UNIVERSITY COLLEGE LONDON
Faculty of Engineering
Department of Civil Environmental and Geomatic Engineering
Space Geodesy and Navigation Laboratory

PhD Thesis

Henry F. S. Martin

Overcoming the challenges of low-cost inertial navigation

Primary Supervisor
Dr P. Groves

Secondary Supervisor
Prof. M. Ziebart

Industrial Supervisor
Mr M. Newman

30th October, 2016

I, Henry F. S. Martin, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, this has been indicated in the thesis.

Abstract

Inertial navigation is always available as a base for multisensor navigation systems on, because it requires no external signals. However, measurement errors persist and grow with time so accurate calibration is crucial.

Large systematic errors are present in the micro-electro-mechanical sensors (MEMS) whose low cost brings inertial navigation to many new applications. Using factory-calibrated MEMS another navigation technology can calibrate these errors with in-run estimation using a Kalman filter (KF). However, the raw systematic errors of low-cost MEMS are often too large for stable performance.

This thesis contributes to knowledge in three areas.

First, it takes a simple GNSS-inertial KF and examines the levels of the various systematic errors which cause the integration to fail. This allows the user to know how well calibrated the sensors need to be to use in-run calibration.

Second, the thesis examines how the end-user could conduct a calibration: it analyses one method in detail showing how imperfections in the procedure affect the results and comparing calculation methods. This is important as frequently calibration methods are only validated by demonstrating consistent results for one particular sensor. These two are primarily accomplished using statistical Monte Carlo simulations.

Third, techniques are examined by which an array of inertial sensors could be used to produce an output which is better than the simple array average. This includes methods that reduce the array's sensitivity to environmental conditions, this is important because the sensors' calibration typically depends strongly on temperature.

Also included in the thesis are descriptions of experimental hardware and experiments which have been carried to support and unify the other parts of the thesis.

Overall, this thesis' contributions will help make low-cost inertial navigation systems more accurate and will allow system designers to concentrate effort where it will make the most difference.

Acknowledgements

I would first like to thank my supervisor, Dr Paul Groves, for all his guidance, expertise and humour. Without Paul I would not have enjoyed this PhD nearly as much; he has been an excellent supervisor and I thoroughly recommend him to future students.

My project is funded by the Engineering and Physical Sciences Research Council and BAE Systems Applied Intelligence Laboratories (Chelmsford), award number 10001940, for which I am very grateful.

My industrial supervisor, Mark Newman, has been a great help particularly with my numerous hardware problems and questions about electronic engineering. My previous industrial supervisor, Dr Ramsey Faragher, is a great inspiration, and very generous with his time. I would also like to thanks my secondary supervisor, Prof. Marek Ziebart, for all his help with my project.

Dr Santosh Bhattarai has been a great help to me during this project with both motivation and software. Thanks also to Dr Stuart Grey and Dr Mounir Adjrad for their help and advice on technical issues.

I would also like to thank Ian Seaton, Warren Gaynor, Les Irwin, Gerald McBrearty, Eduard Kriegler, Alan Rock and Dr Sean Allgeier for their invaluable help with hardware development. Thanks to Dr James Haworth, Mike Dunderdale and Dr John Davies for their help with other technical problems I came across.

My mates have kept me sane during this whole process, so thanks to Dr Seb Pancratz, Hira Virdee and Rajiv Joshi.

Thanks to all my colleagues past and present at SGNL, for being friends, sounding boards and their advice and help. Dave, Chris, Lei, Naomi, Kimon, Vicki, Debbie, Han, Sam, Haraldur, Toby, Xiyi, John, Alex, Mari and Zhen thanks to you all!

I am grateful to all the people who helped me to get to the position of being able to start my PhD, particularly Dr Chris Prior and Dr Maurizio Collu.

Finally, thanks to my wonderful girlfriend, Tuula Eriksson, for being amazing, and to my parents, Richard and Nell, and sister Felicity for their love and support over the years.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the last two decades, the development of cheap user-equipment for global navigation satellite systems (GNSS) has led to many new uses for positioning technology. These include: route-finding and other location-based services on mobile devices; low-cost unmanned aerial vehicles (UAVs); asset tracking; performance monitoring for sports; automatic control systems; and enhanced situational awareness for military or emergency response team commanders. Consumers now expect to be able to know their position at any time and to within a few metres. However, GNSS is not reliable everywhere, particularly in and around tall buildings. An inexpensive solution is needed which can extend the good navigation performance of GNSS systems in open areas to more challenging reception environments, without being vulnerable to jamming, interference and spoofing like GNSS.

This demand for universally available positioning cannot be met by a single technology, at least not without great cost. However, an integrated system of several different navigation technologies can provide a cost-effective navigation system with good performance in a variety of environments [1].

Inertial navigation systems (INS) provide a navigation solution at all times based on determining changes of velocity (acceleration) and changes of angular position (angular rate) using an inertial measurement unit (IMU). This navigation technique relies on tracking the previous position, velocity and attitude and using the measured changes to update their values. Thus at any moment in time the positioning solution depends on all the previous measurements as well as the most recent one, so errors increase with time. However, as it does not rely on external signals it is always available and it can provide a basis for use when other technologies are not available. INS errors can be estimated when more accurate information is available and integration of an INS can even improve the performance of those other techniques [2, 3].

IMUs consisting of micro-electro-mechanical-systems (MEMS) can be inexpensive, and are also small and lightweight, making them suitable for many applications. However, these inexpensive sensors are not supplied calibrated, which means that they have errors so large that they are unsuitable for inertial navigation for more than a few of seconds [1,4]. For optimal performance calibration is required for fixed, time-varying and temperature-dependent systematic errors as well as characterisation of stochastic (random) errors. The latter is typically achieved by examining samples of static data using tools such as Allan variance analysis [5], although the former represents the low-hanging fruit for performance improvement of low-cost IMUs.

There are four main methods which can improve the navigation performance of low-cost MEMS IMUs. These are: improving MEMS sensor technology; IMU calibration prior to use; integration with another navigation technology to calibrate the sensors during use; and using arrays of IMUs to measure the inertial forces more accurately by averaging. Improving MEMS sensor technology requires different expertise to the other three, and so is not considered within the scope of this thesis.

1.1 Research objectives

The aim of this thesis is to improve the navigation performance of low-cost MEMS inertial sensors, particularly in the context of an integrated navigation system. The Kalman filter (KF) is the standard technique for integrating an INS with other navigation sensors. However, it can become unstable when used with low-cost sensors, due to the magnitude of their errors. There is a need to establish where the limits of KF stability are.

The first objective of this thesis is to determine how well calibrated an IMU must be to allow a stable multi-sensor integration, using only a basic Kalman filter.

Pre-calibrating the IMU can significantly improve navigation performance. However, laboratory calibration techniques vastly increase the cost (from <£5 to >£1,000). To reduce the unit cost, there is a need to investigate whether the end-user could calibrate the sensors themselves, without any specialist equipment. Thus, the second objective of this thesis is to assess what level of calibration accuracy can be achieved with various user-conducted calibration manoeuvres.

