the best fit lines shown in Fig. 4.15, where r represents the distance from the center of rotation to the gyroscope. Then, the difference between the two best fit lines is shown in Fig. 4.16. we decide to analyse the results after the second test is done.

Secondly, the back and forth test is performed, which consists in increasing and decreasing the angular velocity, as shown in Fig. 4.17. The best fit lines are thus presented in Fig. 4.18. Then, the difference between two best fit lines is shown in Fig. 4.19.

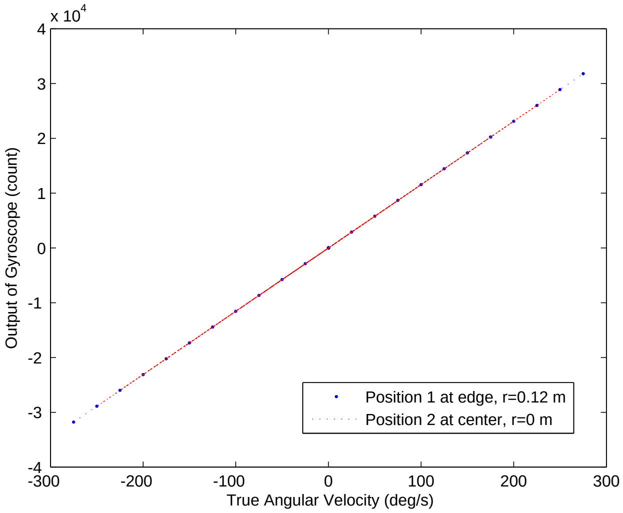


Figure 4.15: Best fit line for two positions

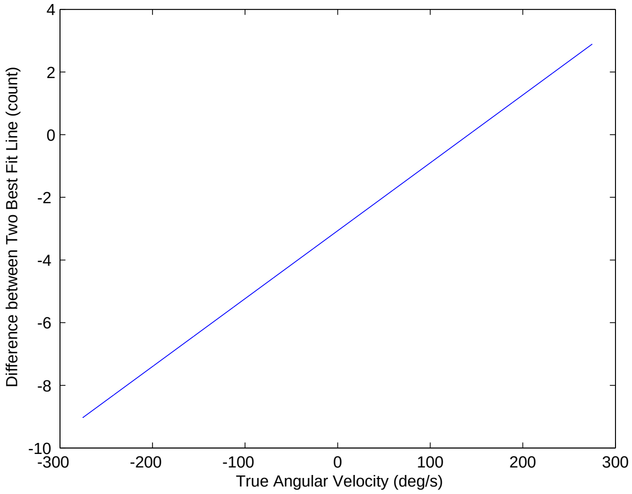


Figure 4.16: Difference between two best fit lines (two positions)

As the difference from the second test is larger than that from the first one, we just analyse the second test. From Fig. 4.19, the absolute value of the difference is at most $d = 40$ digital counts. Meanwhile, the maximum absolute value of the gyroscope output, from Fig. 4.18, is 3×10^4. Then, we have

$$40/(3 \times 10^4) \approx 0.13\%, \tag{4.9}$$

which is small relative to the calibration errors previously observed, which were of 0.56% at most.

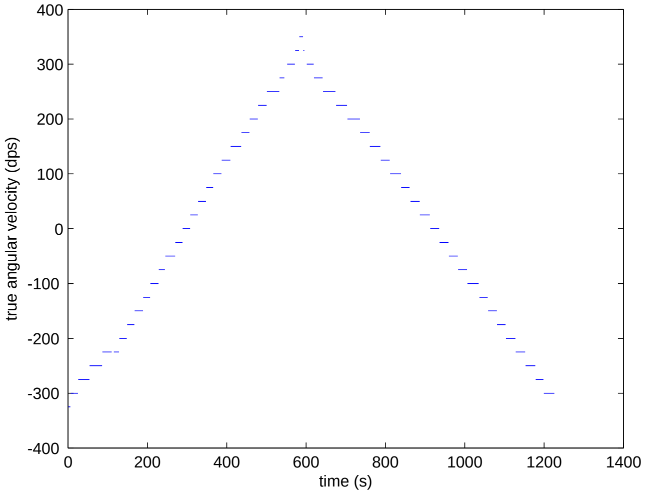


Figure 4.17: True angular velocity (back and forth)

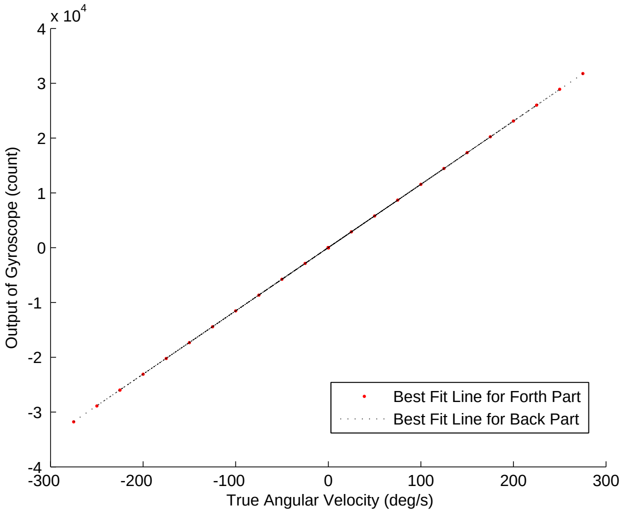


Figure 4.18: Best fit lines for the back and forth test

4.4 Calibration of the magnetometer

As mentioned above, magnetometer is installed for the sake of azimuth angle, as inertial sensors are not capable of providing information on this angle, which is vertical to the gravitational field. Prior to use the magnetometer, we wish to calibrate [87]. To this end, we consider the model illustrated above, where the true North is represented by the unit vector

$$\mathbf{m} \equiv \mathbf{m}_{\mathcal{F}}. \tag{4.10}$$

This true magnetic North is to be measured by the magnetometer attached to a rigid body. The magnetometer measurements are taken in frame $\mathcal{M}$, which is obtained by rotating the

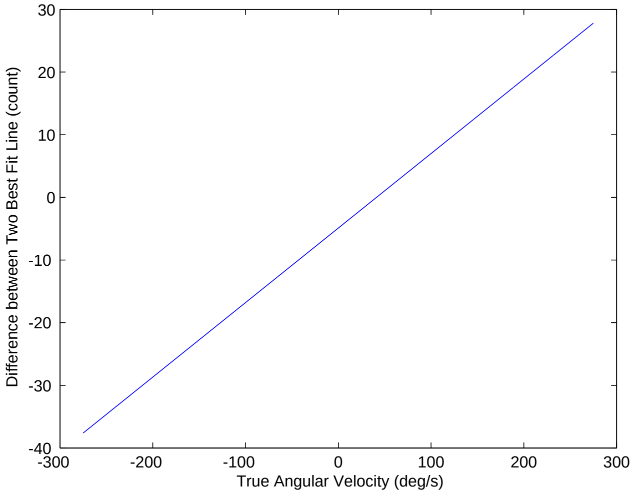


Figure 4.19: Difference between two best fit lines (back and forth)

body-fixed frame $\mathcal{O}$ through a rotation $\mathbf{R}_{\mathcal{OM}}$. The matrix $\mathbf{R}_{\mathcal{OM}}$ remains constant as the rigid body moves, but is yet unknown. Frame $\mathcal{O}$, in turn, is obtained by a rotation $[\mathbf{Q}_k]_{\mathcal{FO}}$ of the fixed frame $\mathcal{F}$. $[\mathbf{Q}_k]_{\mathcal{FO}}$ varies as the rigid body moves, but is known, as it is measured by the Optitrack sensor, which will be described in detail in Section 5.4.

From the magnetometer, we obtain an estimate $\widehat{\mathbf{m}}_{\mathcal{M},k}$ of the true North direction, in its local frame $\mathcal{M}$. Thus we must link the estimate to the true North, which gives

$$\mathbf{m} = \widehat{\mathbf{m}}_{\mathcal{F},k} = [\mathbf{Q}_k]_{\mathcal{FO}}\mathbf{R}_{\mathcal{OM}}\widehat{\mathbf{m}}_{\mathcal{M},k}. \tag{4.11}$$

In eq. (4.11), $[\mathbf{Q}_k]_{\mathcal{FO}}$ and $\widehat{\mathbf{m}}_{\mathcal{M},k}$ are measured, while $\mathbf{m}$ and $\mathbf{R}_{\mathcal{OM}}$ remain unknown. Let us parameterize $\mathbf{R}_{\mathcal{OM}}$ with the Euler angle θ (pitch), ϕ (roll) and ψ (azimuth), according to the XYZ convention. Thus, we obtain

$$\mathbf{R}_{\mathcal{OM}} = \mathbf{R}(\theta, \phi, \psi)_{\mathcal{OM}}. \tag{4.12}$$

Upon premultiplying both sides of eq. (4.11) with $\mathbf{Q}_k^T$, and subtracting the left side from the right one, we obtain

$$f_k(\mathbf{m}, \theta, \phi, \psi) \equiv \mathbf{R}(\theta, \phi, \psi)_{\mathcal{OM}}\widehat{\mathbf{m}}_{\mathcal{M},k} - [\mathbf{Q}_k]_{\mathcal{FO}}^T\mathbf{m}. \tag{4.13}$$

Under the assumption of perfect measurements and model, we must have $\mathbf{f}_k = \mathbf{0}_3$. In practice, however, $\mathbf{f}_k \neq \mathbf{0}_3$, and the problem becomes that of finding the parameters $\mathbf{m}$, θ, ϕ and ψ such that the norm of the vectors $\mathbf{f}_k$, $k = 1, \ldots, n$, is minimized, where n is number of samples.

Symbolically, we have

$$\text{minimize} \quad \frac{1}{n}\sum_{k=1}^{n} \|\mathbf{f}_k(\mathbf{m},\theta,\phi,\psi)\|_2^2, \tag{4.14}$$

$$\text{over} \quad \mathbf{m},\theta,\phi \text{ and } \psi. \tag{4.15}$$

This is a nonlinear least-squares problem, which can be solved in Matlab using the *lsqnonlin* command.

A question remains as to how we obtain the magnetometer measurement $\widehat{\mathbf{m}}_{\mathcal{M},k}$, from its output voltage $\mathbf{v}_k$. According to the calibration procedure provided by STMicroelectronics, the sensor output voltage should lie on the surface of an ellipsoid (seen in Fig. 4.20), i.e.,

$$(\mathbf{v}_k - \mathbf{v}_0)^T \mathbf{E}(\mathbf{v}_k - \mathbf{v}_0) - 1 = 0, \tag{4.16}$$

where $k = 1, \ldots, p$. In practice, the voltage is never exactly on the surface of one ellipsoid. Hence, one must find the ellipsoid that fits best by solving another nonlinear least-squares problem,

$$\text{minimize} \quad \frac{1}{n}\sum_{k=1}^{n} \|\mathbf{g}_k(\mathbf{E},\mathbf{v}_0)\|_2^2, \tag{4.17}$$

$$\text{subject to} \quad \mathbf{E} = \mathbf{E}^T, \tag{4.18}$$

$$\text{over} \quad \mathbf{E}, \mathbf{v}_0. \tag{4.19}$$

where $\mathbf{g}_k(\mathbf{E},\mathbf{v}_0) \equiv (\mathbf{v}_k - \mathbf{v}_0)^T \mathbf{E}(\mathbf{v}_k - \mathbf{v}_0) - 1$. As before, problem (4.17) can be solved using Matlab's *lsqnonlin* command.

Although the equation of the best ellipsoid is obtained, this does not yet tell us how the voltage measurements $\mathbf{v}_k$ are linked with the magnetic field direction $\widehat{\mathbf{m}}_{\mathcal{M},k}$. Since $\widehat{\mathbf{m}}_{\mathcal{M},k}$ is a unit vector, geometrically, we are looking for a one-to-one correspondence between the ellipsoid of eq. (4.16) and the unit sphere centered at the origin. Such a mapping is obtained by the linear relationship

$$\widehat{\mathbf{m}}_{\mathcal{M},k} = \sqrt{\mathbf{E}}(\mathbf{v}_k - \mathbf{v}_0), \tag{4.20}$$

where $\sqrt{\mathbf{E}}$ is the matrix square-root, i.e., the matrix that has the same eigenvectors as $\mathbf{E}$, but whose eigenvalues are the square-roots of those of $\mathbf{E}$. For $\widehat{\mathbf{m}}_{\mathcal{M},k}$ to be a unit vector, $\mathbf{v}_k$ has to lie exactly on the ellipsoid of eq.(4.16). Since this is not the case in general, it is a necessary precaution to normalize $\widehat{\mathbf{m}}_{\mathcal{M},k}$, i.e., to use instead

$$\widehat{\mathbf{m}}_{\mathcal{M},k} = \frac{\sqrt{\mathbf{E}}(\mathbf{v}_k - \mathbf{v}_0)}{\|\sqrt{\mathbf{E}}(\mathbf{v}_k - \mathbf{v}_0)\|_2}. \tag{4.21}$$

Finally, after scale factor and offset compensation, we obtain a centered unit sphere, as seen in Fig. 4.21.

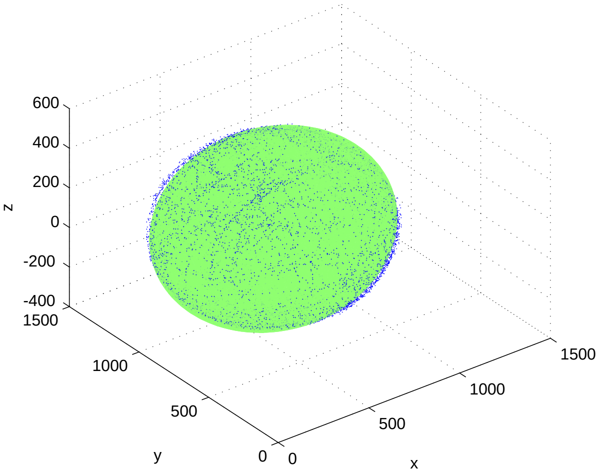


Figure 4.20: Magnetometer output (raw data)

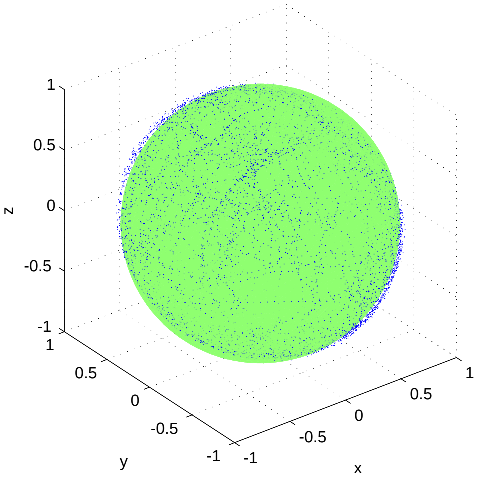


Figure 4.21: Magnetometer output (after compensation)

4.5 Conclusion

In this chapter, we firstly design a game controller in which accelerometers, gyroscope and magnetometer were installed. Then, in order to use the output of these sensors correctly and accurately, we calibrated accelerometers and gyroscope using a one DOF rate table and magnetometer using the Optitrack system to determine their scale factors and offsets. In the calibration, the accelerometers and gyroscopes were observed to be fairly insensitive to cross-axis accelerations, to hysteresis and exhibited good linearity. These observations tend to validate the model that we have been using, whereby errors are assumed to be Gaussian white noise. Meanwhile, the output of magnetometer lied almost on the surface of an ellipsoid, which is obtained by solving a nonlinear least-squares problem. After scale factor and offset compensation, the ellipsoid becomes a centered unit sphere, representing the magnetic field direction. Finally, the fixed true North is found, by using the Optitrack system as the reference.

Chapter 5

Estimating the orientation in space

Deux precautions valent mieux qu'une.

This chapter presents a novel method of estimating the orientation of a rigid body moving in space from inertial sensors, which has been published in [104], by discerning the gravitational and inertial components of accelerations. In this method, both a rigid-body kinematics model and a stochastic model of the human-hand motion are formulated and combined in a nonlinear state-space system. The state equation represents the rigid-body kinematics and stochastic model and the output equation represents the inertial sensor measurements. It is necessary to mention that, since the output equation is the nonlinear function of the state, the application of the extended Kalman filter (EKF) is resorted to. The absolute value of the error from the proposed method is shown to be less than 5° in simulation and in experiments. It is stable, unlike the time-integration of gyroscope measurements, which is subjected to drift, and remains accurate under large accelerations, unlike the tilt-sensor method.

Multimedia extension

- The video *manette_ GUI.avi* presents a simple virtual environment to demonstrate usefulness of the proposed algorithm.

5.1 The proposed estimation method

After estimating the orientation in the plane in Chapter 3, the final objective is to estimate the orientation in space. Instead of the model of Fig. 3.1 in Section 3.1, we are using the model of Fig. 2.1 in Section 2.1.

Like most other researchers, we rely on Kalman filtering to estimate the rotations of the hand from inertial measurements. Because the kinematics of rotations are nonlinear, we must resort to the extended Kalman filter to estimate the attitude of the hand. This, in turn, requires that a model of the hand rotations and sensing be written as a standard state-space system of the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{u}, \tag{5.1a}$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) + \mathbf{v}, \tag{5.1b}$$

where $\mathbf{x}$ is the state vector, $\mathbf{f}(\mathbf{x})$ is the state function, $\mathbf{u}$ is the process noise, $\mathbf{y}$ is the output vector, $\mathbf{h}(\mathbf{x})$ is the output function, and $\mathbf{v}$ is the output noise. Thence, let us describe how we propose to model the rotations of the human hand.

5.1.1 Stochastic model of the hand kinematics

We assume that the moments applied by the user on his or her own hand vary according to a Gaussian white-noise stochastic process. From Newton's second laws of motion, we infer that the angular acceleration is Gaussian white noise, since it is linearly related to the moments. From the angular acceleration, one can compute the angular jerk $\boldsymbol{\gamma}$ [91] by a numerical differentiation, which is a linear operation in discrete time ($\gamma_k = (\alpha_{k+1} - \alpha_k)/\Delta t$). Hence, the angular jerk is a Gaussian white noise as well. It can be rewritten as

$$\boldsymbol{\gamma} = \dot{\boldsymbol{\alpha}}. \tag{5.2}$$

Since it is assumed to be Gaussian white noise, we can also write that $\boldsymbol{\gamma}$ follows a normal distribution, namely,

$$\boldsymbol{\gamma} \sim \mathcal{N}(\mathbf{0}, \sigma_\gamma^2 \mathbf{1}_{3\times3}). \tag{5.3}$$

However crude this approximation may seem, we can only judge of its quality by the results it provides, examples of which are presented in Sections 5.3 and 5.4.