The third objective of this thesis is to investigate ways of getting better performance from MEMS using arrays of sensors. Specifically arrays that outperform a simple average of the sensors' output, by exploiting characteristics of the sensors' design, such as error correlation between sensors of the same model.

1.2 The main contributions of this thesis

The main contributions of this thesis are as follows:

- Development of an integration filter stability testing method, where a convenient consistency test has been implemented for Kalman filter INS/GNSS integration. This could easily be extended to other types of integration filter. This testing method has been used to determine the limits, in terms of IMU errors, within which a basic KF INS/GNSS integration remains stable.
- Determination of the estimation accuracy for existing and improved manoeuvres and algorithms for user-conducted calibration. The effect of factors such as the system casing orthogonality, table levelling and the accuracy with which the human conducts the manoeuvres have been investigated.

- Proposing new manoeuvres and algorithms that have been shown to improve the accuracy of the calibration. The user-calibration techniques calibration estimates are shown to be repeatable with real IMUs and to improve their navigation performance.
- Three methods for improving the calibration performance by using sensor arrays have been examined. These are anti-parallel arrays, mixed measurement range arrays, and asymmetric arrays. The first of these has been shown to improve the performance of real MEMS IMUs.

1.3 Publications arising from this research project

The research that is presented in this thesis has resulted in several publications.

The research on Kalman filter stability limits (Chapter 3) has been published in:
Martin, H. F. S., Groves, P. D., and Newman, M. *The limits of in-run calibration of MEMS inertial sensors and sensor arrays.* NAVIGATION, Journal of the ION, 2016, 63(2), p.127–139.

A preliminary version of which was presented at:
Martin, H. F. S., Groves, P. D., and Newman, M. *The limits of in-run calibration of MEMS and the effect of new techniques.* In Proc. ION GNSS (September 2014).
The presentation of this paper at ION GNSS+ 2014 was awarded with a "Best presentation in session" prize.

A preliminary version of the examination into array techniques (Chapter 6) was published in:
Martin, H. F. S., Groves, P. D., Newman, M., and Faragher, R. *A new approach to better low-cost MEMS IMU performance using sensor arrays.* In Proc. ION GNSS (September 2013).

A journal submission will also be prepared on the user-conducted calibration research presented in Chapter 4.

The author has also been a co-author on work not presented in this thesis but in related areas:

Groves, P. D., Wang, L., Walter, D., Martin, H. F. S., Voutsis, K., and Jiang, Z. *The four key challenges of advanced multisensor navigation and positioning.* In IEEE/ION PLANS (Monterey, CA, May 2014).

Groves, P. D., Martin, H., Voutsis, K., Walter, D., and Wang, L. *Context detection, categorization and connectivity for advanced adaptive integrated navigation.* In Proc. ION GNSS+ (Nashville, Tennessee, 2013).

Groves, P. D., Wang, L., Walter, D., Martin, H., and Voutsis, K. *Toward a unified PNT, part 1: Complexity and context: Key challenges of multisensor positioning.* GPS World 25, 10 (2014).

1.4 Thesis structure

Chapter 2 examines existing approaches to improving the performance of low-cost inertial navigation. This begins with background on inertial navigation, GNSS, navigation sensor integration and other possible sensors that could be included in an integrated navigation system. Then different algorithms that could be used to integrate the data from multiple navigation systems are introduced. The existing work on user-calibration is examined, revealing a wide range of different definitions and approaches to the problem. Then, the state-of-the-art in relation to using arrays of low-cost sensors is presented. Finally, the topics that require further investigation are discussed.

Chapter 3 presents research on the limits of stability of Kalman filter integration of INS with another positioning technology. First the question of, at what point a filter has become unstable, is examined and criteria for filter failure are presented. Next, an approach is presented to simulate INS/GNSS integration and thus determine for what IMU specifications the KF remains stable. The results of these simulations are then presented both for motion representing a road vehicle and for a quadcopter.

Chapter 4 presents user-conducted calibration procedures and simulations to assess their effectiveness. It begins by outlining the principle of the calibration procedure, followed by defining the reference frames needed for the simulations. Then, the IMU error model used is described in detail, followed by the way that these error parameters are estimated from the sensor outputs. The simulation method is presented next, followed by the assessment criteria. Then the simulation results are presented and discussed.

Chapter 5 presents experiments carried out with custom-developed hardware. These experiments are used to characterise the sensor's errors, and demonstrate that the calibration procedure introduced in Chapter 4 provides repeatable results. These results are used to validate the simulation-predicted accuracy and repeatability.

Chapter 6 presents and examines three possible techniques for optimising the performance of arrays of low-cost IMUs, compared to simply averaging the output of an array of parallel IMUs. These are: orienting similar IMUs so that their sensitive axes are parallel but in opposite directions to remove the effect of common-mode errors; combining sensors with different measurement ranges to improve low-dynamics performance while maintaining the ability to measure high-dynamics; and optimising the asymmetric performance of the IMUs in the array to the asymmetry of the application. The performance of these techniques are compared to that of simply averaging an array of parallel IMUs

Chapter 7 presents the conclusions of the thesis, and discusses their implications on low-cost inertial navigation. It also proposes possible future extensions to this work and follow-on research opportunities.

Appendix A describes the bespoke hardware that was designed for this project's experiments. Details are provided of both a less sophisticated version used at the start of the project and in [8], and a fully custom design used for the final experiments.

Chapter 2

Background and Literature Review

This thesis has the title Overcoming the challenges of low-cost inertial navigation. This chapter begins by explaining what is meant by *navigation* and some related terminology in Section 2.1. In Section 2.2, the basics of inertial navigation are explained. This includes the sensors needed, their errors and what is meant in this project by *low-cost inertial navigation*.

Then more detail on the challenges presented by inertial navigation with low cost sensors are discussed. Inertial navigation, particularly low-cost inertial navigation, works best as part of an integrated navigation system. Many different sensors could be included in an integrated navigation system, see Section 2.3. An integration algorithm is used to combine the outputs of multiple sensors. However, the simplest and most commonly used algorithms become unstable when used with low-cost inertial sensors, requiring more complex and computationally expensive algorithms, see Section 2.4. Section 2.5 reviews other methods of obtaining better navigation performance from inertial sensors, including pre-calibration and using arrays of IMUs.

The key challenges that must be overcome are summarised in Section 2.6.

2.1 Navigation and navigation systems

The term *navigation* is commonly used in two different ways [1, 12]. In this project navigation refers to finding one's current location relative to some local or global frame of reference. This is sometimes also called localisation or positioning, although the latter can include determining the position of others. The other commonly used meaning of the term navigation is the method or skill of finding one's way from A to B. This is sometimes called route-finding, and is not the concern of the research in this thesis, hereafter only the first definition is used.

The other important distinction for this project is that it concerns navigation in, or very close to, real time. Positioning a user or device where a delay of tens of seconds or minutes is not important is a different problem, which is sometimes referred to as tracking [1], c.f. parcel tracking or asset tracking. Aside from the storage of data, tracking is at least no harder than real-time navigation. Tracking is also sometimes used to mean estimating the position of an object based on remote measurements, rather from sensors attached to the object being positioned [13], c.f. target tracking.

Also note that the navigation systems considered here are automatic electronic ones, rather than anything manually operated.

When in this thesis a navigation solution is referred to, this means a navigation system's estimate of position, velocity and attitude, relative to a particular reference frame [1]. The attitude of a body or navigation system is the rotation between its axes and those of the reference frame. It can be calculated in terms of a co-ordinate transformation matrix, and is primarily done so in this project. It can also be expressed in quaternions or Euler angles of roll, pitch and yaw [14].

Navigation performance can include several different metrics. Accuracy is the amount by which, on average, the estimated position velocity and attitude (PVA) solution deviates from the true PVA. Availability is the proportion of the time that a position (or PVA) solution is available. Continuity measures the probability that a position (or PVA) solution will be available without interruption during a specified period. Integrity performance criteria measure the ability to maintain a PVA solution within certain bounds during operation and correctly report when the error exceeds these bounds. When this thesis refers to navigation performance it refers to the accuracy of the position, velocity and attitude solution, unless otherwise specified. Note that for many users the position is the most important, although, for example attitude may the most important for a UAV autopilot.

Navigation systems have a wide range of applications with varying performance requirements and correspondingly different price tags [1]. On the high performance end, these can be bespoke, highly accurate systems with high integrity requirements for submarines, surface ships, aircraft and missiles [15, 16]. When aimed for the mass market, a navigation system will not have such high requirements in order to reduce its cost. Bridging these, are devices with high accuracy requirements but not such high integrity requirements, for applications such as surveying [1].

Some applications will require good navigation performance on a global scale, whereas others may only require good accuracy relative to a nearby base-station. An example of the latter is precision farming, where agricultural machinery can be positioned relative to static beacons placed in or near the same field [17]. An accurate position relative to a local base-station can be easier to achieve, particularly for radio positioning systems [1].

A navigation technology can be used to provide or improve upon a position, velocity and attitude (PVA) navigation solution, or a subset of it. Both the navigation solution and the navigation system's measurements are valid at a specific instant in time, which is known as an epoch. Those navigation technologies that can provide a full navigation solution are either position fixing or dead reckoning.

Position fixing technologies use range, proximity or bearing (e.g. angle-of-arrival) measurements to position the user relative to a network or beacons or landmarks [1]. A commonly used example of this is GPS, where the GPS user equipment calculates ranges to several members of a network of satellites whose orbits are known (see Section 2.3.1). Typically, position fixing technologies will need to calculate measurements from several beacons/landmarks to calculate a full navigation solution, e.g. 4 satellites in the case of GPS.

However, often a reduced number of measurements can still aid a navigation solution calculated by other means.

In the case of dead reckoning[1] technologies, the measurements provided are changes in position, velocity and/or attitude, over the period of time since the last measurement [1, 4, 12]. This is sometimes known as delta-position, delta-velocity and/or delta-attitude. These are also sometimes called increments, e.g. an attitude increment. The measured change in position, velocity and attitude is added to the position velocity and attitude at the previous epoch. If you know where you are at the start of the period of navigation, you can calculate where you have been and thus where you are now. The crucial point is that the navigation solution at a particular epoch always depends on that of the previous epoch. For this reason, errors in the navigation solution will always grow with time, as an error introduced at a particular epoch persists in future epochs [1]. Historically, the most well-known example of this type of navigation was nautical navigation by means of a compass and the ship's log, providing a measurement of speed and heading. This method is still used in nautical navigation, and its accuracy has been enhanced by more accurate gyrocompasses and logs based on different technologies. Other examples include foot travellers navigating in poor visibility, a featureless environment or unmapped area by counting paces and using a compass, the modern electronic version of which is called pedestrian dead reckoning (see Section 2.3.2.3). Both of the preceding two examples are essentially two-dimensional dead reckoning. However, the focus of this thesis is on inertial navigation which is a three-dimensional dead reckoning technology, described in Section 2.2.

Navigation technologies can produce many different types of measurements. For position-fixing technologies these are primarily either ranges and/or range-rates (rate of change of range), bearings to beacons/landmarks whose position is known (giving lines-of-position) or proximity measurements [1]. Other navigation technologies produce some subset of measures of position, velocity and/or attitude, or their rate of change (e.g. acceleration or angular-rate). For example, a compass measurement can be used to derive heading, one component of the attitude and a barometer can be used to derive height, one component of position (see Sections 2.3.2.1 and 2.3.2.2).

Navigation solutions and (nearly all) navigation technologies' measurements are vector quantities so must be defined relative to a reference frame. An inertial reference frame is a reference frame that does not accelerate or rotate relative to the rest of the universe [1]. This is particularly important for inertial navigation as it provides measurements relative to an inertial reference frame. The most commonly used inertial reference frame for navigation is Earth centred inertial (ECI). The origin of this frame is at the centre of the Earth, the z-axis towards true north and the y-axis is 90° ahead of the x-axis in the Earth's rotation (so the rotation is positive). While this is not strictly an inertial frame, the rotation of the Earth around the sun ($\approx 1.14 \times 10^{-5}$ °/s) and the sun around the centre of the galaxy are small enough to ignore for navigation applications [1]. Several conventions exist for defining the directions of the x- and y-axes relative to the stars (see [1]). This thesis assumes that it coincides with the Earth centred Earth fixed frame at the start of the period of navigation. The Earth centred Earth fixed (ECEF) frame has its origin at the centre of the Earth, its

[1] A corruption of ded-reckoning for deduced reckoning [12]

x-axis points to the intersection of the Greenwich meridian and the equator and y-axis to 90° E longitude [1]. This frame is frequently used to describe the position of beacons in position fixing technologies.

A local tangent frame is used in this thesis, particularly the simulations in Chapter 4 and the dynamic tests in Chapters 5 and 6. This describes a Cartesian frame, centred at the starting position (relative to the earth) of the navigation system, with the z-axis coincident with the local gravity vector (normal to the geoid) pointing away from the Earth, y-axis the projected direction of the north pole, the x-axis points in towards local East (this convention is known as ENU) and does not rotate relative to ECEF.

The body frame of a vehicle is the frame is centred on a particular point fixed to the vehicle, often the position of its centre-of-mass or navigation system. Its axes are defined relative to the vehicle and rotate with it, this thesis uses forward, left and up for x, y and z, respectively. If a vehicle has several sensors fixed to it at different positions, to differentiate the sensor's body frame from that of the vehicle, sensor frame is used to describe the former. The position and orientation of these sensor frames are usually assumed to be fixed in the body frame. Sensor measurements can be transformed from the sensor frame to the body frame using a lever-arm transformation (p77–78 [1]).

There are several other reference frames used in navigation and alternative conventions, which are not used in this thesis.

More information about all of the subjects covered in this section can be found in books such as [1, 4, 18, 19]. This thesis follows the naming conventions used in [1], except where noted.

2.2 Inertial navigation

Inertial navigation is a dead-reckoning navigation technology that uses measurements from accelerometers and gyroscopes to perform the navigation solution updates [1, 4, 20]. A (strapdown) inertial navigation system (INS) consists of an inertial measurement unit (IMU) and a processing unit. The term inertial sensors is used in this thesis to refer generically to both accelerometers and gyroscopes, usually collectively.