5.1.2 Kinematics of rigid-body rotations

Let us start from the angular acceleration, which is related to the rigid-body angular velocity $\boldsymbol{\omega}$ through the simple relationship

$$\boldsymbol{\alpha} = \dot{\boldsymbol{\omega}}. \tag{5.4}$$

In turn, relating the angular velocity to rigid-body rotations requires a proper parameterization thereof. To this end, we resort to the rotation matrix $\mathbf{Q}$ taking the fixed frame $\mathcal{F}$ onto the moving frame $\mathcal{B}$. Most authors prefer instead Euler angles or quaternions for this purpose. Here, our choice is motivated by the observation that the rotation matrix appears linearly in the accelerometer output equations, as shown in Section 5.1.4. As seen below, the use of rotation matrices leads to simple, compact equations that are easily understood and implemented, which is not the case with other representations. Besides, the rotation-matrix representation of rotations is free of singularities, which is not the case with Euler-angles representations [105].

From [88], we have a relationship between the rotation matrix $\mathbf{Q}$ and $[\mathbf{\Omega}]_{\mathcal{F}} \equiv \mathrm{cpm}([\boldsymbol{\omega}]_{\mathcal{F}})$, the cross-product matrix[1] of the angular velocity expressed in the fixed frame $\mathcal{F}$. This relationship is

$$[\mathbf{\Omega}]_{\mathcal{F}} = \dot{\mathbf{Q}}\mathbf{Q}^T = \mathbf{Q}[\mathbf{\Omega}]_{\mathcal{B}}\mathbf{Q}^T, \tag{5.5}$$

where the latter equality simply represents a change of frame from $\mathcal{B}$ to $\mathcal{F}$. Solving for $\dot{\mathbf{Q}}$, we obtain

$$\dot{\mathbf{Q}} = \mathbf{Q}[\mathbf{\Omega}]_{\mathcal{B}}. \tag{5.6}$$

In order to reach the standard form of a state-space system, we define the rows of the rotation matrix $\mathbf{Q}$ as

$$\mathbf{Q} = [\mathbf{q}_1 \quad \mathbf{q}_2 \quad \mathbf{q}_3]^T. \tag{5.7}$$

Therefore, we have

$$[\dot{\mathbf{q}}_1 \quad \dot{\mathbf{q}}_2 \quad \dot{\mathbf{q}}_3] = \dot{\mathbf{Q}}^T = [\mathbf{\Omega}]_{\mathcal{B}}^T\mathbf{Q}^T = [\mathbf{\Omega}]_{\mathcal{B}}^T[\mathbf{q}_1 \quad \mathbf{q}_2 \quad \mathbf{q}_3]. \tag{5.8}$$

Integration to the Kalman filter requires that we rewrite eq. (5.8) as

$$\begin{bmatrix} \dot{\mathbf{q}}_1 \\ \dot{\mathbf{q}}_2 \\ \dot{\mathbf{q}}_3 \end{bmatrix} = \begin{bmatrix} [\mathbf{\Omega}]_{\mathcal{B}}^T & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & [\mathbf{\Omega}]_{\mathcal{B}}^T & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & [\mathbf{\Omega}]_{\mathcal{B}}^T \end{bmatrix} \begin{bmatrix} \mathbf{q}_1 \\ \mathbf{q}_2 \\ \mathbf{q}_3 \end{bmatrix}. \tag{5.9}$$

5.1.3 Rotation-matrix orthogonality

Matrix $\mathbf{Q}$ is a rotation matrix if and only if it is proper orthogonal, that is, if it satisfies the constraints

$$\det(\mathbf{Q}\mathbf{Q}^T) = 1, \tag{5.10}$$

$$\mathbf{Q}\mathbf{Q}^T - \mathbf{1}_{3\times3} = \mathbf{0}_{3\times3}. \tag{5.11}$$

Because of measurement and modeling errors, we must expect the estimate of $\mathbf{Q}$ to violate these constraints at any time if not all the time. This is not catastrophic, as long as the constraint violation does not rise above a certain level to be decided by the users.

In order to integrate this idea within the state-space formulation, we regard the output of the constraint $\mathbf{Q}\mathbf{Q}^T - \mathbf{1}_{3\times3}$ as a fake measurement. The output of this measurement is allowed

[1] The cross-product matrix of $\mathbf{a}$, $\mathrm{cpm}(\mathbf{a})$, is defined as $\partial(\mathbf{a} \times \mathbf{x})/\partial\mathbf{x}$, for any $\mathbf{a}, \mathbf{x} \in \mathbb{R}^3$.

to remain its true value of $\mathbf{0}_{3\times3}$ by the addition of a measurement error, which is modeled as white noise. The state-space formulation of these ideas requires the definition of substituting eq. (5.7) into eq. (5.11). We define

$$\mathbf{s} = \begin{bmatrix} \mathbf{q}_1^T\mathbf{q}_1 - 1 \\ \mathbf{q}_2^T\mathbf{q}_2 - 1 \\ \mathbf{q}_3^T\mathbf{q}_3 - 1 \\ \mathbf{q}_2^T\mathbf{q}_3 \\ \mathbf{q}_3^T\mathbf{q}_1 \\ \mathbf{q}_1^T\mathbf{q}_2 \end{bmatrix}, \tag{5.12}$$

the constraint-violation measurement. $\mathbf{s}$ is endowed with a small noise $\delta\mathbf{s}$ allowing the measurement to retain exactly null despite measurement errors, and preventing the singularities in the algorithm calculation process,

$$\mathbf{0}_6 = \mathbf{s} + \delta\mathbf{s}. \tag{5.13}$$

$\delta\mathbf{s}$ is assumed to follow the normal distribution,

$$\delta\mathbf{s} \sim \mathcal{N}(\mathbf{0}, \sigma_s^2\mathbf{1}_{6\times6}). \tag{5.14}$$

5.1.4 Accelerometer measurements

Like orthogonality constraints, the inertial measurements are system outputs, symbolized by the accelerations $\widehat{a}_i, i = 1, \ldots, m$, from the m uniaxial accelerometers, and the angular velocity $\widehat{\boldsymbol{\omega}}$ from gyroscope, respectively. The measurement $\widehat{a}_i$ of this ith accelerometer can be modeled as

$$\widehat{a}_i = a_i + \delta a_i, \tag{5.15}$$

where a_i is the true acceleration component along the accelerometer sensitive direction, and δa_i is the measurement error.

We assume that the measurement errors are zero-mean, identically, independently distributed Gaussian white noise $\delta\mathbf{a}$, i.e.,

$$\delta\mathbf{a} \sim \mathcal{N}(\mathbf{0}_m, \sigma_a^2\mathbf{1}_{m\times m}), \tag{5.16}$$

where, $\delta\mathbf{a} \equiv [\delta a_1 \quad \cdots \quad \delta a_m]^T$, $\mathcal{N}$ represents the normal distribution, $\mathbf{0}_m$ is the m-dimensional zero vector, σ_a^2 is the measurement-error variance of a single accelerometer, and $\mathbf{1}_{m\times m}$ represents the $m \times m$ identity matrix.

The accelerometer-array input–output relationships are discussed in Chapter 1. From eq. (2.13), we have

$$\begin{aligned}\hat{\mathbf{a}} &= \mathbf{A}\mathbf{z} + \delta\mathbf{a} \\ &= \mathbf{A}_P\ddot{\mathbf{b}} - \mathbf{A}_P\mathbf{G}\mathbf{q} + \mathbf{A}_T\boldsymbol{\alpha} + \mathbf{A}_C\boldsymbol{\xi} + \delta\mathbf{a}, \end{aligned} \tag{5.17}$$

where $\ddot{\mathbf{b}}$ is the inertial acceleration, $\mathbf{G} = -\begin{bmatrix} g_x\mathbf{1}_{3\times3} & g_y\mathbf{1}_{3\times3} & g_z\mathbf{1}_{3\times3} \end{bmatrix}$, $\mathbf{g} = [g_x\ g_y\ g_z]^T$, and $\mathbf{q} = [\mathbf{q}_1^T\ \mathbf{q}_2^T\ \mathbf{q}_3^T]^T$.

The problem is that we don't know the inertial acceleration $\ddot{\mathbf{b}}_k$. However, it is possible to consider it as random motion whose covariance can be estimated. For a game controller being acted upon by a human, although we have no deterministic way of predicting the trajectory, we may estimate, however coarsely, the probability of moving from one pose to the next within one time step. In particular, we may attempt to quantify the likely accelerations of the human hand in given applications. Therefore, $\ddot{\mathbf{b}}$ is assumed to follow the normal distribution,

$$\ddot{\mathbf{b}} \sim \mathcal{N}(\mathbf{0}_m, \sigma_b^2\mathbf{1}_{3\times3}). \tag{5.18}$$

5.1.5 Gyroscope measurements

For the angular velocity $\widehat{\boldsymbol{\omega}}$ from the gyroscope, like eq. (5.15), we have

$$\widehat{\boldsymbol{\omega}} = \boldsymbol{\omega} + \delta\boldsymbol{\omega}, \tag{5.19}$$

where $\delta\boldsymbol{\omega}$ is the gyroscope measurement noise and which is assumed to follow the Gaussian distribution,

$$\delta\boldsymbol{\omega}_k \sim \mathcal{N}(\mathbf{0}_3, \sigma_\omega^2\mathbf{1}_{3\times3}). \tag{5.20}$$

5.1.6 Magnetometer measurements

Having the gravitational field, where $\mathbf{g} = [g_x\ g_y\ g_z]^T$, Earth also has the magnetic field, which gives the magnetic vector $[\mathbf{m}]_{\mathcal{F}} = [m_x\ m_y\ m_z]^T$. Therefore, a magnetometer, whose output is $[\widehat{\mathbf{m}}]_{\mathcal{O}}$, can be installed to estimate the heading attitude. Similar to eq. (5.15) and eq.(5.19), we have the magnetic intensity vector $\widehat{\mathbf{m}}$,

$$\widehat{\mathbf{m}} = \mathbf{m} + \delta\mathbf{m}, \tag{5.21}$$

where $\mathbf{m}$ is the true magnetic intensity component along the magnetometer sensitive direction, which is discussed in Section 4.4, $\delta\mathbf{m}$ is the magnetometer measurement noise and which is assumed to follow the Gaussian distribution,

$$\delta\mathbf{m}_k \sim \mathcal{N}(\mathbf{0}_3, \sigma_m^2\mathbf{1}_{3\times3}). \tag{5.22}$$

Then, like the expression in eq. (5.17) and from eq. (5.59), we have

$$\begin{aligned}
[\mathbf{m}]_{\mathcal{O}} = \mathbf{Q}^T_{\mathcal{F}\mathcal{O}}\mathbf{m}_{\mathcal{F},k} &= \mathbf{R}^T_{\mathcal{B}\mathcal{O}}\mathbf{Q}^T_{\mathcal{B}}[\mathbf{m}]_{\mathcal{F}} = \mathbf{R}^T_{\mathcal{B}\mathcal{O}}[\mathbf{q}_1\ \mathbf{q}_2\ \mathbf{q}_3][m_x\ m_y\ m_z]^T \\
&= \mathbf{R}^T_{\mathcal{B}\mathcal{O}}(m_x\mathbf{q}_1 + m_y\mathbf{q}_2 + m_z\mathbf{q}_3) \\
&= \mathbf{R}^T_{\mathcal{B}\mathcal{O}}\mathbf{M}\mathbf{q},
\end{aligned} \tag{5.23}$$

where $\mathbf{Q}_{\mathcal{B}} = \mathbf{Q}$, $\mathbf{m} = [m_x\ m_y\ m_z]$ is the vector of magnetic intensity pointing to the magnetic North, $\mathbf{M} = [m_x\mathbf{1}_{3\times 3}\ m_y\mathbf{1}_{3\times 3}\ m_z\mathbf{1}_{3\times 3}]$, $\mathbf{q} = [\mathbf{q}_1^T\ \mathbf{q}_2^T\ \mathbf{q}_3^T]^T$, $\mathbf{Q}_{\mathcal{F}\mathcal{O}}$ and $\mathbf{R}_{\mathcal{B}\mathcal{O}}$ are defined in Section 5.4.

5.1.7 State-space system

Let us regroup the equations that model the system (5.2,5.4,5.9) and those that model the measurements (5.13,5.17,5.19) into the standard form of a (nonlinear) state-space, in which each uniaxial sensor corresponds to one output equation. This yields

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{u}, \tag{5.24}$$

$$\mathbf{y} = \mathbf{h}(\mathbf{x}) + \mathbf{v}, \tag{5.25}$$

where

$$\begin{aligned}
\mathbf{x} &= [\mathbf{q}_1^T\quad \mathbf{q}_2^T\quad \mathbf{q}_3^T\quad \boldsymbol{\alpha}^T\quad \boldsymbol{\omega}^T]^T, \\
\mathbf{f}(\mathbf{x}) &= \left[\left([\boldsymbol{\Omega}]^T_{\mathcal{B}}\mathbf{q}_1\right)^T\quad \left([\boldsymbol{\Omega}]^T_{\mathcal{B}}\mathbf{q}_2\right)^T\quad \left([\boldsymbol{\Omega}]^T_{\mathcal{B}}\mathbf{q}_3\right)^T\quad \mathbf{0}_3^T\quad \boldsymbol{\alpha}^T\right]^T, \\
\mathbf{u} &= \left[\mathbf{0}_9^T\quad \boldsymbol{\gamma}^T\quad \mathbf{0}_3^T\right]^T, \\
\mathbf{y} &= [\widehat{\mathbf{a}}^T\quad \widehat{\boldsymbol{\omega}}^T\quad \widehat{\mathbf{m}}^T\quad \mathbf{0}_6^T]^T, \\
\mathbf{h}(\mathbf{x}) &= [(-\mathbf{A}_P\mathbf{G}\mathbf{q} + \mathbf{A}_T\boldsymbol{\alpha} + \mathbf{A}_C\boldsymbol{\xi})^T\quad \boldsymbol{\omega}^T\quad (\mathbf{R}_{\mathcal{B}\mathcal{O}}\mathbf{M}\mathbf{q})^T\quad \mathbf{s}^T]^T, \\
\mathbf{v} &= [(\mathbf{A}_P\ddot{\mathbf{b}} + \delta\mathbf{a})^T\quad \delta\boldsymbol{\omega}^T\quad \delta\mathbf{m}^T\quad \delta\mathbf{s}^T]^T.
\end{aligned}$$

Because the experimental data are obtained by sampling, we need to convert equations (5.24) and (5.25) from continuous to discrete time. Let us first compute the state-transition matrix from one rotation-matrix to the next,

$$\boldsymbol{\Phi}_k = \mathbf{1}_{3\times 3} + [\boldsymbol{\Omega}_k]^T_{\mathcal{B}}\Delta t, \tag{5.26}$$

where Δt is the sampling period. Thence, the state-space system can be rewritten as

$$\mathbf{x}_{k+1} = \begin{bmatrix} \boldsymbol{\Phi}_k\mathbf{q}_{1,k} \\ \boldsymbol{\Phi}_k\mathbf{q}_{2,k} \\ \boldsymbol{\Phi}_k\mathbf{q}_{3,k} \\ \boldsymbol{\alpha}_k \\ \boldsymbol{\omega}_k + \Delta t\boldsymbol{\alpha}_k \end{bmatrix} + \begin{bmatrix} \mathbf{0}_3 \\ \mathbf{0}_3 \\ \mathbf{0}_3 \\ \Delta t\boldsymbol{\gamma}_k \\ (1/2)\,\Delta t^2\boldsymbol{\gamma}_k \end{bmatrix}, \tag{5.27}$$

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k. \tag{5.28}$$

5.1.8 Jacobian matrices

Because of the system nonlinearity, the EKF covariance matrices cannot be directly estimated from $\mathbf{f}$ and $\mathbf{h}$, but only through their Jacobian matrices, $\mathbf{F}$ and $\mathbf{H}$. These are computed as

$$\mathbf{F} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} \boldsymbol{\Phi}_k & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\Delta t \mathrm{cpm}(\mathbf{q}_{1,k}) \\ \mathbf{0}_{3\times3} & \boldsymbol{\Phi}_k & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & -\Delta t \mathrm{cpm}(\mathbf{q}_{2,k}) \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \boldsymbol{\Phi}_k & \mathbf{0}_{3\times3} & -\Delta t \mathrm{cpm}(\mathbf{q}_{3,k}) \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{1}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & \Delta t \mathbf{1}_{3\times3} & \mathbf{1}_{3\times3} \end{bmatrix}, \tag{5.29}$$

$$\mathbf{H} = \frac{\partial \mathbf{h}}{\partial \mathbf{x}} = \begin{bmatrix} -\mathbf{A}_P \mathbf{G} & \mathbf{A}_T & \mathbf{A}_C (\partial \boldsymbol{\xi} / \partial \boldsymbol{\omega}) \\ \mathbf{0}_{3\times9} & \mathbf{0}_{3\times3} & \mathbf{1}_{3\times3} \\ \mathbf{R}^T_{\mathcal{BO}} \mathbf{M} & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \partial \mathbf{s} / \partial \mathbf{q} & \mathbf{0}_{6\times3} & \mathbf{0}_{6\times3} \end{bmatrix}, \tag{5.30}$$

$$\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\omega}} = \begin{bmatrix} 2\mathrm{diag}(\boldsymbol{\omega}_k) & \mathrm{sm}(\boldsymbol{\omega}_k) \end{bmatrix}^T, \tag{5.31}$$

$$\frac{\partial \mathbf{s}}{\partial \mathbf{q}} = \begin{bmatrix} 2\mathbf{q}_1 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{q}_3 & \mathbf{q}_2 \\ \mathbf{0}_3 & 2\mathbf{q}_2 & \mathbf{0}_3 & \mathbf{q}_3 & \mathbf{0}_3 & \mathbf{q}_1 \\ \mathbf{0}_3 & \mathbf{0}_3 & 2\mathbf{q}_3 & \mathbf{q}_2 & \mathbf{q}_1 & \mathbf{0}_3 \end{bmatrix}^T, \tag{5.32}$$

where sm stands for “special matrix”. [2]

5.1.9 Covariances of the errors

In addition, the EKF method requires the covariances of the process and measurement noises. The covariance $\mathbf{V}_k$ of the process noise is computed as

$$\mathbf{V} = E\left(\mathbf{u}_k \mathbf{u}_k^T\right) = \begin{bmatrix} \mathbf{0}_{9\times9} & \mathbf{0}_{9\times3} & \mathbf{0}_{9\times3} \\ \mathbf{0}_{3\times9} & \Delta t^2 \sigma_\gamma^2 \mathbf{1}_{3\times3} & (1/2)\Delta t^3 \sigma_\gamma^2 \mathbf{1}_{3\times3} \\ \mathbf{0}_{3\times9} & (1/2)\Delta t^3 \sigma_\gamma^2 \mathbf{1}_{3\times3} & (1/4)\Delta t^4 \sigma_\gamma^2 \mathbf{1}_{3\times3} \end{bmatrix}, \tag{5.33}$$