The IMU consists of at least 3 accelerometers and 3 gyroscopes, each of which are sensitive to motion in a particular direction — their sensitive axis[2]. For each type of sensor, its three sensitive axes must be non-coplanar, to allow sensitivity in all directions. Typically these triples are arranged so their sensitive axes are orthogonal, which gives equally good sensitivity in all directions. This project is concerned with microchip-scale micro-electro-mechanical-systems (MEMS) technology which can be used to miniaturise the inertial sensors, also vastly reducing their weight and power consumption [20]. A MEMS IMU can

[2]It is possible, in some circumstances, to track two-dimensional position using a lower number of inertial sensors, a so-called "reduced IMU" [21, 22], typically 2 accelerometers and 1 gyroscope. However 2 dimensional navigation is only relevant to a small number of applications and so shall not be a major concern of this project.

be inside a first level package (FLP), i.e. single electrical component, as small as $3 \times 3 \times 1$ mm [23].

The IMU also powers the sensors, digitises their outputs, and sends this data via a communications protocol to the systems' processor. An IMU may apply corrections from a calibration to the measurements, and many also contain a temperature sensor, e.g. [23, 24]. A MEMS IMU component may also contain other navigation sensors, such as magnetometers and sometimes barometers (see Sections 2.3.2.1 and 2.3.2.2, respectively).

A single component containing three orthogonally sensitive inertial sensors is sometimes referred to as a 3-axis sensor. Similarly if a single component contains both a 3-axis accelerometer and gyroscope, i.e. a complete IMU, it can be referred to as a 6-axis sensor. If a 3-axis magnetometer is also included, it may be referred to as a 9-axis sensor (e.g. [25]), and adding a barometer to that makes a "10-axis MEMS IMU" according to [26], these are marketing terms but widely used in the field.

It is important to note that the manufacturing techniques used for low-cost MEMS sensors, e.g. surface micromachining, remove material from a wafer of silicon [20]. Thus, the structures created in low-cost MEMS are inherently 2D. This is not because 3D construction cannot be done, rather it is because it is far cheaper to build planar structures partly because techniques and machinery from the microprocessor construction industry can be used [20]. This means a 1D accelerometer must have a different design if it is to operate out-of-plane (Z-axis) than if it operates in-plane. In order to create 3-axis sensors which have nearly orthogonal sensitive axes, all 3 sensors can be constructed on the same piece of silicon, or even all three gyroscopes and all three accelerometers. There are two possible ways to accomplish this: a parallel implementation and/or multi-DOF sensing elements. In a parallel implementation the sensors are separate 1D sensors positioned side-by-side, e.g. [27]. In multi-DOF configuration, a single proof mass is allowed to move in several directions and the strain in the springs restraining it is measured. This allows the forces to be inferred from assumptions about the deformation of these under load, e.g. from finite element analysis (FEA). For example, in [28] a single proof mass accelerometer is restrained by 4 leaf springs in a cross formation whose deformation is predicted by FEA. [20] notes that this sensor design is capable of sensing specific force/moments in all 6 DOF, although the authors themselves do not mention this.

2.2.1 Accelerometers

Accelerometers, despite their name, do not actually measure acceleration. Instead, they measure specific force, which is non-gravitational force per unit mass. This is, in simple terms, the reaction force exerted on the accelerometer to counter acceleration due to gravity (for instance, by the ground) in addition to any acceleration [4, 29]. Accelerometers' specific force measurements are of the body frame relative to inertial space [1]. The symbol used in this thesis is $\boldsymbol{f}$. Sensor specifications often describe specific force measurements in terms of g ($\approx 9.8\,\mathrm{ms}^{-2}$), rather than the metric ms^{-2}, e.g. [23], presumably this is to make the specifications equally accessible in the US and the rest of the world. This is because conversions of g-force are common when using either metric or imperial measurement systems.

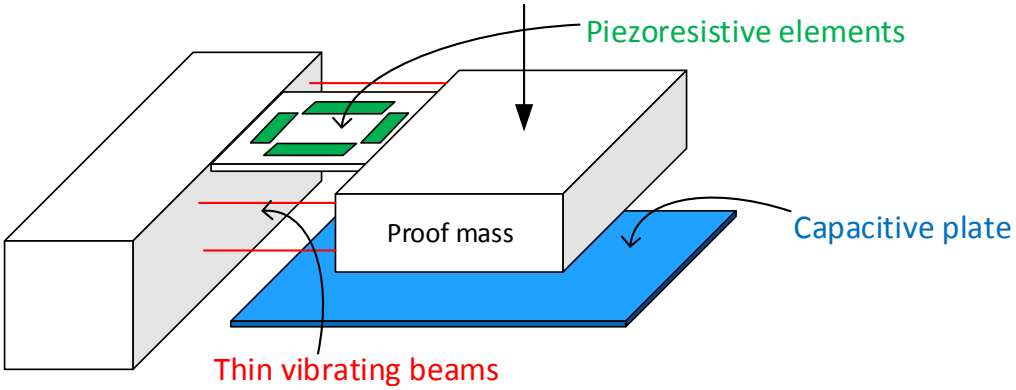


Figure 2.1: Three possible methods for measuring deflection in a (z-axis) pendulous accelerometer. Only one would be used in a real sensor. Adapted from several figures in [20].

There are a number of ways in which an accelerometer can operate but it can be most simply visualised as a mass, known as the proof-mass, inside a box, which is free to move in one particular direction but restrained by a spring. The displacement in MEMS sensors is read by a transducer which can be piezoelectric, piezoresistive or capacitative and may exploit the change in natural frequency of a bar under tension or compression [20]. One possibility is that this mass can be allowed to displace from the central (zero-force) position and this displacement measured, for example by proximity to a capacitative plate; this is called "open loop". The other possibility, known as closed loop, is that a sensitive pickup is placed at the end of the lever arm, where small displacements are read and a control loop electronically exerts a force, e.g. through an electromagnet, to hold the proof mass in the central position. The measurement in this case is proportional to the force required to hold the proof mass in the centre.

The proof-mass must be attached stiffly in five out of the six possible degrees of freedom (DOF) (3 linear and 3 angular), and compliantly in its sensitive direction [20]. In practice this cannot be done perfectly and can be a source of cross-coupling errors (see Section 2.2.5). For example, a simple cantilevered beam will be sensitive to moments about its hinge. However, more complex spring-mass system designs can isolate the proof mass better.

In general closed-loop accelerometers are superior to open loop ones, this is because the proof-mass displacement is kept small, which reduces stresses and fatigue on the springs also they can be more compliant reducing hysteresis losses, also the reduced displacement can reduce sensitivity to cross-coupling errors [20].

Closed-loop MEMS accelerometers are available [30]. Although they are only used (p.149 [31]) in high precision applications [32] almost all automotive-grade sensors are open-loop [1, 20].

It is possible to have angular accelerometers which are sensitive to angular accelerations, rather than linear ones [33]. These exist commercially [34] and are used for applications such as detecting angular shocks in hard discs [35]. It would be possible to conduct 3D dead reckoning with 3 linear and 3 angular accelerometers. However, in practice this is not done as MEMS gyroscopes give a superior attitude solution [20].

2.2.2 Gyroscopes

The gyroscopes that this project is concerned with measure angular rate, sometimes they are called rate-gyroscopes for this reason. This distinction is mostly historical as many 20^{th}-century gyroscope designs used gyroscopes that had at their heart a spinning mass suspended within (at least) three rings (gimbals) linked by bearings which start 90° apart. This allowed direct reading of the craft's attitude at the gimbals' bearings, as the spinning mass' attitude remained fixed and the craft moved around it. However, these designs had a number of problems including many expensive high precision moving parts and gimbal lock could occur where one degree of freedom of the gyroscope could be lost by two of the rings lining up. This kind of gimballed gyroscope is now only used in high-cost high-performance applications, which are outside the scope of this project. All further references to gyroscopes refer to rate-gyroscopes.

Some authors use the term strap-down inertial navigation system (SDINS or SINS) to refer specifically to those INSs with rate-gyroscopes, e.g. [4]. This report will use INS on the assumption that the reader understands that a SDINS is being referred to.

There are many different types of gyroscope design in current use, including ring-laser gyroscopes (RLG) [36], fibre-optic gyroscopes (FOG) [37] and hemispherical resonator gyroscopes (HRG) [38]. However, these designs are not used for current low-cost MEMS gyroscopes [1]. These gyroscopes are usually vibratory gyroscopes, based on either a tuning fork principle [39] or by vibrating a single proof mass, with either a translational (e.g. [40]) or rotational oscillation [20].

Vibratory gyroscopes measure Coriolis forces [1, 4, 20]. A Coriolis force is not an actual force, rather it is a consequence of a rotating reference frame. A centrifugal force acts on all objects in a rotating reference frame. In addition to this, when an object is moving with constant velocity with respect to a rotating reference frame (and not parallel to the rotation axis), it is moving in a curve with respect to inertial space. Thus to maintain this apparently constant velocity, a force orthogonal to the velocity and the rotation axis is required to create the curve [1].

In order for a vibratory gyroscope to work, it must have both a drive axis and a sensing axis. It will be sensitive to rotations about the axis orthogonal to these (input- or sensitive-axis). The proof mass will be vibrated along the drive-axis and, if the gyroscope is rotating about the input-axis, this motion will create a Coriolis force in the sensing-axis (also vibratory), where it will be picked up by what is effectively an accelerometer (measuring specific Coriolis force) [20]. One possible design is shown in Figure 2.2. This means that a gyroscope is fundamentally sensitive to accelerations in its sensing direction, and its design must compensate for this. One option is to have a pair of masses which vibrate out-of-phase, e.g. a tuning fork [39], then the phase difference on the two mass' sensing axes can be used to infer angular rate. Another option is to have both driving and sensing axes be rotational, which isolates the sensor from linear accelerations. Note that, the Coriolis vibrations that need to be sensed are orders of magnitude smaller than the driving vibrations [20], and also that the need to drive the gyroscope means that vibratory gyroscopes are inherently higher power consumption than accelerometers.

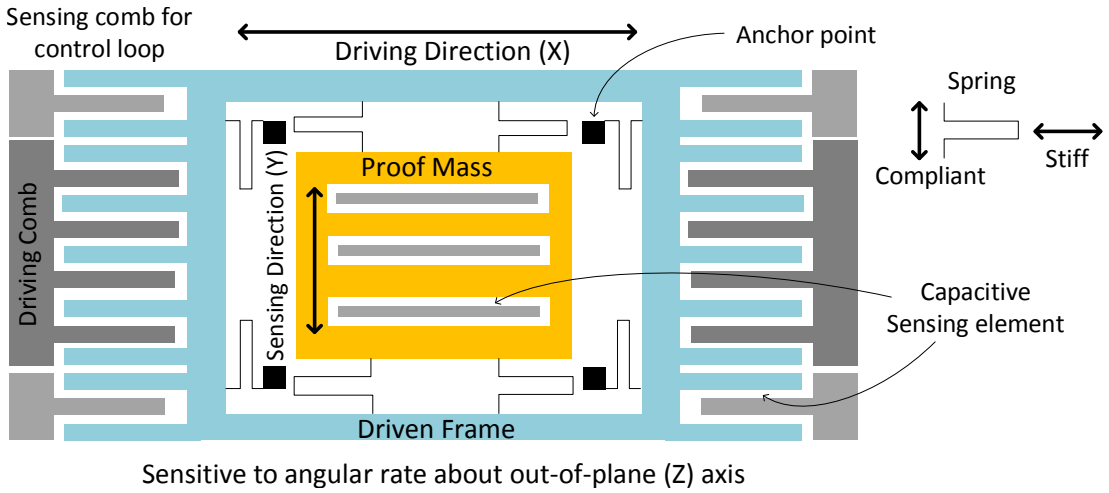


Figure 2.2: One possible operating principle for a z-axis vibratory gyroscope. Grey and black parts are fixed to the body of the gyroscope. Adapted from [20].

This also means that an open-loop gyroscope under a high angular rate will have its vibratory axis shift, due to the deformation of the springs in the sensing axis, potentially changing the sensitive axis (depending on the design). It is also worth noting that a 3-axis gyroscope with a parallel implementation (3 proof masses) must have sufficiently different driving frequencies for each mass to avoid cross-talk, which is one source of cross-coupling error.

Due to the vibratory nature of the motion, most of the effect of the accelerations on a gyroscope can be removed by a low-pass filter, if the filter's characteristic frequency is considerably lower than the driving frequency. However, in the presence of external vibration (e.g. from a vehicle's engine) similar to the driving frequency of the gyroscope, sensor performance can become extremely poor. Sensors intended for the automotive industry are frequently designed with this in mind [20]. Low-cost UAVs (e.g. quadcopters), frequently have general-use sensors not designed for high-vibration environments, so engine vibration interfering with the gyroscopes can be a significant problem, leading researchers to test alternatives techniques to aid the attitude solution [41].

2.2.3 Inertial navigation equations

At the start of its use the INS is first initialised with a starting position, velocity and attitude. Then the inertial navigation equations are run at every epoch to update the navigation solution with specific force and angular rate measurements from the IMU. The usual assumption is that these measurements represent the average of the quantity in question over the period since the last measurement [1].

First, the old attitude solution must be updated with the angular rate measurements. The angular rates can be used to calculate the angular increment (change in angle) over the time period since the last measurement. If these angular increments are not very small, problems can arise related to the non-commutativity of rotations. This is one reason why the update rate should be kept high, to keep angular increments small [1].

In order to derive acceleration from the specific force measurement, the component due to the gravitational attraction of the Earth must be added to remove its effect. This relies on the attitude estimate. Thus if there has been significant attitude drift, then the reaction to gravity correction will be made in the wrong direction and so spurious accelerations will be derived. Simple trigonometry shows that an attitude error in pitch or roll of 5.9° gives an incorrect acceleration of 1 ms^{-2} horizontally and 0.05 ms^{-2} vertically, and 0.59° gives 0.1 ms^{-2} horizontally and 0.0005 ms^{-2} vertically. The acceleration is used to update the estimate of velocity and then this estimate of velocity must be used to update the estimate of position. As such an instantaneous error in the acceleration measurement causes a constant error in the velocity estimate and thus an increasing error in the position estimate. This double integration makes the position deteriorate quickly when there are errors in the specific force measurement or attitude solution.