[2] The special matrix of vector $\mathbf{x} = [x_1 \; x_2 \; x_3]^T \in \mathbb{R}^3$ is defined as $\mathrm{sm}\left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}\right) \equiv \begin{bmatrix} 0 & x_3 & x_2 \\ x_3 & 0 & x_1 \\ x_2 & x_1 & 0 \end{bmatrix}$.

whereas the covariance matrices of the measurement noise $\mathbf{R}_k$ is

$$\begin{aligned}
\mathbf{R} &= E(\mathbf{v}_k \mathbf{v}_k^T) \\
&= \begin{bmatrix} E\left((\mathbf{A}_P \ddot{\mathbf{b}} + \delta \mathbf{a})(\mathbf{A}_P \ddot{\mathbf{b}} + \delta \mathbf{a})^T\right) & \mathbf{0}_{m\times 3} & \mathbf{0}_{m\times 3} & \mathbf{0}_{m\times 6} \\ \mathbf{0}_{3\times m} & E(\delta\boldsymbol{\omega}\delta\boldsymbol{\omega}^T) & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 6} \\ \mathbf{0}_{3\times m} & \mathbf{0}_{3\times 3} & E(\delta\mathbf{m}\delta\mathbf{m}^T) & \mathbf{0}_{3\times 6} \\ \mathbf{0}_{6\times m} & \mathbf{0}_{6\times 3} & \mathbf{0}_{6\times 3} & E(\delta\mathbf{s}\delta\mathbf{s}^T) \end{bmatrix} \qquad (5.34) \\
&= \begin{bmatrix} \sigma_b^2 \mathbf{A}_P \mathbf{A}_P^T + \sigma_a^2 \mathbf{1}_{m\times m} & \mathbf{0}_{m\times 3} & \mathbf{0}_{m\times 3} & \mathbf{0}_{m\times 6} \\ \mathbf{0}_{3\times m} & \sigma_\omega^2 \mathbf{1}_{3\times 3} & \mathbf{0}_{3\times 3} & \mathbf{0}_{3\times 6} \\ \mathbf{0}_{3\times m} & \mathbf{0}_{3\times 3} & \sigma_m^2 \mathbf{1}_{3\times 3} & \mathbf{0}_{3\times 6} \\ \mathbf{0}_{6\times m} & \mathbf{0}_{6\times 3} & \mathbf{0}_{6\times 3} & \sigma_s^2 \mathbf{1}_{6\times 6} \end{bmatrix}.
\end{aligned}$$

5.1.10 Tilt sensor method

For comparison, the two methods, tilt-sensor and integration, are also tested. As mentioned in the introduction of this thesis, an accelerometer is considered as a tilt sensor when the inertial acceleration is sufficiently small. Thence, for the tilt-sensor method, if we look back at equations (5.18) and (5.25), the covariance σ_b^2 of inertial acceleration $\ddot{\mathbf{b}}$ is set to zero in the algorithm, whether or not there is a hand motion, which gives

$$\sigma_b^2 = 0. \qquad (5.35)$$

Meanwhile, the covariance σ_ω^2 of measurement noise $\delta\omega$ is set

$$\sigma_\omega^2 \to \infty, \qquad (5.36)$$

which indicates that the gyroscope is not considered at all in this method.

5.1.11 Integration method

For the integration method, through inspecting eq. (5.9), the orientation (rotation matrix) can be estimated by an integration of the cross-product matrix of the angular velocity $\boldsymbol{\omega}$. Therefore, this method relies on both the gyroscope and the accelerometers, from which angular velocity can be estimated [82]. In the algorithm of the EKF, looking at equations (5.24) and (5.25), we need to set the covariances σ_a^2 and σ_m^2 of measurement noise δa and δm to infinity, which makes the algorithm completely rely on the state equation. That is,

$$\sigma_a^2 \equiv E(\delta a \delta a^T) \to \infty \qquad (5.37)$$

and

$$\sigma_m^2 \equiv E(\delta m \delta m^T) \to \infty. \qquad (5.38)$$

Hence, both of these two methods (tilt sensor and integration methods) are implemented as particular cases of what has been developed in Section 5.1.7.

5.1.12 Method of Marins et al. [62]

As mentioned in the introduction of this thesis, lots of researches were done trying to avoid the problems given by integration method and tilt sensor method. In [62, 63], firstly, acceleration and local magnetic field measurements are used as input to the Gauss-Newton iteration algorithm or Quest algorithm to find the optimal quaternion. As we find that the Quest algorithm [62, 64, 106, 107] has the disadvantage of sometimes requiring special computations to avoid singular cases and of being less accurate and less robust [108], while the Gauss-Newton iteration is simple and succinct, and can converge in 3 or 4 steps in most cases, we finally choose the method proposed in [62] for comparison, which is summarized in Fig. 5.1.

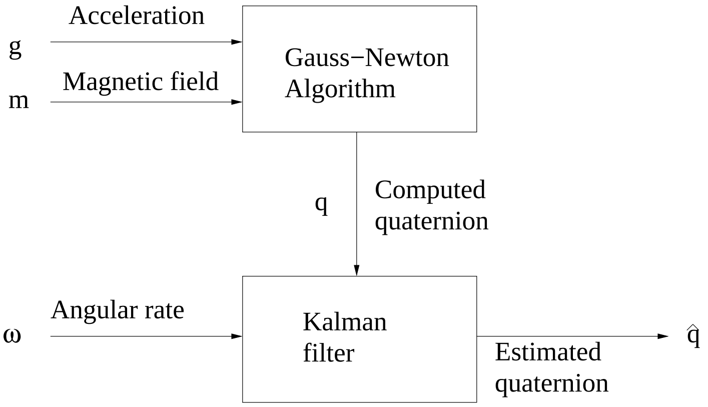


Figure 5.1: Structure of method of Marins et al.

Let us first define the quaternion $\mathbf{q}_M$,

$$\mathbf{q}_M = \begin{bmatrix} \mathbf{r}_M \\ r_0 \end{bmatrix}, \tag{5.39}$$

where $\mathbf{r}_M$ and r_0 (scalar component) are the four components of quaternion $\mathbf{q}_M$. Then, the rotation matrix $\mathbf{Q}_M$ can be expressed as a function of the quaternion, namely,

$$\mathbf{Q}_M = (r_0^2 - \mathbf{r}_M^T \mathbf{r}_M)\mathbf{1}_{3\times 3} + 2\mathbf{r}_M \mathbf{r}_M^T + 2r_0 \mathbf{R}_M, \tag{5.40}$$

where $\mathbf{R}_M$ is the cross product matrix of $\mathbf{r}_M$.

Let us define e_{GN} the error in Gauss-Newton method, then we have

$$e_{GN} = \epsilon_{\mathcal{F}}^T \epsilon_{\mathcal{F}} = (\mathbf{y}_{\mathcal{F}} - \mathbf{M}_{GN}\mathbf{y}_{\mathcal{O}})^T (\mathbf{y}_{\mathcal{F}} - \mathbf{M}_{GN}\mathbf{y}_{\mathcal{O}}), \tag{5.41}$$

where, $\mathbf{y}_{\mathcal{F}}$ is a 6×1 vector with values of gravity and magnetic field in the fixed frame $\mathcal{F}$, $\mathbf{y}_{\mathcal{O}}$ is a 6×1 vector with the measurements of gravity and magnetic field in the body frame

$\mathcal{O}$ defined by Optitrack system, and $\mathbf{M}_{GN} = \begin{bmatrix} \mathbf{Q}_M & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{Q}_M \end{bmatrix}$. Therefore, the objective is to find iteratively the values of quaternion components that yield the minimum error e_{GNmin}.

For the Gauss-Newton method, the computed optimal quaternion components are given by

$$\mathbf{q}_{opt}(k+1) = \mathbf{q}_{opt}(k) - \left[\mathbf{J}(\mathbf{q}_{opt}(k))^T \mathbf{J}(\mathbf{q}_{opt}(k))\right]^{-1} \mathbf{J}(\mathbf{q}_{opt}(k))^T \epsilon_{\mathcal{F}}(\mathbf{q}_{opt}(k)), \tag{5.42}$$

where $\mathbf{q}_{opt}$ is a vector with the four components of the quaternion and $\mathbf{J}$ is the Jacobian matrix, which is defined as

$$\mathbf{J} = -\left[\begin{array}{cccc} \left(\frac{\partial \mathbf{M}_{GN}}{\partial qx}\mathbf{y}_{\mathcal{O}}\right) & \left(\frac{\partial \mathbf{M}_{GN}}{\partial qy}\mathbf{y}_{\mathcal{O}}\right) & \left(\frac{\partial \mathbf{M}_{GN}}{\partial qz}\mathbf{y}_{\mathcal{O}}\right) & \left(\frac{\partial \mathbf{M}_{GN}}{\partial qw}\mathbf{y}_{\mathcal{O}}\right) \end{array} \right]. \tag{5.43}$$

After that, the optimal quaternion is used as part of the measurement for the Kalman filter, in which the other part of measurement is angular velocities from gyroscope. For the state equation, let us define $\underline{\mathbf{x}}$ a vector, with the first three components being the angular velocity $\underline{\boldsymbol{\omega}}$, and the last four components being the quaternion $\mathbf{q}_{opt}$, namely,

$$\underline{\mathbf{x}} = \begin{bmatrix} \underline{\boldsymbol{\omega}} \\ \mathbf{q}_{opt} \end{bmatrix} \tag{5.44}$$

Then, we have the state equations

$$\begin{bmatrix} \underline{\dot{x}}_1 \\ \underline{\dot{x}}_2 \\ \underline{\dot{x}}_3 \end{bmatrix} = \frac{1}{\tau}\left(- \begin{bmatrix} \underline{x}_1 \\ \underline{x}_2 \\ \underline{x}_3 \end{bmatrix} + \begin{bmatrix} \underline{\omega}_1 \\ \underline{\omega}_2 \\ \underline{\omega}_3 \end{bmatrix} \right), \tag{5.45}$$

$$\begin{bmatrix} \underline{\dot{x}}_4 \\ \underline{\dot{x}}_5 \\ \underline{\dot{x}}_6 \\ \underline{\dot{x}}_7 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \underline{x}_4 \\ \underline{x}_5 \\ \underline{x}_6 \\ \underline{x}_7 \end{bmatrix} \otimes \begin{bmatrix} 0 \\ \underline{x}_1 \\ \underline{x}_2 \\ \underline{x}_3 \end{bmatrix}, \tag{5.46}$$

where $\otimes$ represents quaternion multiplication. With the help of $\mathbf{q}_{opt}$ from eq. (5.42), the output equation will be

$$\underline{z}_i = \underline{x}_i + \underline{v}_i, \qquad i = 1, \cdots, 7 \tag{5.47}$$

where $\underline{v}_i$ is the white noise measurement.

5.2 Inertial measurement units

We illustrate the effectiveness of our approach by using two different IMUs, a game-controller and the Octahedral Constellation of Twelve Accelerometers (OCTA) [82]. Both of these IMUs are designed and assembled so that they can not only be simulated but also tested in experiments.

For the game-controller, this model has been described in detail in Section 4.1 in Chapter 3. All the parameters using in the simulation and experiments are the same as those presented there.

For the OCTA, six biaxial ADXL320 accelerometers [84] from *Analog Devices* are mounted on a rigid body to form an accelerometer array called OCTA [82], as shown in Fig. 2.5. As a matter of fact, we use unreservedly the same model as that in Section 2.3. Thence, the geometry of OCTA is shown in Fig. 2.7.

Likewise, the nominal sensitive directions and positions of accelerometers are the same as those presented there. In order to estimate the azimuth angle, a triaxial magnetometer MAG3110 [109] from Freescale is fixed on the OCTA, which will be described in detail in Section 5.4.

5.3 Validation through the simulation of a target-pointing trajectory

We illustrate the effectiveness of our approach by two examples: the simulation of a game-controller application and the motion tracking of a real accelerometer array.

5.3.1 Target-pointing trajectory

In order to test the proposed method, a target-pointing application is simulated, inspired from motion-tracking problems found in video games. The user holds a game controller and points at targets 10 meters away. The game-controller direction is parameterized by the pitch (θ) and azimuth (ψ) angles, as shown in Fig. 5.2. Therefore, the rotation matrix taking $\mathcal{F}$ onto $\mathcal{B}$ is

$$\mathbf{Q} = \begin{bmatrix} \cos\psi & -\cos\theta\sin\psi & -\sin\theta\sin\psi \\ -\sin\psi & \cos\theta\cos\psi & -\sin\theta\cos\psi \\ 0 & \sin\theta & \cos\theta \end{bmatrix}. \tag{5.48}$$

We define that

$$\mathbf{e} = [-\cos\theta\sin\psi \quad \cos\theta\cos\psi \quad \sin\theta]^T, \tag{5.49}$$

$$\mathbf{p} = \lambda\mathbf{e} = [x \quad y \quad z]^T = [y_0\tan\psi \; y_0 \; y_0\tan\theta\sec\psi]^T, \tag{5.50}$$

where $\mathbf{e}$ is the vector pointing from the game-controller $\mathbf{G}$ to the target $\mathbf{P}$ and $\mathbf{p}$ is the position of the target. Meanwhile, as mentioned above, we assume the distance y between the game controller and targets to be of 10 m, which leads to $y_0 = 10$.

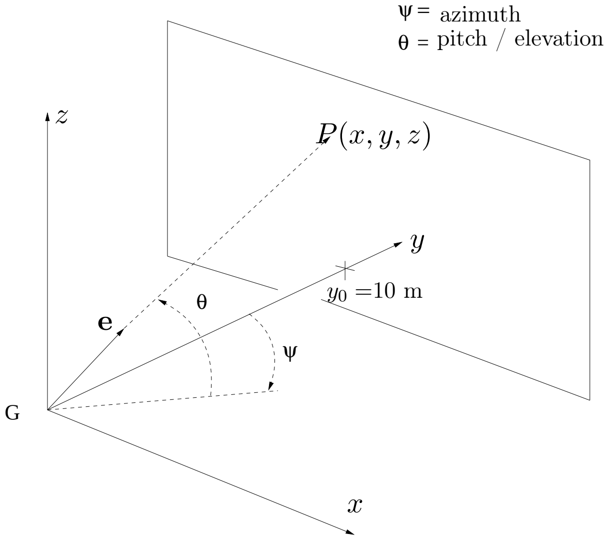


Figure 5.2: Game controller and target

The simulated measurements are computed from the true trajectories of the game controller, by the addition of Gaussian white noise as a measurement error. For the calculation in Matlab, we set the standard deviation of each accelerometer error to $\sigma_a = 0.1g$ (3% of the amplitudes of the values we expect to measure) and those of the gyroscope and the magnetometer errors to $\sigma_\omega = 0.1$ rad/s and $\sigma_m = 0.01\ \mu$T (2% and 1% of the amplitudes of the values we expect to measure), respectively. Meanwhile, the gravitational acceleration vector is $\mathbf{g} = [0\ 0\ \ -g]^T$ and the sampling time is $\Delta t = 0.01$ s.

5.3.2 Target-pointing trajectory generation

For the simplicity of the simulation, only a few targets are first generated randomly, as seen in Fig. 5.3. In this figure, the dotted line represents the limits of the two angles θ and ψ, mentioned in Section 5.3.1, where $-\pi/4 \le \theta, \psi \le \pi/4$.

Additionally, in the simulation, the rotation angle ψ for each time step is described by a fifth-order polynomial, namely,

$$\varphi = \overline{\varphi}(6\tau^5 - 15\tau^4 + 10\tau^3), \tag{5.51}$$

where $\overline{\varphi}$ is the rotation angle from one target to the next, $\tau = t/T$, $t = 0, \Delta t, 2\Delta t, \ldots, T$, Δt is the sampling time and T is the total time from one target to the next. This polynomial is from the robot trajectory planning [88, 110]. The order five is not arbitrary. A polynomial of such an order has six coefficients, if we include the zero-order term it contains. We can

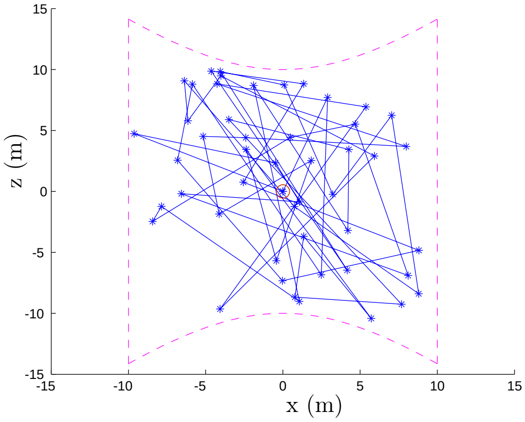


Figure 5.3: Targets and trajectory

choose those coefficients to simultaneously satisfy six constraints. This allows us to satisfy continuity conditions of orders zero, one and two at both limits of the interval over which the polynomial is used, which makes for six constraints. The order of the polynomial was thus chosen to satisfy constraints on orientation, angular velocity and angular acceleration. The relationship between φ and τ is shown in Fig. 5.4, where $\overline{\varphi}$ is chosen randomly.