There are several variants of the inertial navigation equations, these depend on, for example which frame the results are relative to (e.g. Earth centred inertial (ECI), Earth centred Earth fixed (ECEF) or local navigation frame) [1, 4, 19].

2.2.4 Grades of INS performance

As any inaccuracies in specific force or angular rate persist in the INS's estimates of position, velocity and attitude accurate measurements of angular rate and acceleration are essential. If very high-performance (so called 'marine grade') sensors are used, then the accuracy can be good enough to navigate with these sensors alone for hours (1.8km navigation solution drift per day according to [1]). However, these high performance sensors cost many hundreds of thousands of pounds, and are typically used in military ships and submarines.

Lower performance and (therefore) lower cost sensors are available at several different grades [1]. Aeronautical grade sensors have around 1.5km drift in the first hour, and are typically used in military aircraft and commercial airliners. Intermediate grade IMUs are used for smaller aircraft, and still cost tens of thousands of pounds and have around 10× worse performance than Aeronautical grade sensors. Tactical grade sensors are can refer to a wide range of performance with prices ranging from £1,500 to around £10,000. Typical applications of tactical grade sensors are guided weapons and unmanned aerial vehicles (UAVs). The lowest grade of IMUs are consumer or automotive-grade. These are typically supplied without individual calibration. These were originally developed for applications other than navigation, such as airbags and ABS systems. Consumer grade inertial sensors, or at least the accelerometers, are included in every smartphone, originally to determine the direction of gravity for automatic screen rotation, but many other uses, such as gesture detection, activity monitoring and gaming, have been added since.

It should be noted that effective use a higher-grade IMU requires a more complex error model than that of a lower-grade IMU. A marine-grade IMU would benefit from, for example, a very high quality model of how the value of g varies over the surface of the Earth and with height [42]. When using a tactical-grade IMU this would be of little benefit relative to a simple one-parameter (latitude) model.

It is generally accepted that tactical is the lowest grade that can realistically be used for standard inertial navigation [1,4]. This is not to say that automotive-grade sensors cannot be used for navigation, rather that they cannot be used unconstrained without calibration. For examples of how quickly a consumer-grade INS drifts from the truth, the reader is referred to Chapter 5, particularly Figures 5.21 and 5.25. Common applications of consumer-grade IMUs in navigation include motion classification, pedestrian dead-reckoning and foot-pod mounted IMUs (see Section 2.3.2.3 and 2.3.3)

The precise point between tactical and consumer-grade IMUs where inertial navigation becomes a reasonable possibility is an open research question which shall be addressed in Chapter 3.

2.2.5 IMU Errors

Real accelerometers and gyroscopes do not measure specific force and angular rate perfectly. The difference between the output that a sensor produces and the true input is its error. These can be split into stochastic and systematic errors. This section focusses on the errors relevant to low-cost MEMS IMUs.

The stochastic errors, also called random errors, are noise-like errors and are not predictable and cannot be calibrated away. The noise on an inertial sensor is typically composed mainly of white noise, plus other sources such as Markov processes or flicker noise (see e.g. [43, 44]). Accurate measurement of their stochastic properties (see Section 2.5.1.1) can help sensor integration (see Section 2.4). The total output noise is caused by a combination of electronic noise, from the electronic measurement of the sensor deformation and amplification of the signal, and Brownian noise, from the impact of gas molecules on the proof-mass [20]. However, other physical error sources are often included in the stochastic model. These include the quantisation error, which is the error from rounding the output to an integer to output. Quantisation residuals (the truncated part) in high performance sensors are typically added to the next measurement, although in many low-cost sensors they are discarded [1]. The most frequent noise characteristics specified for low-cost IMUs are root mean squared (RMS) total noise and noise root power spectral density (PSD). Typical specified values for root-PSD for current generation low-cost MEMS IMUs are 150 $\mu g/\sqrt{Hz}$ [45] to 300 $\mu g/\sqrt{Hz}$ [23] (accelerometer) and 0.01 $^\circ/s/\sqrt{Hz}$ [23] to 0.014 $^\circ/s/\sqrt{Hz}$ [45] (gyroscope).

For systematic errors, past behaviour can be used to predict future behaviour. As such, their effect on navigation performance can be reduced by *individual* calibration, the error remaining after calibration is termed a residual error. Systematic errors are commonly classified by their relationship with the sensor input, thus an n^{th}-order error depends on the n^{th}-power of the sensor output. A particular sensor's errors can thus be thought of as fitting polynomial coefficients to a curve. However, the calibration estimates (i.e. the coefficients) can be affected by environmental conditions such as temperature. This makes a calibration carried out at one temperature less useful at another. To counter this effect calibration often includes under different temperatures, to allow the effect of temperature on, at least some, of the error parameters to be determined (see Section 2.5.1). The errors can also vary slightly with time, both day-to-day and during use [1].

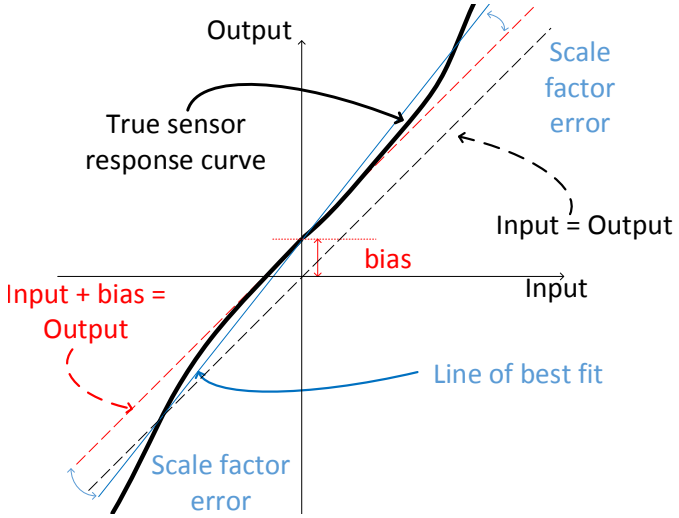


Figure 2.3: Illustration of bias and scale factor errors.

The most simple systematic error is bias, also known as zero-g offset or zero-rate offset for accelerometers and gyroscopes, respectively. The bias is added to the output of the inertial sensor irrespective of the input, this makes it 0^{th} order. It is usually visualised (and measured) as the output when the input is known to be zero, e.g. when a gyroscope is static. This thesis uses $\boldsymbol{b_a}$ and $\boldsymbol{b_g}$ for accelerometer and gyroscope bias, respectively. It has the same units as the sensor to which it refers, i.e. °/s for the gyroscopes and ms^{-2} (or milli-g) for the accelerometers. Typical gyroscope bias specifications for current consumer-grade MEMS are from ±1 °/s [45] to ±5 °/s [23] although some are much higher ±25 °/s or ±75 °/s ([46] and [24] at ±2000 °/s range). Typical accelerometer biases are ±60 mg (milli-g) [46] to ±70 mg for [45], for the 2 – 16g ranges used for most navigation applications.