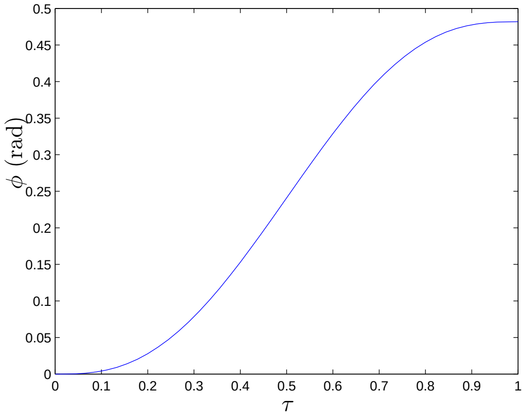


Figure 5.4: Relationship between φ and τ

In order to compute the angle $\overline{\varphi}$, the unit vector $\widetilde{\mathbf{e}}$ of rotation axis, which is normal to the plane constituted by the two target points and game-controller point, is first obtained. Finally, the rotation matrix for each time step [88] is

$$\widetilde{\mathbf{Q}} = \widetilde{\mathbf{e}}\widetilde{\mathbf{e}}^T + \cos(\varphi)(\mathbf{1}_{3\times 3} - \widetilde{\mathbf{e}}\widetilde{\mathbf{e}}^T) + (\mathrm{cpm}(\widetilde{\mathbf{e}}))^T \sin(\varphi). \tag{5.52}$$

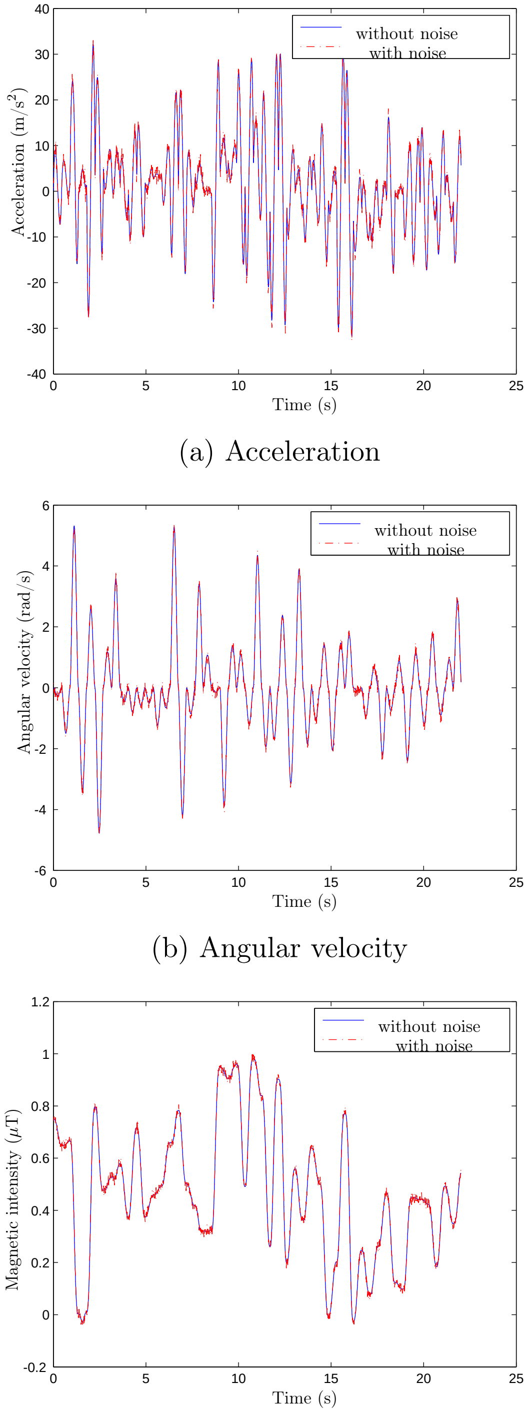
Then, the first derivative of the rotation matrix is computed for the cross-product matrix $\widetilde{\Omega}$

of the angular velocity. We have

$$\dot{\widetilde{\mathbf{Q}}}_k = (\widetilde{\mathbf{Q}}_{k+1} - \widetilde{\mathbf{Q}}_{k-1})/(2\Delta t), \tag{5.53}$$

$$\widetilde{\Omega}_k = \widetilde{\mathbf{Q}}_k \dot{\widetilde{\mathbf{Q}}}_k^T. \tag{5.54}$$

Meanwhile, in the simulation, we add some noises $\delta\mathbf{a}$, $\delta\boldsymbol{\omega}$ and $\delta\mathbf{m}$, which are from equations (5.16), (5.20) and (5.22), to the accelerometers, gyroscopes and magnetometer, respectively (Fig. 5.5(a)–Fig. 5.5(c)).

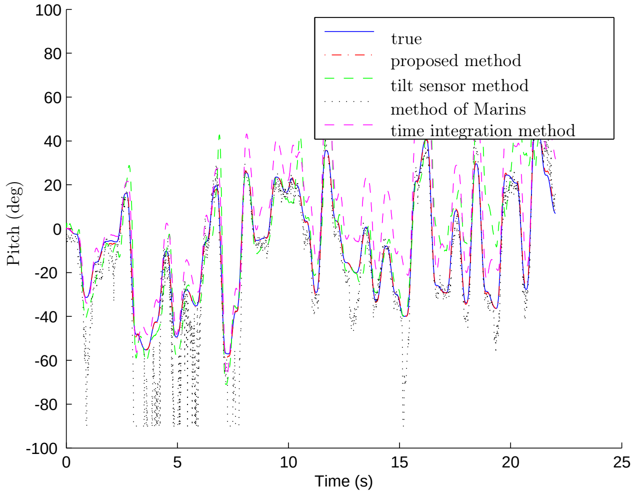


(a) Acceleration

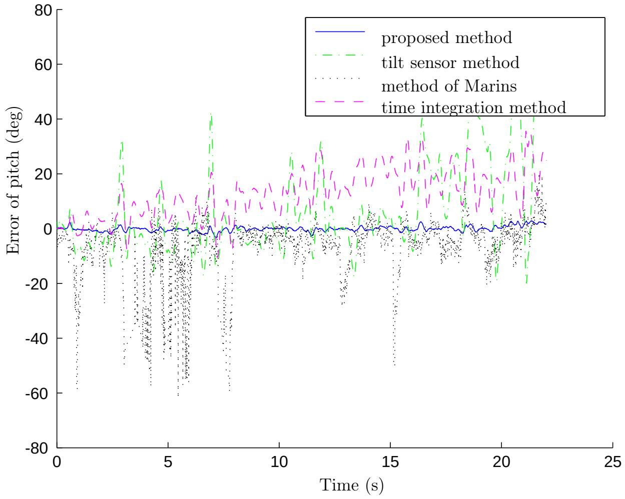
(b) Angular velocity

(c) Magnetic intensity

Figure 5.5: Acceleration, angular velocity and magnetic intensity with and without noise



(a) Pitch angle

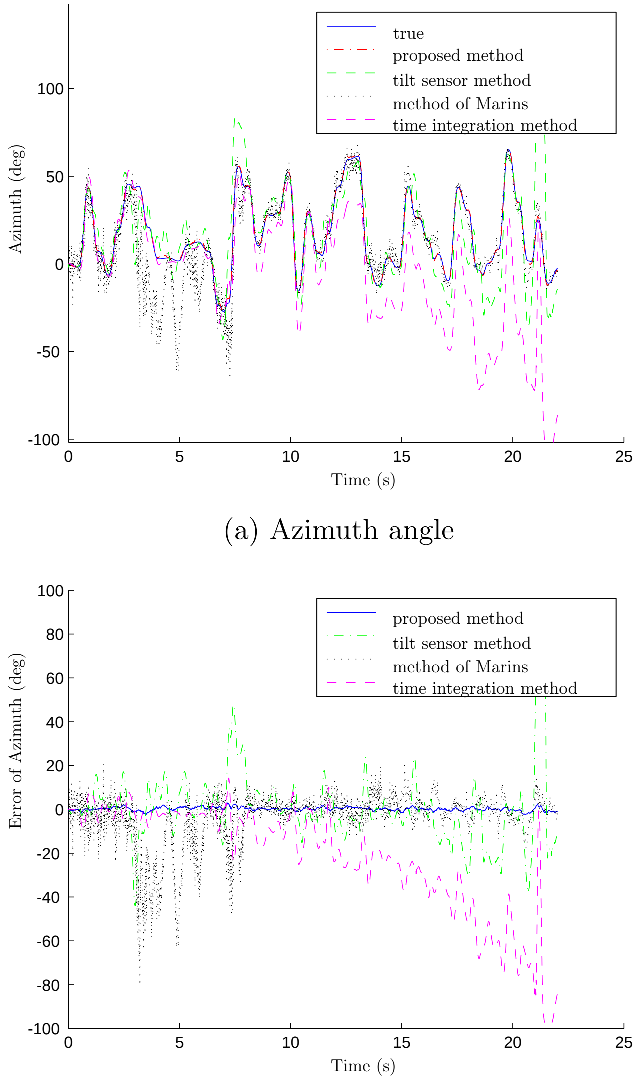


(b) Error of pitch angle

Figure 5.6: Estimate of pitch angle and its errors

5.3.3 Simulation results of the target-pointing trajectory with the game controller

Figures 5.6 and 5.7, which present typical results, show us estimates of the pitch and azimuth angles from different methods. We can see that the estimate from the proposed method matches well the truth. Fig. 5.8 gives us the true and the estimated trajectories. Meanwhile, it is evident that both the estimates from the method of Marins and the tilt sensor method have large errors, while the integration method exhibits important drift problems.



(a) Azimuth angle

(b) Error of azimuth angle

Figure 5.7: Estimate of azimuth angle and its errors

5.3.4 Simulation results of OCTA model

After simulation for game controller, we did simulation for OCTA as well. Because of the same target-pointing model, the only difference is the model parameters, mentioned in Section 5.2. Therefore, similar to Fig. 5.6 and Fig. 5.7, Fig. 5.9 and Fig. 5.10 are obtained. It is evident that the pitch and azimuth angles are well estimated from OCTA model, similar to that from the game-controller model. The estimated trajectory is obtained as well, as shown in Fig. 5.11. It is noted that the true trajectory is different from that in Fig. 5.8, as the simulated trajectory data is created randomly in Matlab. Moreover, the estimates from the method of Marins and the tilt sensor method have large erratic errors, while those from time integration method

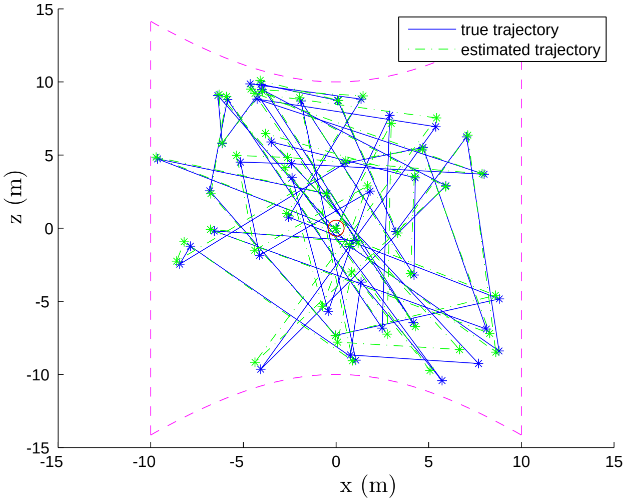


Figure 5.8: Targets and trajectory

have the problem of drift.

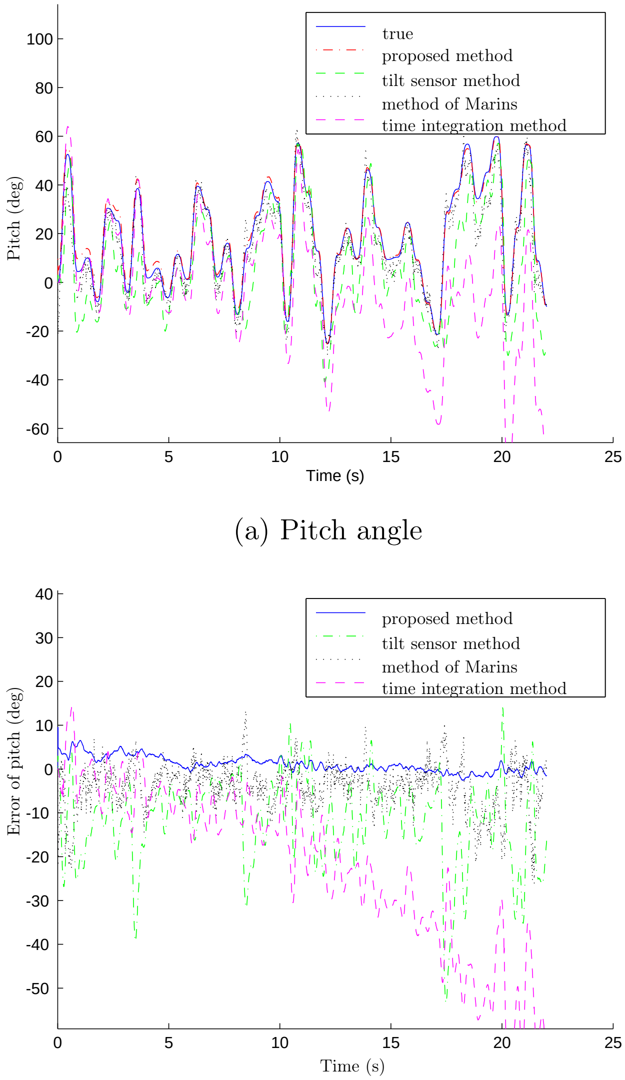
5.4 Experimental validation through an arbitrary hand-generated motion

We validated the proposed method by performing an experiment that reproduces the conditions in which the algorithm would work in real life. As the targeted application is a hand-held game controller, all the motions for our algorithm are generated by hand. While this type of validation has the merit of testing the algorithm close to its operating conditions, it also presents the drawback of assessing its performance in one given situation only. An alternative validation approach would consist in testing the method over a wide spectrum of trajectories, either through simulations or experiments. This task was not achieved here, however, as the goal is to demonstrate the usefulness of this method in its operating conditions [82].

We have exprimented with both the game-controller model and OCTA model. Except the different models, all other things necessary for the experiments are similar. Therefore, we will focus on OCTA while introducing the experimental testbed and preprocessing the measurements.

5.4.1 Experimental testbed

In the experimental validation, the Optitrack, a motion capture system from *Naturalpoint, Inc.*, is used in the experiments to provide reference measurement. Six cameras are fixed to



(a) Pitch angle

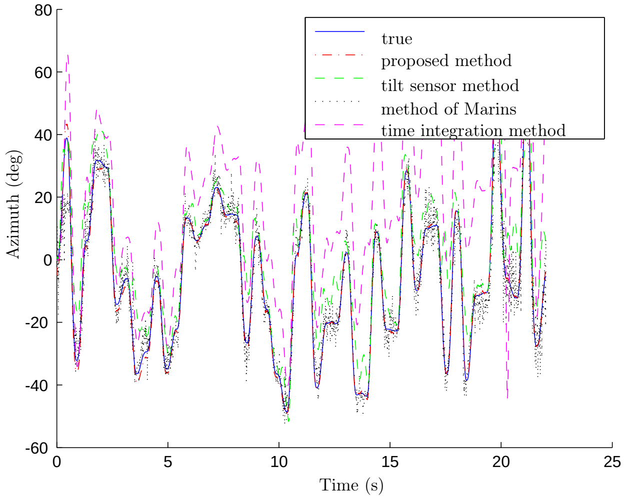
(b) Error of pitch angle

Figure 5.9: Estimate of pitch angle and its errors

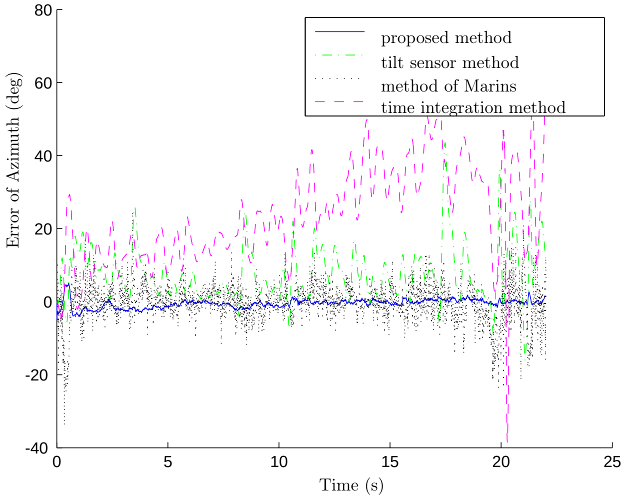
the ground to film five and seven markers that are attached on game controller and OCTA, respectively, as shown in Figures 5.12, 5.13 and 5.14.

The position precision of the Optitrack can reach to millimeter. Fig. 5.15 shows the relationship between the position and orientation error in a plane, where l is the length of the game controller, $\delta\theta$ is the orientation error and δy is the position error. The relationship between them is

$$\delta\theta = \arcsin\frac{2\delta y}{l}. \tag{5.55}$$



(a) Azimuth angle



(b) Error of azimuth angle

Figure 5.10: Estimate of azimuth angle and its errors

Recall that $\arcsin \frac{2\delta y}{l} \approx \frac{2\delta y}{l}$ when $\frac{2\delta y}{l} \to 0$. If $l = 0.3$ m and $\|\delta y\|_\infty \leq 1$ mm, then we have

$$-\frac{2\delta y}{l} \leq \delta\theta \leq \frac{2\delta y}{l}, \tag{5.56}$$

$$0.384^\circ \leq \delta\theta \leq 0.384^\circ. \tag{5.57}$$

This error is somewhat small enough. We can promote this relationship to the three-dimensional space. Therefore, Optitrack can be taken as a reference as well in the experiments.

Then, the *OptiTrackTrackables* block in Simulink can give the 6-DOF position and orientation of objects tracked by the OptiTrack camera system. The position and orientation are

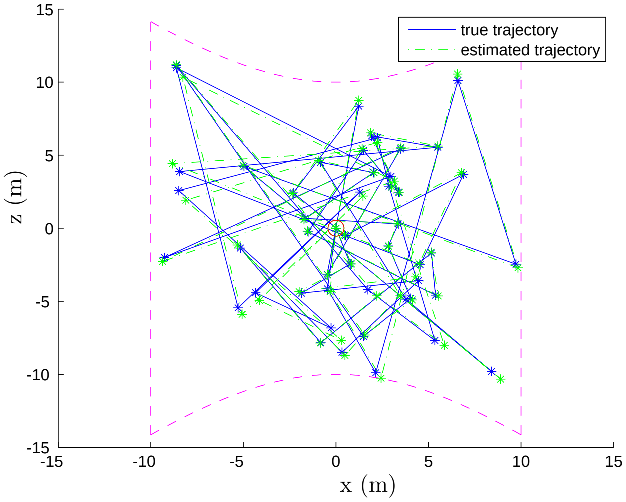


Figure 5.11: Targets and trajectory

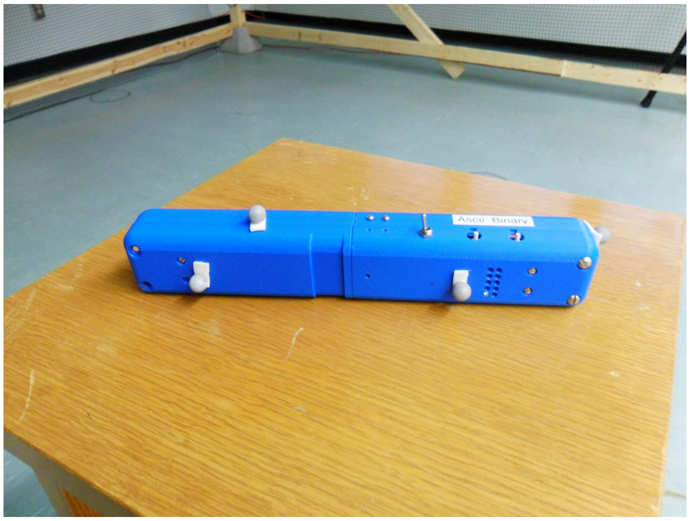


Figure 5.12: Markers on game controller

defined by a software called Tracking Tools [111], shown in Fig. 5.16. It is used for calibrating the cameras, defining the workspace coordinate frame, and defining trackables. Once calibrated, a calibration file (.cal) and a trackables definition file (.tra) must be saved to the computer and referenced in the Simulink block's parameters. The displacements are produced by shaking OCTA by hand [82].