The next class of errors are first-order or linear errors. One of these is the scale factor error where the conversion factor from sensors' output (integer or voltage) to physical units is incorrect [1], also known as "sensitivity tolerance" [45]. Typically, the sensor's output might be, for example, 98% or 103% of the input. Another 1^{st}-order error is the cross-coupling or cross-axis sensitivity. An example of this would be if an input force or rate on the x-axis were picked up by the y-axis sensor. The cause of this could be physical misalignment of the whole inertial triad, or one sensor of the triad. There are other causes, particularly for the gyroscopes where the vibrating frequencies of the different axis sensors may interfere with each other. In this thesis these errors are combined in one 3×3 (unitless) matrix for each of the gyroscope and accelerometer, named $\mathbf{A_g}$ and $\mathbf{A_a}$, respectively. This error is assumed to be linear, and thus the same throughout the operating range. The entry $a_{ij} \in \mathbf{A}$ representing the multiple of the true specific force or angular rate of axis j that is measured on axis i. Thus, for a perfect sensor $\mathbf{A} = \mathbf{I_3}$. ±1% [45] to ±2% [23, 25] are fairly typical cross-coupling errors for consumer-grade MEMS. Typical scale factor specifications often include temperature variation, for example "sensitivity tolerance ±1% at 25°C " and "±0.03%/K through operating range −40 to 85°C (gyroscope [45]) or "±3% initial tolerance" and "'±0.026%/°C " (accelerometer [23]).

The operating mechanism of vibratory gyroscopes makes them inherently susceptible to specific force [1], although they are usually designed to minimise this by, for example, having

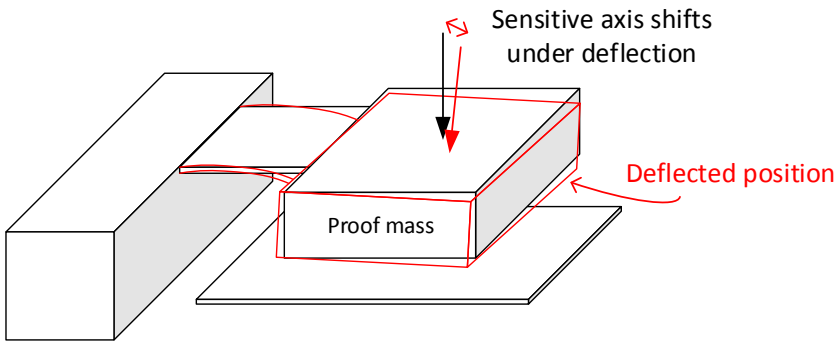


Figure 2.4: Illustration of accelerometer sensitive axis shifting when deflected. This is one potential cause of scale factor non-linearity and anisoinertia errors.

a pair of sensing elements and taking the difference between them [20], as was discussed in Section 2.2.2. This may not be achieved perfectly leaving a small effect of specific force on the gyroscope. This is called g-dependent bias or g-dependent error; this thesis denotes it by $\mathbf{G_g}$. Gyroscopes can also exhibit g-dependent scale factor errors [1], caused as applied specific forces cause the proof mass of the gyroscope to move, although these are often not specified for consumer grade sensors, see Section 2.2.6.

Higher order errors, i.e. 2^{nd} or greater, are only rarely given for low cost MEMS, e.g. [47]. For automotive-grade sensors, these are generally lumped into a single term called non-linearity, or sometimes scale factor non-linearity, which is a more accurate description. This error describes the variation in the scale factor throughout the range of output. A typical specification sheet might read "0.5% of Full Scale Output" [23]. In general non-linearity gets worse at the extremes of the sensor's output range [47]. This error is discussed in more detail in Section 2.2.6. Another higher-order error, which is frequently not specified for low-cost MEMS is the anisoinertia error, which is the variation in cross coupling depending on applied force and is a particular problem for open-loop pendulous accelerometers [1]. An illustration of how might occur is in Figure 2.4.

A source of error which does not fit well into the n^{th}-order characterisation is hysteresis. This is arises from frictional losses in the spring elements of the sensor [20] and its effect depends on whether the specific force is increasing or decreasing [4].

Vibratory gyroscopes are also sensitive to applied angular acceleration (see Equation 4.7 of [1])

The largest amount of specific force or angular rate that an inertial sensor can measure is described as its range, measurement range or full-scale. This is usually written as, for example $\pm 2g$ or $\pm 250\,°/\text{s}$. Typically when exposed to inertial forces greater than they can measure, the sensors output their maximum value. This is sometimes called saturating the sensors, or as clipping because of its effect of removing the peaks of the output. If the inertial sensors signal is clipped large navigation errors can result [1].

It is important to note that for many sensors many of the error sources may be affected by the sensor's mounting, both the heat cycling involved in soldering and the internal stresses

applied by their multi-point (i.e. overdetermined) mounting [20], as the sensor's FLP will be attached to the circuit board by all its pins. Sensors error characteristics have been shown to change markedly (e.g. bias by $> 50\,°/s$ in [48] p25–26) when the board they are mounted on is subjected to external stress. For this reason manufacutrers recommend 3-point attachment to housing for the PCB on which the sensor is mounted, e.g. [23]. For these reasons, wherever possible the calibration should always be carried out after construction and in the final IMU housing.

Further details on IMU errors can be found in books such as [1, 4, 20].

The next two sections discuss the validity of and some implications of the simple linear error model typically used for low-cost MEMS IMUs (Section 2.2.6) and the statistical distribution of its error coefficients (Section 2.2.7).

2.2.6 The linear plus non-linearity IMU error model

When considering a low-cost IMU generally only a very simple error model is assumed, where the measured specific force,

$$\widetilde{\boldsymbol{f}_{\boldsymbol{ib}}^{\boldsymbol{b}}} = \boldsymbol{b_a} + \mathbf{A_a}\boldsymbol{f}_{\boldsymbol{ib}}^{\boldsymbol{b}} + \boldsymbol{w_a} + n\left(\boldsymbol{f}_{\boldsymbol{ib}}^{\boldsymbol{b}}\right) \tag{2.1}$$

and the measured angular rate,

$$\widetilde{\boldsymbol{\omega}_{\boldsymbol{ib}}^{\boldsymbol{b}}} = \boldsymbol{b_g} + \mathbf{A_g}\boldsymbol{\omega}_{\boldsymbol{ib}}^{\boldsymbol{b}} + \mathbf{G_g}\boldsymbol{f}_{\boldsymbol{ib}}^{\boldsymbol{b}} + \boldsymbol{w_g} + n\left(\boldsymbol{\omega}_{\boldsymbol{ib}}^{\boldsymbol{b}}\right). \tag{2.2}$$

Here the sensor output is a combination of a fixed bias term ($\boldsymbol{b}$) and a term linearly depending on the output ($\mathbf{A}$) plus noise ($\boldsymbol{w}$), and in the gyroscopes' case linear error due to specific force ($\mathbf{G_g}$). All other error sources are lumped together into a catch-all non-linearity term, n which is a function of the inertial input, or assumed to be small enough to be ignored.