The Optitrack, however, defines its own default frame $\mathcal{O}$, also attached to OCTA. This frame is unknown a priori, as it is automatically defined by the seven markers. Therefore, to allow a comparison between the reference Optitrack measurements and the accelerometer and

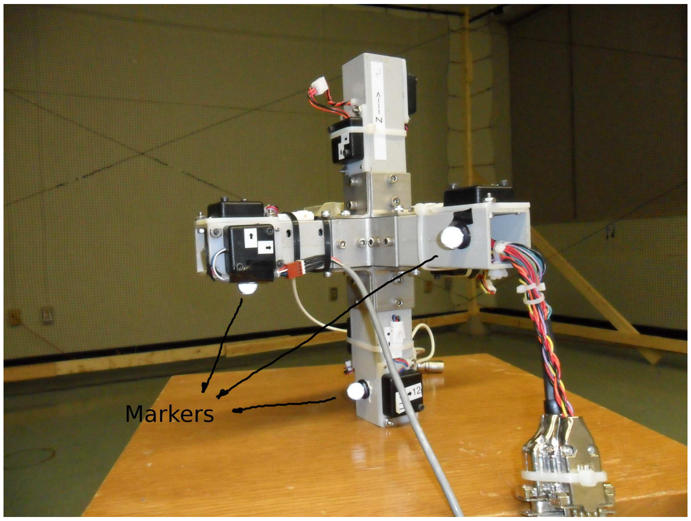


Figure 5.13: Markers on OCTA

magnetometer measurements, it is necessary to find $\mathbf{R}_{\mathcal{BO}}$, the rotation matrix transforming a vector from frame $\mathcal{O}$ to frame $\mathcal{B}$. The approach taken to identify this matrix is described in Appendix B.

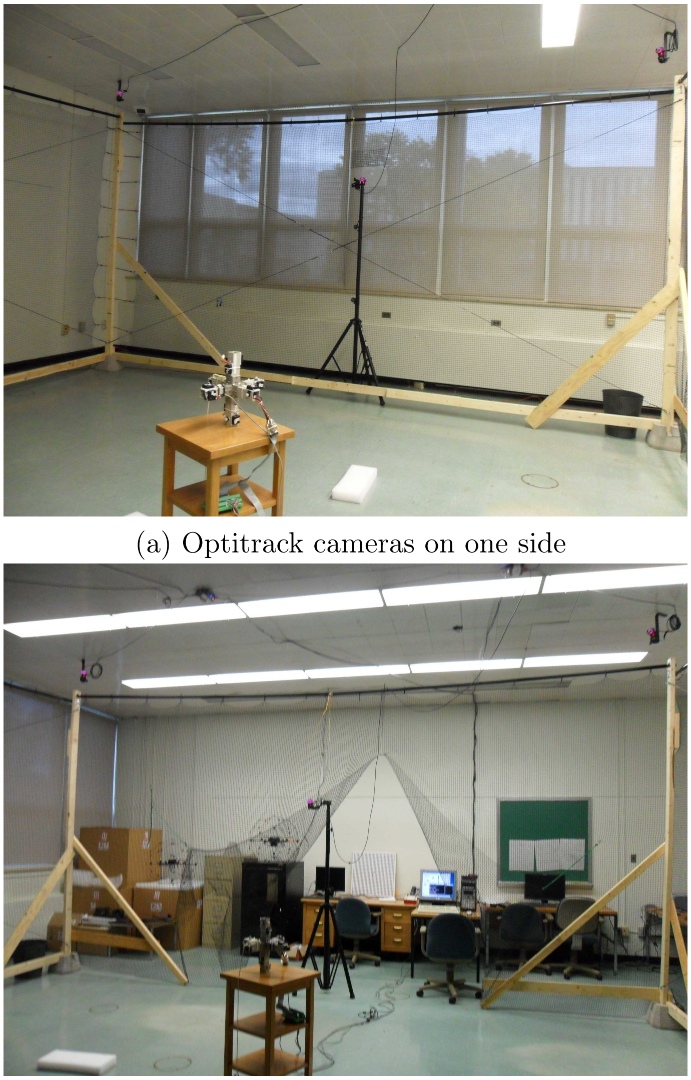
On the other hand, it is important to prepare all the hardware and software in order to receive the data from the sensors we have installed. For OCTA model, the digital sensor MAG3110 uses the I2C protocol, and thus a USB-I2C converter from *Robot Electronics*, shown in Fig. 5.17, is connected to it. A virtual serial COM port is thus created, where magnetometer outputs will be received. Totally, all data from sensors, including accelerometers and magnetometers, are obtained through Simulink and RT-Lab, which is the open real-time simulation software from $OPAL - RT\ Technologies$.

For the game-controller model, the Bluetooth [102] protocol was used for receiving data. Similarly, it creates also a virtual serial COM port. A Simulink model for collecting the measurements was designed as well.

5.4.2 Preprocessing the measurements from OCTA and the Optitrack

The main objective is to estimate the accuracy of the angular-velocity and orientation estimates computed from the proposed method. In this vein, the angular-velocity measurements are provided by the Optitrack.

The output voltages $v_k, k = 1, \ldots, n$, from the 12 accelerometers are acquired. From the



(a) Optitrack cameras on one side

(b) Optitrack cameras on another side

Figure 5.14: Optitrack cameras

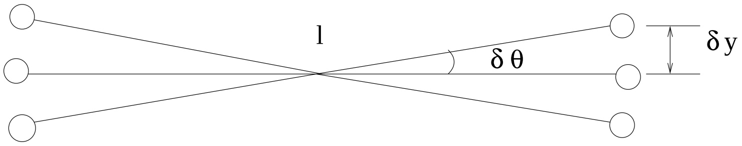


Figure 5.15: Relationship between position and orientation error

calibration laws [87], we compute the corresponding accelerations

$$a_{i,k} = \widetilde{\alpha}_i v_{i,k} + \widetilde{\beta}_i, \tag{5.58}$$

where, $a_{i,k}$ is the acceleration of the accelerometer i along direction $\mathbf{e}_i$ at time t_k, $\widetilde{\alpha}_i$ and $\widetilde{\beta}_i$ are the scale factor and bias for the ith accelerometer, and $v_{i,k}$ is its voltage output at time

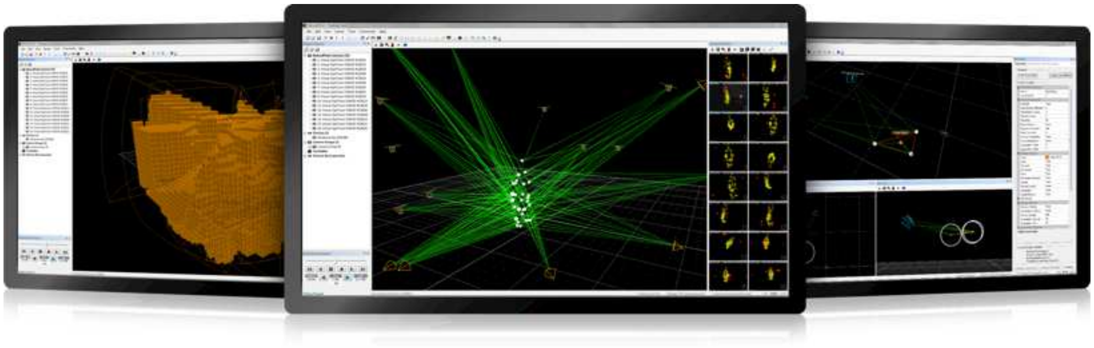


Figure 5.16: Tracking tools screens

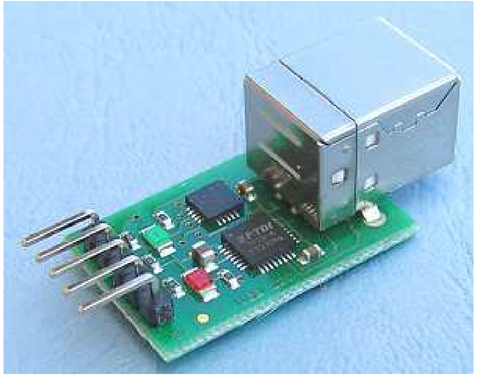


Figure 5.17: USB-I2C converter

t_k.

Meanwhile, we also need to determine the vector representing the true magnetic North and the rotation matrix $\mathbf{R}_{\mathcal{OM}}$ that takes the frame $\mathcal{O}$ onto a magnetometer's frame $\mathcal{M}$. This has been discussed in Section 4.4.

From the Optitrack, the orientation of OCTA was measured every 0.01 s, for periods of 100 s. These orientations were returned in the form of rotation matrices $[\mathbf{Q}_k]_{\mathcal{FO}}$ taking the fixed frame $\mathcal{F}$ onto the moving frame $\mathcal{O}$. According to the definition of $\mathbf{R}_{\mathcal{BO}}$ in Section 5.4.1, we have

$$\mathbf{Q}_k = [\mathbf{Q}_k]_{\mathcal{B}} = [\mathbf{Q}_k]_{\mathcal{FO}}\mathbf{R}_{\mathcal{BO}}^T, \tag{5.59}$$

which is the rotation matrix taking the fixed frame $\mathcal{F}$ onto the moving frame $\mathcal{B}$.

In order to compute the corresponding angular velocity $\boldsymbol{\omega}_k$, a first-order low-pass filter, with a cut-off frequency $f_c = 5$ Hz, was first applied over the entries of $\mathbf{Q}_k$. Also, to avoid the time shifts caused by the filter, we filtered the data in both forward and backward time, and retained the average. After filtering the data, a central difference approximation of the first derivative was used to compute $\dot{\mathbf{Q}}_k$, namely,

$$\dot{\mathbf{Q}}_k = (\mathbf{Q}_{k+1} - \mathbf{Q}_{k-1})/(2\Delta t). \tag{5.60}$$

The cross-product matrix $[\boldsymbol{\Omega}_k]_{\mathcal{B}}$ of the angular velocity $\boldsymbol{\omega}_k$ is computed from eq. (5.6) as

$$[\boldsymbol{\Omega}_k]_{\mathcal{B}} = \mathbf{Q}_k^T \dot{\mathbf{Q}}_k. \tag{5.61}$$

Because of measurement errors, this matrix is generally not antisymmetric as it should be. We retain only the skew-symmetric component of $\boldsymbol{\Omega}_k$ by resorting instead to the relation

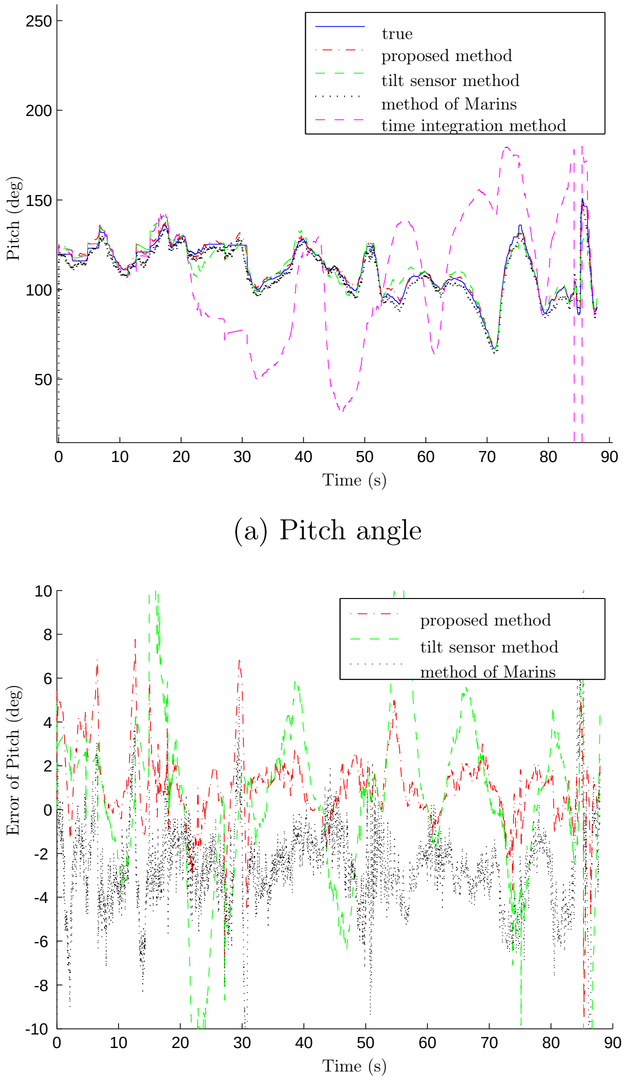
$$[\boldsymbol{\Omega}_k]_{\mathcal{B}} = (\mathbf{Q}_k^T \dot{\mathbf{Q}}_k - (\mathbf{Q}_k^T \dot{\mathbf{Q}}_k)^T)/2. \tag{5.62}$$

5.4.3 Experimental results of game-controller model

With the help of Optitrack, as the reference, the experimental results are shown in Figs. 5.18 and 5.19. It is evident that the estimated pitch and azimuth angles from the proposed method match well the true ones, while those from the tilt-sensor method have larger errors, e.g., at the time of 23 s and 87 s, in Fig. 5.18(a). The estimate errors from the method of Marins are between ±10°, but still larger than those from the proposed method, as seen in Fig. 5.19(b). For the integration method, the error got very large after just 20 seconds in Fig. 5.18(a) and Fig. 5.19(a). It is noted that, because of some external factors, e.g., the light or obstacles which may disturb the infrared cameras in the room, or marker misalignments during the movement, the reference attitude matrix from the Optitrack sometimes gave discontinuities, which slightly affected the judgement of accuracies of the methods. This truth, however, is somewhat within the acceptable range in experiments.

5.4.4 Experimental results of OCTA model

Similar to the game controller experiments, typical experimental results from the OCTA were obtained as well and are shown in Figures 5.20 and 5.21. The pitch and azimuth angles are well estimated by proposed method. Its errors are in the range from $-5°$ to $5°$, seen in Fig. 5.20(b), even better than that in Fig. 5.18(b). In addition, the estimate from the tilt sensor method has errors within ±10°, similar to that from the method of Marins, while the integration method, undoubtedly, has larger errors drifting away from the reference angle in just a few instants. It is worth noting that, for the two experiments, we moved the game controller and OCTA in space with different inertial accelerations so as to evaluate the tilt-sensor method more convincingly. The inertial acceleration in the game controller experiment is smaller than that in the OCTA experiment.



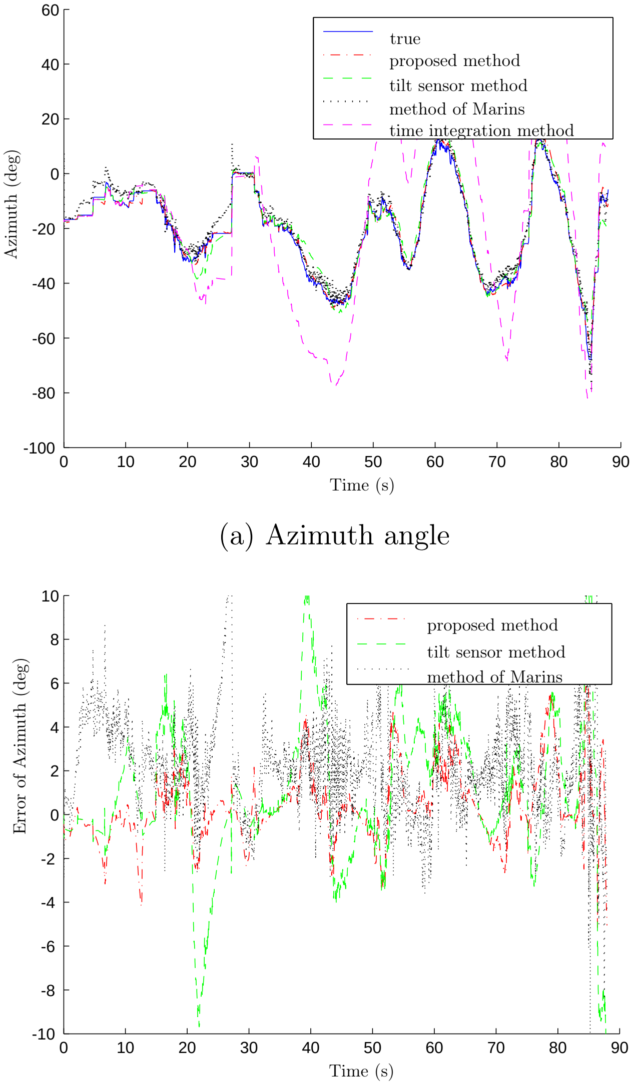
(a) Pitch angle

(b) Error of pitch angle

Figure 5.18: Estimate of pitch angle and its errors

5.5 Comparison and discussion

Firstly, to get basic ideas of the estimate results, let us pay attention to the different models, simulation and experimental results. Both the game-controller and OCTA are used in simulations and experiments. It is apparent that, for these two models, the errors of estimate results are similar to each other, not only in simulations, but also in experiments. Thus, it seems that the accelerometer-array geometry has little effect on the estimate accuracy. One should take care, however, to avoid singular accelerometer-array configurations, where a part of the angular velocity or angular acceleration becomes unobservable [90]. In a word, the output of the accelerometers should provide enough information of acceleration for **y** to be computed



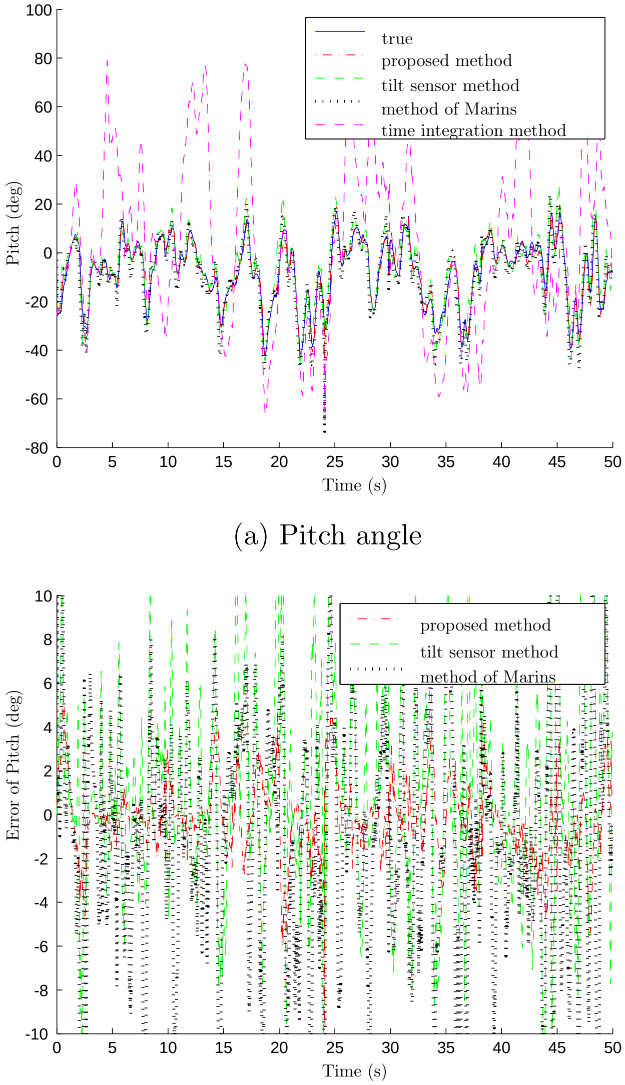
(a) Azimuth angle

(b) Error of azimuth angle

Figure 5.19: Estimate of azimuth angle and its errors

through eq. (5.25).

Moreover, it is noted that the errors of angle estimates in simulations are smaller than those from experiments. If seen in Fig. 5.6 and Fig. 5.7, the error is between $-3°$ and $3°$, while it is between $-10°$ and $10°$ in experiments, shown in Figs. 5.18, 5.19, 5.20 and 5.21. This seems logical, because within the acceptable range, some inevitable errors, like from camera captures, sensor misalignments, marker misalignments during the movement, and so on, exist in experiments, and affect the accuracy of the estimates, but were not accounted for in simulation. In fact, when combined with the symmetric marker arrangement, or because of



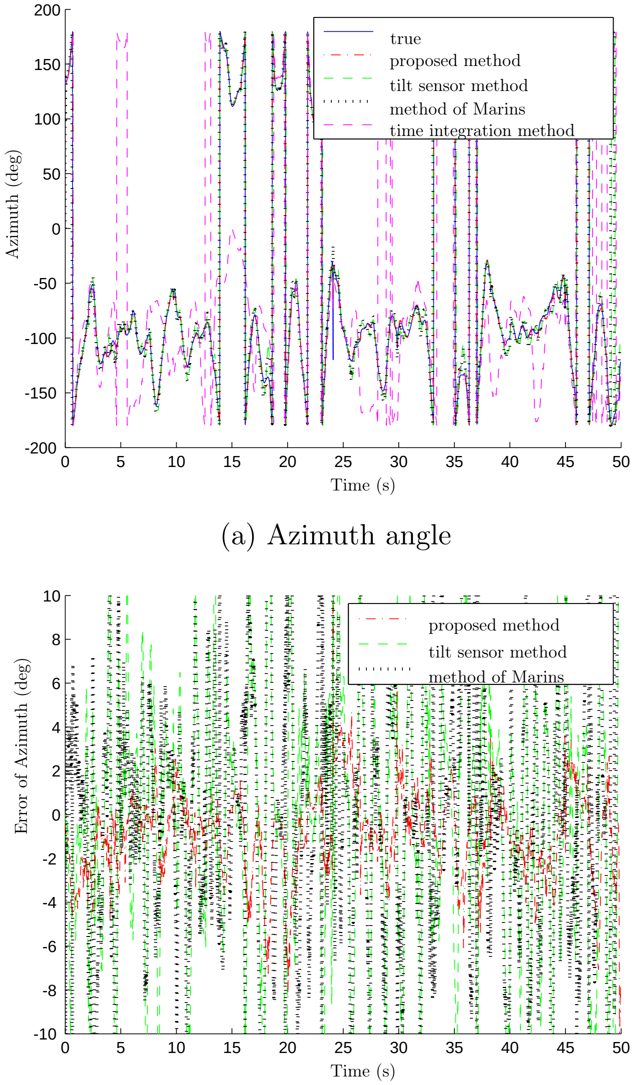
(a) Pitch angle

(b) Error of pitch angle

Figure 5.20: Estimate of pitch angle and its errors

the obstacles (for example, human body), the loss of a camera due to poor perspective of the markers can make the rigid body orientation flip very common. What we do is to do our best to avoid these. Finally, Table 5.1 gives us the basic information of the results.

Then, the different methods are compared and analysed in detail. From eq. (5.8), the rotation matrix can be estimated by integrating the angular velocity over time. The errors will accumulate, however, inevitably causing a drift, which prevents us from using the time-integration method alone for more than a few seconds.



(a) Azimuth angle

(b) Error of azimuth angle

Figure 5.21: Estimate of azimuth angle and its errors

Meanwhile, in the tilt-sensor method, the inertial acceleration is assumed to be null. In the event that the inertial acceleration is small, the orientation estimate is sufficiently accurate, shown in Figures 5.18 and 5.19, but its errors are still larger than those from the proposed method, while in the case of high accelerations, large errors are generated, as shown in Figures 5.20 and 5.21.

For our proposed method, both simulation and experiments show that the pitch angle can be estimated with a reasonable accuracy, independently from the intensity of the accelerations. This can be proved by Fig. 5.22, which is traced from experimental data of OCTA. In addition,

Table 5.1: Comparison of the model, simulation and experiment

IMU	orientation error in simulation (deg)	orientation error in experiments (deg)
Game-controller	-3—3	-10—10
OCTA	-3—3	-5—5
	similar	similar

a close observation of Fig. 5.22(*a*) reveals that the cloud formed by all the measured data points is confined within a cone with its apex at the origin. The mathematical relationship between the pitch angle, and the inertial and gravitational accelerations allows for an analytical determination of the aperture of this cone:

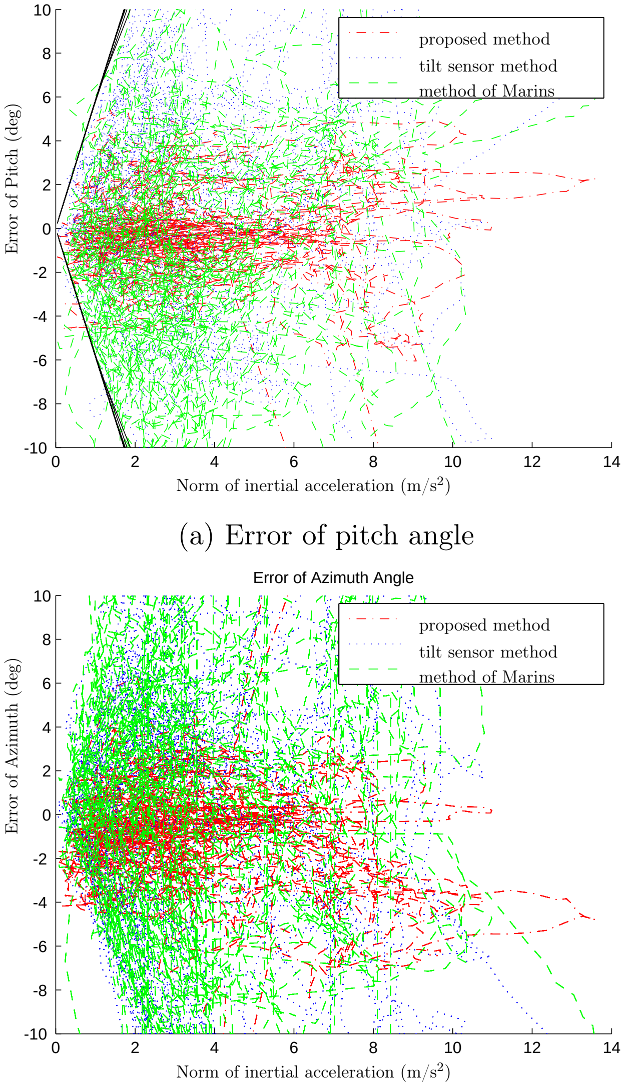
$$-\tan^{-1}\left(\frac{\|\ddot{\mathbf{b}}\|}{g}\right) \leq \delta\theta \leq \tan^{-1}\left(\frac{\|\ddot{\mathbf{b}}\|}{g}\right), \tag{5.63}$$

where $\ddot{\mathbf{b}}$ is the inertial acceleration, g is the gravitational acceleration and $\delta\theta$ is the error of pitch angle. Because the estimate of the azimuth angle is affected by that of the pitch angle, the azimuth-angle error exhibits a similar limit in Fig. 5.22(*b*).

As a matter of fact, inertial acceleration $\ddot{\mathbf{b}}$ is considered as a part of the measurement noise, by inspecting eq. (5.34). In the calculation process, the covariance of the inertial acceleration is required. The higher this covariance, the more the system relies on the state equation. It is easily understood that when the covariance is nearly zero, which denotes that the rigid body is stationary, the system relies mostly on the measurements and the accelerometers are taken as the tilt sensors.

In addition, it is found that the accuracy of the estimates from the game-controller (with gyroscope) are similar to those from OCTA (without gyroscope), as can be seen in Table 5.1. This indicates that one can do with or without gyroscope in some applications, especially those involving impacts, as mechanical gyroscopes are known to be more sensitive than accelerometers to these acceleration peaks. To be rigorous, a gyroscope is suggested to be used, however. We found that the accelerometers give still large errors of angular velocity estimates when at low angular velocities, which is a problem in several applications. Moreover, an accelerometer array requires more space than a single gyroscope. If space is not a problem, then one may find the ideal solution to be a combination of both.

For the estimate of the azimuth angle, we can see from Fig. 5.7 and Fig. 5.21, that there is no drift of errors, given by the magnetometer installed. Because the measurements from inertial



(a) Error of pitch angle

(b) Error of azimuth angle

Figure 5.22: Estimate of pitch and azimuth angle versus the norm of inertial acceleration

sensors only supply information of pitch and roll angles because of the gravitational acceleration, it is impossible to estimate the azimuth angle only by these inertial sensors. Mathematically, if we look back at eq. (5.25), where $-\mathbf{G}[\mathbf{q}_1^T\ \mathbf{q}_2^T\ \mathbf{q}_3^T]^T = -g\mathbf{q}_3$ and $\mathbf{q}_3 = \mathrm{f}(\theta, \phi)$ (θ and ϕ are the pitch and roll angle, respectively), the output equation does not include state variable for the azimuth angle. Hence, in this case (without magnetometer measurements),the state equation works alone, which comes to be the time-integration method. Therefore, another special sensor like magnetometer or optical sensor should be added to prevent the associated drift along that axis.

By using the same parameters and measurements, the method of Marins [62, 63] is compared with tilt-sensor method and our proposed method. The resulting pitch and azimuth angle estimates are shown in Figs. 5.6, 5.7, 5.9, 5.10, 5.18, 5.19, 5.20 and 5.21. It is evident, however, from Figs. 5.6, 5.7, 5.9 and 5.10, that the errors from the method of Marins are larger than those from the proposed method, even sometimes larger than those from the tilt-sensor method. We conjecture that this reduced accuracy of the method of Marins occurs because, as for the tilt-sensor method, it makes no distinction between inertial and gravitational accelerations [62, 63]. Although in the second step of the method, gyroscope measurements come to help estimate the orientation, it does not essentially solve this problem. In the cases of very high inertial accelerations and large Euler angles, the computed quaternions with large errors from the first step will still make the final results less precise. As a matter of fact, the inertial accelerations used in the simulation are larger than those in the experiments. Therefore, the method of Marins in the simulation performs worse than in the experiments. Meanwhile, one disadvantage of the Gauss-Newton method is that it may become lost with poor initial estimates. Moreover, as the final objective is to estimate the rotation matrix which represents the attitude of rigid body, it is better to avoid intermediate rotation parameters such as quaternions.

5.6 Application programming interface (API)

After developping the algorithms, it is necessary to test them in practice. In order to imitate the game controller Wii, we would like to create an application programming interface (API) so that the virtual game controller can follow our designed true game controller. We built a Matlab GUI connecting with Simulink to create a simple virtual environment for the game controller.

We firstly used Simulink to receive data of all sensors through the virtual serial port created by Bluetooth. Then, we connected the Simulink with the Matlab graphical user interface (GUI). Our algorithm was embedded in the program so that the virtual game controller in GUI can be controlled by the outputs of sensors, caused by the movement of the real game controller. In order to make things clearer, we summarize the system by four components, as seen in Fig. 5.23.

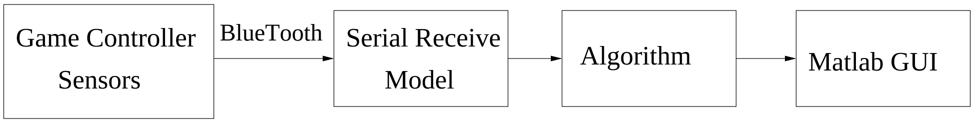


Figure 5.23: Simulink model of interface

When the user moves the game controller in space, the Simulink model receives the output of the game controller sensors. The orientation is then estimated from the measurements by using the algorithm presented in Section 5.1. Finally, the virtual game controller is traced in the GUI, as seen in Fig. 5.24, according to the estimated orientation.

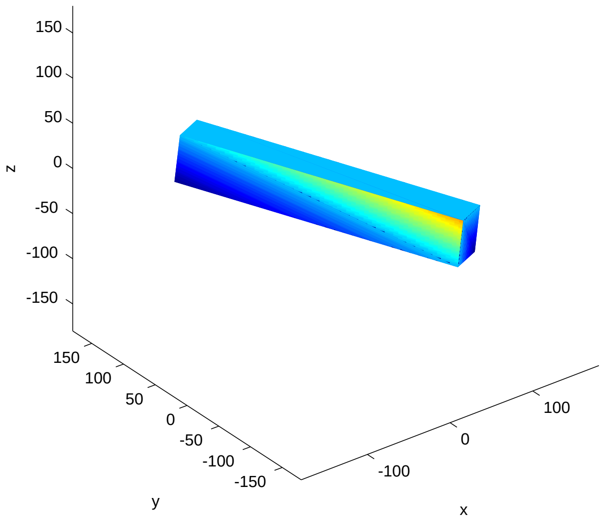


Figure 5.24: Virtual game controller

The results of testing the virtual gaming environment seemed good when we handled the game controller in front of a computer screen. The virtual game controller can follow approximately every movement of the true one. There were some delays, however, which are caused by the communication, not algorithms. This will be ignored, as our interface is kind of qualitative appreciation, as opposed to quantitative. In addition, it is difficult to evaluate the precision of the movement estimation. Because the views in the computer (with $-37.5°$ of azimuth and $30°$ of pitch) is not the same as that in reality. Hence, it can be misleading and give the impression that there is a great error in reality.

Moreover, as the proposed algorithm is designed to estimate the orientation, the position of a point of the game controller is not reliably estimated. The method used consists in the integration of the point acceleration, which generates drift. As a result, if the game controller is still on a table, the virtual game controller moves slowly. Mathematically, the point position is computed from the basic kinematic equations

$$\dot{\mathbf{b}} = \mathbf{v}, \tag{5.64}$$

$$\dot{\mathbf{v}} = \ddot{\mathbf{b}}, \tag{5.65}$$

where, $\mathbf{b}$ and $\mathbf{v}$ are the point position and velocity, respectively. Because of measurement errors, the error accumulates after integration. In order to eliminate the accumulated error and artificially stabilize the virtual game controller, we add a tuning term, which is somewhat larger than the error, for each step. Therefore, eqs. (5.64) and (5.65) become,

$$\dot{\mathbf{b}} = \mathbf{v} - K_{d,b}\mathbf{b}, \tag{5.66}$$

$$\dot{\mathbf{v}} = \ddot{\mathbf{b}} - K_{d,v}\mathbf{v}, \tag{5.67}$$

where, $K_{d,b}$ and $K_{d,v}$ are tuning parameters, with the unit s^{-1}. With this ad hoc correction method, the visible position error in the interface appears to be very subtle to the user.

5.7 Conclusion

Estimating the orientation of a rigid body moving in space is of high importance in game playing. In this chapter, a new algorithm using extended Kalman filter to estimate the orientation of a rigid body moving in space is proposed, validated both by simulation and experimental results. The main novelty of the proposed approach lies in the consideration of the inertial acceleration as a part of measurement noise. Both a rigid-body kinematics model and a stochastic model of the human-hand motion are formulated and combined in a nonlinear state-space system, where the components of the rotation matrix are state variables.

Unlike the time-integration method generating the drift, the proposed one is shown to be stable. This can explain well why the sensor magnetometer is used in the experiments. The time-integration method is unstable, because it does not use the fixed external reference for estimating the orientation. This fixed external reference can be the Earth's gravitational field or the magnetic field. The gravitational field, however, is not sufficient as reference for all three axes of rotation. Therefore, we need the magnetic field as well, measured by the magnetometer. On the other hand, considering the inertial acceleration as white noise proved to yield more accurate results than neglecting it altogether, as is generally done with tilt sensors or with other existing methods [62]. In other words, it works efficiently with both high and low accelerations. Meanwhile, it is noted that, compared with the method of Marins [62] that uses two stages and does not explicitly distinguish the inertial and gravitational accelerations, the proposed method, using a simpler structure, i.e., a single Kalman filter applied on a state-space system, is more stable and accurate.

Additionally, we found that the accuracy of the estimates from the game-controller (with gyroscope) are similar to those from OCTA (without gyroscope). Therefore, in some applications, especially those involving impacts, one can do with or without gyroscope. In other words, the proposed method is independent of the gyroscope. On the contrary, one important point is to consider the geometry of the device, as an accelerometer array requires more space than a single gyroscope. In this case, to be rigorous, a gyroscope is suggested to be used, as we have done for the game controller in the thesis. In reality, one may find a best way to combine both if space is not a problem.

In conclusion, the extended Kalman filter designed in this paper will be useful in applications, such as game controllers and human-machine interfaces in general. In addition to inertial sensors, another sensor like magnetometer or optical sensor should be used as the output measurement for estimating the azimuth angle, as it is not related to the gravitational acceleration.

It is of much importance to test and demonstrate the proposed methods not only by using the estimate accuracy, but also by performing in human-machine interface, which is one of the final applications of them. The smartphones, game controller Wii, and so on, are the examples of this. Therefore, we have developped an application programming interface using the Matlab GUI so that the virtual game controller appearing on screen can follow the orientation of the real one.

Conclusion

Every end is a new beginning.

Summary and contributions of the thesis

The products of technology, media and telecom have had a rapid development in the last decade. In particular, the mass-market of video game controllers, which includes the Wiimote and other gaming interfaces are inexpensive and easily available. A drawback, however, is that the precision in the motion detection hardware leaves a lot to be desired for many gaming applications, let alone high-precision industrial or scientific applications [1]. Therefore, in this research, we have proposed and developed algorithms to estimate more accurately the orientation of a game controller moving in space.

In order to better estimate the orientation of a rigid-body moving in space, we first focused on estimating its time derivative, the angular velocity. Gyroscopes are standardly used to measure the angular velocity. MEMS gyroscopes have been reported unreliable under high acceleration by some authors, however. Hence, we investigated the possibility of replacing them with low-cost accelerometers in different locations of the game controller. From the kinematics equations, the minimum number of accelerometers required to estimate the angular velocity of a rigid body in space without resorting to the angular acceleration is nine. Therefore, an accelerometer-array called OCTA was designed, which will be used for both angular velocity and orientation estimation. To estimate the angular velocity, we combined the two basic methods available, which are time-integration (TI) and polynomial-roots (PR) methods, using their weighted sum. This proposed method has been published in the Journal of dynamic systems, measurement and control [82]. We chose the weighting factors to minimize the variance of estimated angular velocity. Experiments were done to evaluate this proposed method, in which a magnetic motion tracking sensor was used as a reference. The proposed method was shown to yield stable results, unlike the TI method which has the problem of drift. Meanwhile, it solves the sign problem, which affects other stable methods like the PR method and the method of Parsa et al. [38]. Moreover, the PR method and that of Parsa et al. are less accurate at low angular velocities than the proposed method, which shows better robustness.

Having devised a method for angular velocity estimation, we turned our attention to orientation estimation. This second phase of the research appeared more challenging than the first, so we set ourselves the intermediary goal of estimating the orientation in the vertical plane. This work has been presented by me in the conference called IDETC2012 [89]. In the research, a game controller model similar to the Wiimote was designed. Then, three new algorithms using EKF and UKF were proposed and compared with two existing basic methods, which are the time-integration method and the gravitational method. Any of these two methods cannot be used alone because of its inherent problems: drift and error from inertial acceleration, respectively. Moreover, because of the high-sensitivity at low angular velocities, the proposed complete-pose method, which is to estimate the complete rigid-body pose by using EKF, only works occasionally. After comparison and analysis of the proposed and some representative existing methods in the simulations, it is found that our proposed orientation-only methods (EKF and UKF) are stable and give excellent orientation estimates, error of which is within ±10°, both at low and high angular velocities. This indicates that its accuracy is independent from the trajectory, and is promising for the generalization of the method to spatial orientations.

To benchmark the proposed estimation method, we designed and created a true game controller, which comprises two triaxial accelerometers, one triaxial gyroscope and one triaxial magnetometer. The end goal was to use our proposed method in a simple virtual environment, to compare its response to that of real hand-held game controllers. Before doing that, all the sensors were calibrated. A high-precision rate table with one DOF was used for the calibration of the accelerometers and the gyroscope. For the accelerometers, the scale factors and offsets were well obtained compared to the specification in the datasheet, although the relative error is somewhat larger than expected. Moreover, it was found that there is no noticeable hysteresis in the response of this low-cost sensor. Similarly, for the gyroscopes, the scale factors and offsets were obtained, close to the estimated values. Furthermore, two more tests (different position test and back and forth test) were performed to find that the output is robust, as the difference between them is small relative to the calibration errors. In summary, both the accelerometers and gyroscopes were observed to be fairly insensitive to cross-axis accelerations, to hysteresis and exhibited good linearity. Meanwhile, the tracking system Optitrack was used for the calibration of the magnetometer. After scale factor and offset compensation, the magnetometer measurements lie well on the surface of a centered unit sphere, instead of an ellipsoid. Then, the true magnetic North and the rotation matrix from the body-fixed frame to the magnetometer frame were well estimated, as the root mean square value in the calculation was found small enough, relative to the Earth's magnetic field.

The final objective is to estimate the orientation in 3D world. In our research, a new algorithm using the extended Kalman filter was proposed, which has been published in the Multbody system dynamics [104]. In order to evaluate it, we have used both the game controller and OCTA. Similarly, compared with the basic time integration method and tilt-sensor method, the proposed method avoids the drift problem and error given by tilt-sensor method. Furthermore, we have implemented the method of Marins et al. [62], with the same initial parameters. Unlike the method of Marins using two steps (Gauss-Newton iteration and Kalman filter), our proposed method uses a simpler structure, i.e., a single Kalman filter applied on a state-space system. Meanwhile, instead of estimating the intermediate parameters, like quaternions, the proposed method directly estimates the rotation matrix. Lastly, the method of Marins and some other similar methods still do not attempt at distinguishing the inertial acceleration and the gravitational acceleration. This makes the final estimates have the unpredictable errors or noises from the rigid body motions in space. In this case, our proposed method presents stable and accurate results (error of which within $\pm 3^\circ$ in simulation and $\pm 10^\circ$ in experiments), both in simulation and in experiments. Then, the virtual human-machine environment, i.e., real-time experiment, was created using the Matlab GUI. It is found that the virtual game controller can follow the true one approximately, which seems to be good enough for the game controller applications.

In summary, our research proposes new algorithms to estimate the orientation of the game controller moving in space. Compared with some existing basic methods and other methods given by predecessors, both in simulation and experiments (offline and online), our new algorithms are shown to be stable and accurate. We believe that the proposed methods can be widely used to measure hand motion, e.g., in game controllers, robot surgery [112], vehicles and aircraft.

Future work

Real-time tracking of rigid body motion in space has been playing a more and more important and critical role in the human-machine interaction applications, such as artificial intelligent (AI) robotics, military unmanned aircraft, vehicles and game controllers. In this thesis, we proposed new algorithms to estimate the orientation of the rigid body moving in space. Some further work, however, needs to be done to improve these estimation methods.

Firstly, the better the sensors, the more accurate the measurements. For improving the estimating results, there is no doubt that choosing better sensors is one of the most natural ways. In mass-market products, however, we must limit ourselves to low-cost sensors. Hence, the problem of producing accurate sensors at a low cost is still of importance. On the other hand, keeping using these inertial sensors, we could try to improve the proposed algorithms. Because in the algorithm, we estimated coarsely the probability of moving from one pose to the next within one time step. In order to accurately obtain knowledge of inertial acceleration covariance, we may analyse deeply the statistical properties of the hand motions. Furthermore, because of the use of EKF (or UKF), we could do more research on the sensor errors to verify whether they are well represented by Gaussian white noise. More detailed information on this could be analysed instead of simply using the specification of the sensors.

Secondly, in order to estimate the azimuth angle, a magnetometer was used in our research. This sensor, however, is sensitive to environmental magnetic interference outside of the handheld device, e.g., wifi, mobile phones, metallic objects and magnetic tracking devices. Therefore, an alternative sensor should be sought for azimuth angle estimation. Unfortunately, we did not find any better sensor. We could probably use LEDs and fix the reference point, as Wiimote is doing.

Thirdly, as the main work is on the orientation estimation, there are still large noises on the position estimates. Although in the virtual environment, we did add parameterized component in the basic kinematic equations for tuning to eliminate the accumulated error caused by integration. It is, however, not a rigorous way. Therefore, we could add some other position capturing sensors like cameras. In order to merge this into our extended Kalman filter, it would be interesting to try to include the camera measurements as a part of the outputs, like the other sensors.

Furthermore, for the sake of higher accuracy, robustness, stability, and position estimation, Simultaneous Localization and Mapping (SLAM) [113] is suggested to be used. SLAM is a technique to build up a map of an unknown environment or to update a map within a known environment, while at the same time keeping track of the current location of the observer. Surely, the SLAM still has some problems, e.g., computational complexity, nonlinearity, data association and landmark characterization. During the past years, a lot of researchers have tried to solve these problems. Meanwhile, recent research on SLAM has primarily focused on developing algorithms that can be used in real-time implementation as well [114, 115, 116]. Finally, we need to find a way to combine the SLAM with our proposed methods so that both the orientation and the position can be estimated accurately.

Last but not least, as the research aims at the game controller product, more work on software design for human machine interface is needed to be conducted. In this thesis, user's interface is designed only to validate the proposed algorithms for one time once, which is not sufficient and artificial. We may add some more buttons or choices, e.g., *stop* and *continue*. *Edittext* can be used to adjust the variance of inertial acceleration, or measurement noises. Meanwhile, it is a good idea to embed some existed databases of human hand motions into the software. In this case, the interface would become artistic.

Bibliography

[1] C. Smith and H. I. Christensen. Wiimote robot control using human motion models. In Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS), pages 5509–5515, 2009.

[2] P. Cardou, G. Fournier, and P. Gagnon. A nonlinear program for angular-velocity estimation from centripetal-acceleration measurements. IEEE/ASME Transactions on Mechatronics, 16(5):932–944, 2011.

[3] C. C. Chou and S. C. Sinha. On the kinematics of the head using linear acceleration measurements. J. Biomech., 69(3):607–613, 1976.

[4] N. K. Mital and A. I. King. Computation of rigid-body rotation in three-dimensional space from body-fixed linear acceleration measurements. ASME Journal of Applied Mechanics, 46:925–930, 1979.

[5] G. S. Nusholtz, J. Wu, and P. Kaiker. Passenger air-bag study using geometric analysis of rigid-body motion. Experimental Mechanics, 31(3):264–270, 1991.

[6] R. W. G. Anderson, C. J. Brown, P. C. Blumbergs, A. J. McLean, and N. R. Jones. Impact mechanics and axonal injury in a sheep model. Journal of Neurotrauma, 20(10):961–974, 2003.

[7] R. T. Shea and D. C. Viano. Computing body segment trajectories in the hybrid iii dummy using linear accelerometer data. ASME Journal of Biomechanical Engineering, 116:37–43, 1994.

[8] F. Luan, K. H. Yang, B. Deng, P. C. Begeman, S. Tashman, and A. I. King. Qualitative analysis of neck kinematics during low-speed rear-end impact. Clinical Biomechanics, 15(9):649–657, 2000.

[9] E. J. Ohlmeyer and T. R. Pepitone. Guidance, navigation and control without gyros: A gun-launched munition concept. In Proceedings of AIAA Guidance, Navigation, and Control Conference and Exhibit, pages 2336–2349, Monterey, CA, 2002.

[10] W. C. Pickel. Estimation of postlaunch angular motion for kinetic energy projectiles. Journal of Guidance, Control, and Dynamics, 28(4):604–610, 2005.

[11] J. R. W. Morris. Accelerometry—A technique for the measurement of human body movements. Journal of Biomechanics, 6:729–736, 1973.

[12] W. C. Hayes, J. D. Gran, M. L. Nagurka, J. M. Feldman, and C. Oatis. Leg motion analysis during gait by multiaxial accelerometry: Theoretical foundations and preliminary validations. ASME Journal of Biomechanial Engineering, 105:283–289, 1983.

[13] D. Giansanti and G. Maccioni. Comparison of three different kinematic sensor assemblies for locimotion studies. Physiol. Meas., 26(5):689–705, 2005.

[14] A. Grammatikos. Gimballess Inertial Systems for Space Navigation. PhD thesis, University of Pennsylvania, PA, 1965.

[15] A. J. Padgaonkar, K. W. Krieger, and A. I. King. Measurement of angular acceleration of a rigid body using linear accelerometers. ASME. J. Appl. Mech., 42(E3):552–556, 1975.

[16] R. H. Huijser and J. P. B. Vreeburg. Assembly of linear accelerometers mounted on a rigid body remote from the center of mass measuring three dimensional movements, 02 1993.

[17] J.-H. Chen, S.-C. Lee, and D. B. DeBra. Gyroscope free strapdown inertial measurement unit by six linear accelerometers. AIAA Journal of Guidance, Control, and Dynamics, 17(2):286–290, 1994.

[18] J. Genin, J. Hong, and W. Xu. Accelerometer placement for angular velocity determination. ASME Journal of Dynamic Systems, Measurement and Control, 119:474–477, 1997.

[19] K. Parsa. Dynamics, State Estimation, and Control of Manipulators with Rigid and Flexible Subsystems. PhD thesis, McGill University, Montreal, Canada, March 2003.

[20] K. Parsa, T. A. Lasky, and B. Ravani. Design and mechatronic implementation of an accelerometer-based, kinematically redundant inertial measurement unit. In IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Monterey, California, USA, 2005.

[21] A. Linder, M. Svensson, and D. Viano. Evaluation of the bioRID P3 and the hybrid iii in pendulum impacts to the back: A comparison with human subject test data. Traffic Inj. Prev., 3:159–244166, 2002.

[22] P. Cappa, L. Masia, and F. Patane. Numerical validation of linear accelerometer systems for the measurement of head knematics. ASME Journal of Biomechanical Engineering, 127:919–928, 2005.

[23] Z. Qin, L. Baron, and L. Birglen. Robust design of inertial measurement units based on accelerometers. Journal of Dynamic Systems, Measurement, and Control, 2009.

[24] B.M̃. Nigg and W. Herzog. Biomechanics of the Musculo-skeletal System. University of Calgary, Calgary, Alberta, Canada, third edition, 2007.

[25] M. C. Algrain and J. Quinn. Accelerometer based line-of-sight stabilization approach for pointing and tracking systems. In Proceedings of Second IEEE Conference on Control Applications, pages 159–163, Vancouver, Canada, 1993.

[26] K. Parsa, J. Angeles, and A. K. Misra. Estimation of the flexural states of a macro-micro manipulator using point-acceleration data. IEEE Trans. Robot., 21(4):565–573, 2005.

[27] P.-C. Lin, H. Komsuoḡlu, and D.E. Koditschek. Sensor data fusion for body state estimation in a hexapod robot with dynamical gaits. IEEE Trans. Robot., 22(5):932–375, 2006.

[28] M. D. Miles. Measurement of six-degree of freedom model motions using strapdown accelerometers. In 21st American Towing Tank Conference, pages 369–375, Washington DC, 1986.

[29] V. A. Subramanian and C. P. Vendhan. An efficient algorithm for strapdown accelerometer-based motion measurement. Ocean Eng., 20(4):421–432, 1993.

[30] K. S. Mostov, A. A. Soloviev, and T. K. J. Koo. Accelerometer based gyro-free multi-sensor generic inertial device for automotive applications. In Proceedings of ITSC, IEEE Conference on Intelligent Transportation Systems, pages 1047–1052, Boston, MA, 1997.

[31] Y. K. Peng and M. F. Golnaraghi. A vector-based gyro-free iniertial navigation system by integrating existing accelerometer network in a passenger vehicle. In IEEE Position Location and Navigation Symposium, pages 234–242, Monterey, CA, 2004.

[32] Nicolas Franceschini Franck Ruffier. Optic flow regulation: the key to aircraft automatic guidance. Robotics and Autonomous Systems, 50:177–194, 2005.

[33] V. Sankaramarayaman, M. E. Emekli, B. A. Güvenç, L. Güvenç, E. S. Öztürk, Ş. S. Ersolmaz, I. E. Eyol, and M. Sinal. Semiactive suspension control of a light commercial vehicle. IEEE/ASME Transactions on Mechatronics, 13(5):598–604, 2008.

[34] E. Edwan, S. Knedilik, J. Zhou, and O. Loffeld. GPS/INS integration for GF-IMU of twelve mono-axial accelerometers configurations. In Proceedings of The 6th Workshop

on Positioning, Navigation and Communication 2009 (WPNC'09), pages 183–188, Hannover, Germany, 2009.

[35] K. D.-Ulhaas. Optical Tracking From User Motion To 3D Interaction. PhD thesis, Vienna Unversity of Technology, Vienna, Germany, 2002.

[36] A. R. Schuler. Design and Analysis of Analytic Platform Inertial Navigation Systems. PhD thesis, University of Pennsylvania, PA, 1965.

[37] M. Costello and T. Jitpraphai. Determining angular velocity and angular acceleration of projectiles using triaxial acceleration measurements. J. Spacecr. Rockets, 39(1):73–80, 2002.

[38] K. Parsa, T. A. Lasky, and B. Ravani. Design and implementation of a mechatronic all-accelerometer inertial measurement unit. IEEE/ASME Transactions on Mechatronics, 12(6):640–650, 2007.

[39] P. Cardou and J. Angeles. Estimating the angular velocity of a rigid body moving in the plane from tangential and centripetal acceleration measurements. Multibody Syst Dyn, 2007.

[40] M. B. Ignagni. Optimal strapdown attitude integration algorithms. Journal of Guidance, 12:363–369, 1990.

[41] H. J. Luinge and P. H. Veltink. Measuring orientation of human body segments using miniature gyroscopes and accelerometers. Medical & Biological Engineering & Computing, 43:273–282, 2005.

[42] H. Dejnabadi, B. M. Jolles, E. Casanova, and P. Fua. Estimation and visualization of sagittal kinematics of lower limbs orientation using body-fixed sensors. IEEE Transactions on Biomedical Engineering, 53(7):1385–1393, 2006.

[43] H. Ren and P. Kazanzides. Investigation of attitude tracking using an integrated inertial and magnetic navigation system for handheld surgical instrument. IEEE/ASME Trans. on Mechatronics, 17(2):210–217, 2012.

[44] E. A. M. Filho, H. K. Kuga, and A. R. Neto. Integrated GPS/INS navigation system based on a gyroscope-free IMU. In Brazilian Conference on Dynamicc, Control and Their Applications DINCON, pages 183–188, Guaratinguetá, SP, Brazil, 2006.

[45] K. Parsa, J. Angeles, and A. K. Misra. Attitude calibration of an accelerometer array. In Proceedings of the IEEE International Conference on Robotics and Automation, volume 1, pages 129–134, Washington, DC, 2002.

[46] H. J. Luinge and P. H. Veltink. Inclination measurement of human movement using a 3D accelerometer with autocalibration. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 12(1):112–121, 2004.

[47] E. Bernmark and C. Wiktorin. A triaxial accelerometer for measuring arm movements. Appl. Ergon., 33:541–547, 2002.

[48] G. A. Hansson, P. Asterland, N. G. Holmer, and S. Skerfving. Validity and reliability of triaxial accelerometers for inclinometry in posture analysis. Med. Biol. Eng. Comput., 39:405–413, 2001.

[49] B. Kemp, A. J. M. W. Janssen, and B. Van Der Kamp. Body position can be monitored in 3D using miniature accelerometers and earth-magnetic field sensors. Electroenceph. Clin. Neurophysiol./ Electromyogr. Motor Control, 109:484–488, 1998.

[50] J. Lötters, J. Bomer, T. Verloop, E. Droog, W. Olthuis, P. H. Veltink, and P. Bergveld. In-use calibration procedure for a triaxial accelerometer. Sensors Actuators A, Phys., 66:205–212, 1998.

[51] A. T. M. Willemsen, C. Frigo, and H. B. K. Boom. Lower extremity angle measurement with accelerometers—error and sensitivity analysis. IEEE Transactions on Biomedical Engineering, 38(12):1186–1193, 1991.

[52] A. M. Sabatini. Quaternion based extended Kalman filter for determining orientation by inertial and magnetic sensing. IEEE Trans. on Biomedical Engineering, 53(7):1346–1356, 2006.

[53] X. Yun, E. R. Bachmann, and R. B. McGhee. A simplified quaternion based algorithm for orientation estimation from earth gravity and magnetic field measurements. IEEE Trans. on Instrumentation and Measurement, 57(3):638–650, 2008.

[54] B. Huyghe and J. Doutreloigne. 3D orientation tracking based on unscented Kalman filtering of accelerometer and magnetometer data. In IEEE Sensors Applications Symposium, pages 148–152, New Orleans, LA, USA, 2009.

[55] K. Liu, T. Liu, K. Shibata, Y. Inoue, and R. Zheng. Novel approach to ambulatory assessment of human segmental orientation on a wearable sensor system. Journal of Biomechanics, 42:2747–2752, 2009.

[56] K. Liu, K. Shibata, Y. Inoue, and E. G. Cao. Ambulatory estimation of knee-joint kinematics in anatomical coordinate system using accelerometers and magnetometers. IEEE Transactions on Biomedical Engineering, 58(2):435–442, 2011.

[57] K. Liu, Y. Inoue, and K. Shibata. Physical sensor difference-based method and virtual sensor difference-based method for visual and quantitative estimation of lower

limb 3D gait posture using accelerometers and magnetometers. Computer Methods in Biomechanics and Biomedical Engineering, 15(2):203–210, 2012.

[58] K. J. O'Donovan, K. Roman, D. T. O'Keefe, and G. M. Lyons. An inertial and magnetic sensor based technique for joint angle measurement. Journal of Biomechanics, 40(12):2604–2611, 2007.

[59] V. Vikas and C. D. Crane III. Inclination parameter estimation for manipulator and humanoid robot links. In Proceedings of the ASME 2011 International Design Engineering Technical Conference & Computers and Information in Engineering Conference, Washington, DC, USA, Aug 2011.

[60] V. Vikas and C. D. Crane III. Robot inclination estimation using vestibular dynamic inclinometer. In IASTED International Conference Robotics and Applications, volume 706, ACTA Press, 2010.

[61] V. Vikas and C. D. Crane III. Inclination estimation and balance of robot using vestibular dynamic inclinometer. In IEEE-RAS International Conference on Humanoid Robots, pages 245–250, 2010.

[62] J. L. Marins, X. Yun, E. R. Bachmann, R. B. McGhee, and M. J. Zyda. An extended Kalman filter for quaternion-based orientation estimation using MARG sensors. In Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst., pages 2003–2011, Maui, HI, USA, Oct. 2001.

[63] X. Yun and E. R. Bachmann. Design, implementation, and experimental results of a quaternion based Kalman filter for human body motion tracking. IEEE Trans. on Robotics, 22(6):1216–1227, Dec. 2006.

[64] M. D. Shuster and S. D. Oh. Three-axis attitude determination for vector observations. J. Guid. Control, 4(1):70–77, 1981.

[65] J. L. Crassidis, F. L. Markley, and Y. Cheng. Survey of nonlinear attitude estimation methods. AIAA Journal of Guidance, Control and Dynamics, 30(1):12–28, 2007.

[66] E. Foxlin. Head-tracking relative to a moving vehicle or simulator platform using differential inertial sensors. In Proceedings of Helmet and Head-Mounted Displays V, SPIE Vol. 4021, AeroSense Symposium, Orlando FL, USA, 2000. DOI:10.1117/12.389141.

[67] T. Nieminen, J. Kangas, S. Suuriniemi, and L. Kettunen. An enhanced multi-position calibration method for consumer-grade inertial measurement units applied and tested. Measurement Science and Technology, 21(10):105204, 2010.

[68] STMicorelectronic Inc. *Using LSM303DLH for a tilt compensated electronic compass*, 2010. `www.st.com`.

[69] F. Ferraris, I. Gorini, U. Grimaldi, and M. Parvis. Calibration of three-axial rate gyros without angular velocity standards. Sensors and Actuators A: Physical, 42(1-3):446–449, 1994.

[70] P. Schopp, L. Klingbeil, C. Peters, and Y. Manoli. Design, geometry evaluation, and calibration of a gyroscope-free inertial measurement unit. Sensors and Actuators A: Physical, 162(2):379–387, 2010.

[71] P. Batista, C. Silvestre, P. Oliveira, and B. Cardeira. Accelerometer calibration and dynamic bias and gravity estimation: Analysis, design, and experimental evaluation. IEEE Transactions on Control Systems Technology, 19(5):1128–1137, 2011.

[72] A. Umeda, M. Onoe, K. Sakata, T. Fukushia, K. Kanari, H. Iioka, and T. Kobayashi. Calibration of three-axis accelerometers using a three-dimensional vibration generator and three laser interferometers. Sensors and Actuators A: Physical, 114(1):93–101, 2004.

[73] Z. F. Syed, P. Aggarwal, C. Goodall, X. Niu, and N. El-Sheimy. A new multi-position calibration method for MEMS inertial navigation systems. Measurement Science and Technology, 18(7):1897, 2007.

[74] J. C. Lötters, J. Schipper, P. H. Veltink, W. Olthuis, and P. Bergveld. Procedure for in-use calibration of triaxial accelerometers in medical applications. Sensors and Actuators A: Physical, 68(1-3):221–228, 1998.

[75] D. Lee, S. Lee, S. Park, and S. Ko. Test and error parameter estimation for MEMS-based low cost IMU calibration. International Journal of Precision Engineering and Manufacturing, 12(4):597–603, 2011.

[76] D. Unsal and K. Demirbas. Estimation of deterministic and stochastic IMU error parameters. In Position Location and Navigation Symposium (PLANS), IEEE/ION, pages 862–868, 2012.

[77] E. R. Bachmann, X. Yun, D. McKinney, R. B. McGhee, and M. J. Zyda. Design and implementation of MARG sensors for 3-DOF orientation measurement of rigid bodies. In Proceedings of the 2003 IEEE, International Conference on Robotics & Automation, pages 1171–1178, 2003.

[78] R. L. Needham. Calibration of strapdown system accelerometer dynamic errors. Master's thesis, Massachusetts Institute Technology, Cambridge, MA, USA, 1994.

[79] X. Han, S. Hu, and H. Luo. A new method for bias calibration of laser gyros using a single-axis turning table. Optik-International Journal for Light and Electron Optics, 124(22):5588–5590, 2013.

[80] M. Glueck, D. Oshinubi, and Y. Manoli. Automatic realtime offset calibration of gyroscopes. In Sensors Applications Symposium (SAS), IEEE, pages 214–218, 2013.

[81] M. Hwangbo and T. Kanade. Factorization-based calibration method for MEMS inertial measurement unit. In IEEE International Conference on Robotics and Automation, pages 1306–1311, 2008.

[82] P. He and P. Cardou. Estimating the angular velocity from body-fixed accelerometers. J. Dyn. Sys., Meas., Control, 134(6):061015–1–061015–10, Nov. 2012.

[83] J. E. Horvath, E. A. Logiudice, C. Riveros, and H. Vucetich. Einstein equivalence principle and theories of gravitation: A gravitationally modified standard model. Phys. Rev. D, 38:1754–1760, 1988.

[84] Analog Devices. ADXRS300 Datasheet. One Technology Way, P.O. Box 9106, Norwood, MA, USA, 2004. www.analog.com.

[85] G. T. Schmidt. INS/GPS technology trends. In Military Capabilities Enabled by Advanced in Navigation Sensors, pages KN2–1–KN2–20, Neuilly-sur-Seine, France, 2007.

[86] Ascension Technology Corporation. *3D Guidance TrackSTAR Installation and Operation Guide*. P.O. Box 527, Burlington, VT, USA, 2009. www.ascension-tech.com.

[87] Dany Dubé and P. Cardou. The calibration of an array of accelerometers. Internal Report, Laboratoire de robotique, Département de génie mécanique, Université Laval, Québec, Canada, 2010.

[88] J. Angeles. Fundamentals of Robotic Mechanical Systems. Springer, New York, third edition, 2007.

[89] P. He, P. Cardou, and A. Desbiens. Estimating the orientation of a game controller moving in the vertical plane using inertial sensors. In ASME International Design Engineering Technical Conferences, Chicago, Illinois, USA, 2012.

[90] P. Cardou and J. Angeles. Linear estimation of the rigid-body acceleration field from point-acceleration measurements. ASME Journal of Dynamic Systems, Measurement and Control, 131(4):041013, 2009.

[91] P. Cardou and J. Angeles. Estimating the angular velocity of a rigid body moving in the plane from tangential and centripetal acceleration measurements. Multibody System Dynamics, 19(4):383–406, 2008.

[92] G. Welch and G. Bishop. An introduction to the Kalman filter. Internal Report, Department of Computer Science, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599-3175, USA, 2006.

[93] A. Almagbile, J. Wang, and W. Ding. Evaluating the performances of adaptive Kalman filter methods in GPS/INS integration. Journal of Global Positioning Systems, 9(1):33–40, 2010.

[94] A. H. Mohamed and K. P. Schwarz. Adaptive Kalman filtering for INS/GPS. Journal of Geodesy, 73:193–203, 1998.

[95] J. Wang, H. K. Lee, Y. J. Lee, T. Musa, and C. Rizos. Online stochastic modelling for network-based GPS real-time kinematic positioning. Journal of Global Positioning Systems, 4(1-2):113–119, 2005.

[96] E. Kraft. A quaternion-based unscented Kalman filter for orientation tracking. In Proceedings of the Sixth International Conference of Information Fusion, 2003, pages 47–54, Cairns, Queensland, Australia, 2003.

[97] Y. Beaudoin, E. Gagnon, A. Desbiens, and R. Landry. Introduction au filtre de Kalman non parfumé. Internal Report, R & D pour la défense Canada-Valcartier, Note technique, DRDC Valcartier TN 2011-XXX, 2011.

[98] E.A. Wan and R. van der Merwe. The unscented Kalman filter for nonlinear estimation. In Adaptive Systems for Signal Processing, Communications, and Control Symposium 2000. AS-SPCC. The IEEE 2000, Lake Louise, Alta., Canada, 2002.

[99] J. Ko, D. J. Klein, D. Fox, and D. Haehnel. GP-UKF: Unscented Kalman filters with Gaussian process prediction and observation models. In International Conference on Intelligent Robots and Systems (2007), pages 1901–1907, San Diego, CA, USA, 2007.

[100] Analog Devices. ADXL345 Datasheet. One Technology Way, P.O. Box 9106, Norwood, MA, USA, 2009. `www.analog.com`.

[101] STMicorelectronic Inc. *MEMS motion sensor: ultra-stable three-axis digital output gyroscope*, 2010. `www.st.com`.

[102] Roving Networks. *RN-42/RN-42-N Data Sheet*. 809 University Avenue, Los Gatos, CA 95032, 2010. `www.rovingnetworks.com`.

[103] Actidyn systemes SA. *RT 1100 series*. 1, rue du groupe Manoukian, 78990 Elancourt, France, 2008. `www.actidyn.com`.

[104] P. He, P. Cardou, A. Desbiens, and E. Gagnon. Estimating the orientation of a rigid body moving in space using inertial sensors. Multibody System Dynamics, 2014. DOI 10.1007/s11044-014-9425-8.

[105] M. H. Ang and V. D. Tourassis. Singularities of Euler and roll-pitch-yaw representations. IEEE Transactions on Aerospace and Electronic Systems, 23(3):317–324, 1987.

[106] F. L. Markley. Attitude determination using two vector measurements. In Flight Mechanics Symposium, pages 89–106, 1999.

[107] F. L. Markley and D. Mortari. How to estimate attitude from vector observations. Astrodynamics, 103:1979–1996, 1999.

[108] Y. Cheng and M. D. Shuster. Robustness and accuracy of the QUEST algorithm. Advances in the Astronautical Sciences, 127:41–61, 2007.

[109] Freescale Semiconductor. MAG3110 Datasheet. Austin, Texas, USA, 2011. www.freescale.com.

[110] A. J. Koivo. Fundamentals for Control of Robotic Manipulators. John Wiley and Sons, Inc, New York, first edition, 1989.

[111] NaturalPoint Inc. *NaturalPoint Tracking Toolkits (Point Cloud and Rigid Body) Users Manual*. P.O. Box 2317, Corvallis, OR, 97339, USA, 2008. www.naturalpoint.com.

[112] R. Phillips. The accuracy of surgical navigation for orthopaedic surgery. Current Orthopaedics, 21(3):180–192, 2007.

[113] Y. J. Lee and S. Sung. Vision based SLAM for mobile robot navigation using distributed filters. Mobile Robots Navigation, Alejandra Barrera (Ed.), ISBN:978-953-307-076-6, 2010. DOI:10.5772/8991.

[114] A. I. Mourikis and S. I. Roumeliotis. Analysis of positioning uncertainty in simultaneous localization and mapping (SLAM). In Proceedings of the IEEE/RSJ international conference on intelligent robots and systems (IROS), pages 13–20 vol.1, 2004.

[115] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba. A solution to the simultaneous localization and map building (SLAM) problem. IEEE Transactions on Robotics and Automation, 17(3):229–241, 2001.

[116] G. Dissanayake, H. Durrant-Whyte, and T. Bailey. Computationally efficient solution to the simultaneous localisation and map building (SLAM) problem. In Proceedings of the IEEE international conference on Robotics Automation (ICRA), pages 1009–1014 vol.2, 2000.

[117] Analog Devices. ADXL320 Datasheet. One Technology Way, P.O. Box 9106, Norwood, MA, USA, 2007. www.analog.com.

Appendix A

The Method of Parsa et al.

For comparison purposes, we reproduce here the method proposed by Parsa et. al. [38]. Similar to the method proposed in this thesis, the one by Parsa et. al. provides an estimate $\widetilde{\boldsymbol{\omega}}$ of the angular velocity from both TI and PR estimates. However, the underlying idea is different, as it consists in using a first-order Taylor series approximation to correct the TI estimate $\overline{\boldsymbol{\omega}}_k$.

From Section 2.1, recall that

$$\begin{aligned}\boldsymbol{\xi}(\boldsymbol{\omega}_k) = \big[\omega_{x,k}^2 \quad \omega_{y,k}^2 \quad \omega_{z,k}^2 \\ \omega_{y,k}\omega_{z,k} \quad \omega_{z,k}\omega_{x,k} \quad \omega_{x,k}\omega_{y,k}\big]^T .\end{aligned} \tag{A.1}$$

We also have

$$\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\omega}_k}\delta\boldsymbol{\omega}_k = \delta\boldsymbol{\xi}_k, \tag{A.2}$$

where $\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\omega}} = [2\text{diag}(\boldsymbol{\omega}) \quad \text{sm}(\boldsymbol{\omega})]^T \in \mathbb{R}^{6\times 3}$. According to the first-order Taylor series approximation of $\boldsymbol{\xi}(\widetilde{\boldsymbol{\omega}}_k)$, we have

$$\boldsymbol{\xi}(\widetilde{\boldsymbol{\omega}}_k) = \boldsymbol{\xi}(\overline{\boldsymbol{\omega}}_k) + \frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\omega}_k}|_{\boldsymbol{\omega}_k = \overline{\boldsymbol{\omega}}_k}(\widetilde{\boldsymbol{\omega}}_k - \overline{\boldsymbol{\omega}}_k) = \widehat{\boldsymbol{\xi}}_k. \tag{A.3}$$

Thus, the final estimate is obtained by solving eq. (A.3) for $\widetilde{\boldsymbol{\omega}}_k$, which yields

$$\widetilde{\boldsymbol{\omega}} = \overline{\boldsymbol{\omega}} + \left(\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\omega}}|_{\boldsymbol{\omega}=\overline{\boldsymbol{\omega}}}\right)^{\dagger}\left(\widehat{\boldsymbol{\xi}} - \boldsymbol{\xi}(\overline{\boldsymbol{\omega}})\right). \tag{A.4}$$

Here, $\overline{\boldsymbol{\omega}}$ and $\frac{\partial \boldsymbol{\xi}}{\partial \boldsymbol{\omega}}|_{\boldsymbol{\omega}=\overline{\boldsymbol{\omega}}}$ are given by eq. (2.15), while $\widehat{\boldsymbol{\xi}}_k$ is given by eq. (2.13).

Appendix B

The Rotation Matrix $\mathbf{R}_{\mathcal{BO}}$

OCTA was calibrated [87], which allowed to identify the actual sensitive directions and positions of its accelerometers in frame $\mathcal{B}$, as well as their biases and scale factors. Using the same static calibration method, but with the Optitrack as a reference [87], the sensitive directions $\mathbf{e}_{i,\mathcal{O}}, i = 1, ..., 12$, in the Optitrack frame are found to be

$$\begin{bmatrix} \mathbf{e}_{1,O}^T \\ \mathbf{e}_{2,O}^T \\ \cdot \\ \cdot \\ \mathbf{e}_{12,O}^T \end{bmatrix} = \mathbf{E}_{\mathcal{O}}^T = \begin{bmatrix} 0.6678 & 0.0224 & -0.7440 \\ 0.0119 & -0.9998 & -0.0159 \\ -0.6772 & -0.0745 & 0.7320 \\ 0.7276 & 0.0034 & 0.6860 \\ 0.0007 & -0.9989 & -0.0473 \\ 0.7605 & -0.0114 & 0.6493 \\ -0.0047 & 0.9999 & -0.0127 \\ 0.6713 & -0.0009 & -0.7412 \\ 0.7349 & -0.0189 & 0.6779 \\ 0.6746 & 0.0475 & -0.7366 \\ -0.7731 & 0.0076 & -0.6342 \\ -0.0138 & -0.9997 & -0.0226 \end{bmatrix}.$$

From the definition of $\mathbf{R}_{\mathcal{BO}}$, we have,

$$\mathbf{E}_{\mathcal{B}} = \mathbf{R}_{\mathcal{BO}} \mathbf{E}_{\mathcal{O}}, \tag{B.1}$$

and,

$$\widehat{\mathbf{R}}_{\mathcal{BO}}^T = \left(\mathbf{E}_{\mathcal{O}}^T\right)^\dagger \mathbf{E}_{\mathcal{B}}^T, \tag{B.2}$$

where, $\left(\mathbf{E}_{\mathcal{O}}^T\right)^\dagger = \left(\mathbf{E}_{\mathcal{O}} \mathbf{E}_{\mathcal{O}}^T\right)^{-1} \mathbf{E}_{\mathcal{O}}$ is the left Moore-Penrose pseudoinverse of $\mathbf{E}_{\mathcal{O}}^T$ and $\widehat{\mathbf{R}}_{\mathcal{BO}}$ is the estimate of $\mathbf{R}_{\mathcal{BO}}$. Notice that this estimate does not represent, in general, a rotation matrix. In order to "normalize" this matrix into a rotation matrix, we perform its singular value decomposition:,

$$\widehat{\mathbf{R}}_{\mathcal{BO}}^T = \widehat{\mathbf{U}} \Sigma \widehat{\mathbf{V}}^T, \tag{B.3}$$

and thus we have

$$\mathbf{R}_{\mathcal{BO}}^T = \widehat{\mathbf{U}}\widehat{\mathbf{V}}^T = \begin{bmatrix} 0.6629 & 0.7487 & 0.0071 \\ -0.0044 & -0.0056 & 1.0000 \\ -0.7487 & 0.6629 & 0.0004 \end{bmatrix}, \tag{B.4}$$

where, $\mathbf{R}_{\mathcal{BO}}$ is the true one.

As in the simulations, the initial conditions in the experiments are assumed to be known. The standard deviation of each accelerometer error is $\sigma_a = 0.1g$ m/s^2 [117] and the σ_m of each magnetometer error is $\sigma_m = 0.01$ μT, which gives better estimates after tuning, although the Data Sheet [109] tells us that $\sigma_m = 0.25$ μT. The gravitational acceleration vector is $\mathbf{g} = [0\ g\ 0]^T$, and the sampling period is $\Delta t = 0.01$ s.