None of the other error sources from Section 2.2.5 are mentioned on the datasheets of many consumer-grade MEMS sensors, including [23–25, 45, 46, 49–52]. In addition, many of the gyroscope specifications do not mention $\mathbf{G_g}$, e.g. [23–25]. Exceptions where sensor manufacturers specify $\mathbf{G_g}$ include: [45] which gives it as "maximum 0.1 °/s/g", where maximum "represents 3σ values"; [53] which gives "typical 0.1 °/s/g" without defining typical; and [54] which gives it as "0.075 °/s/g any axis 1σ".

There are two ways to interpret this linear-plus-deviation-from-linear error model used in the specifications. First, optimistically, that this provides an adequate model and other types errors have an insignificant effect. Second, cynically, that these error sources are the easiest ones for the customer to verify. The important question is whether this linear model is a good fit to the input-to-output relationship.

Often $\boldsymbol{b}$ and (the diagonal elements of) $\mathbf{A}$ have a specification of their variation with temperature. This is sometimes specified as a range over the whole operating temperature range, e.g. ±1.7% accelerometer sensitivity (i.e. scale factor) variation over temperature -40 °C to +85 °C [51], or as a linear function of temperature, e.g. $\pm 0.015\,°/s/K$ gyroscope bias change [45]. When the temperature sensitivity is given as an apparently linear function, there is an optimistic way to interpret it, namely the temperature response is almost linear

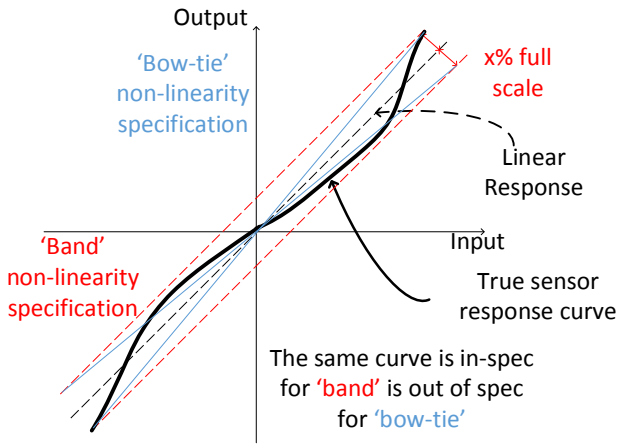


Figure 2.5: Two ways to interpret non-linearity specifications.

with a fixed but unknown coefficient, and a pessimistic one, namely that the change from one degree to the next is not more than 0.015 °/s (in this example) but the amount is not fixed. Where experimental evidence exists, e.g. Section 6.6.2 of this thesis or [48,55], the truth lies somewhere in the middle; temperature response generally being shown to be a continuous curve but not completely linear. One might consider a less optimistic interpretation likely in cases where only the variation over operating temperature range is given.

The catch-all non-linearity term is usually given as a percentage of full-scale output. For example, "best fit straight line ±0.05 % FS" [45]. Although it is occasionally given in physical units. For example, the Bosch SMI130 [51], gives non-linearity as ±1 °/s and ±25mg (milli-g). Given that the scale factor terms are typically specified to vary by as much as 2% over the measurement range, it would be reasonable to expect the non-linearity (i.e. the goodness of fit of this linear scale factor term) to also vary significantly with temperature. This is shown to be the case for at least one type of low-cost MEMS sensor in [56], where the error goes from positive to negative. One issue with this way of specifying the error is how to interpret the result. One way would be to interpret it as an amount by which, throughout the range, the sensor will repeatedly report incorrectly by, with the same input, this might be thought of as a band-like error. This is a reasonable explanation for the physical units case above. Another way might be to interpret the number as the amount by which it may be wrong at the extreme ends of its measurement range, assuming the linearity error to be lower for lower inputs. Almost equivalently it might be considered as the range of error in the slope of the input-to-output function. These are both a kind of bow-tie shaped error on the input-to-output function. Sometimes it is reported as "best fit straight line" (e.g. in [52]), in which case the former of the two explanations is more reasonable. Both a band and a bow-tie error are illustrated in Figure 2.5, note that the band specification is more generous and thus may be more likely.

The evaluation of higher order errors is rarely reported for low-cost MEMS sensors. It is more frequently reported for higher performance sensors. For example, [57] reports a high-performance open-loop accelerometer and states it has a very close to linear response with non-linearity of $< 10\mu g/g^2$. Similarly [47] reports linearity error in terms of input squared and cubed.

However, in one example of a low-cost sensor's non-linearity from [56], it appears to be closer to piecewise linear than quadratic in response. This means it has an almost linear response but with a different slope for the lower magnitude responses than the higher magnitude. However, without conducting independent tests, there is not an alternative way to characterise the errors.

It is important to consider that there are different ways to fit this error model given to an input-output curve. The standard definition for bias implies that it is the y-intercept, however, if the response is not completely linear then the best-fit line will not, in general, intersect this point, unless forced to. Additionally even where the non-linearity (and thus the scale factor error) is defined as "best fit straight line" (e.g. [23]), the actual straight line will depend on the finite number of input values tested and also on the fitting objective function (e.g. least squares or magnitude of residuals). This has implications which will be discussed further, in Section 6.6.1.

2.2.7 IMU error model coefficient distributions

If this error model introduced in Equations 2.1 and 2.2 is assumed, then to use it there must be coefficients for the terms in it. That is, there must be values for $\boldsymbol{b_g}$, $\mathbf{A_g}$ etc. Over the population of sensors these coefficients must have some distribution, which may differ from one particular production batch of sensors compared to the overall population of sensors. Note that the manufacturers' specifications are written to cover the whole population of sensors.

In terms of noise for low-cost sensors, there is usually only a single number specified to describe the noise. For example, a rate noise spectral density of $0.01\,°/s/\sqrt{\text{Hz}}$. This single parameter specification of noise means that realistically the manufacturers are expecting users to use a simple white-noise-only stochastic model. For higher performance IMUs, other stochastic terms might be mentioned, e.g. "In-Run Bias Stability" is specified by [58] at $< 10°/\text{hr}$, but for low-cost IMUs these terms are generally not specified. In a similar way nearly every IMU systematic error is only specified by a single number, and frequently only typical values are given. One such as scale-factor might be expressed as e.g. sensitivity minimum 230, typical 256 and maximum 282 LSB/g (from [49]), "typically" $4096 \pm 1.5\%$ LSB/g (same characteristic from [52]) or "typical" sensitivity scale factor tolerance of $\pm 3\%$ [23].

The meaning of these terms are usually as follows: "Min/Max values represent 3-sigma values. Typical values are not guaranteed." [52]. Although sometimes on or both of these terms are not explicitly defined.

As only a single number describes each error term, the most simple model is to assume either a uniform distribution or a zero-mean Gaussian. The latter is the general assumption in this thesis, particularly in Chapter 3 and 4, and additionally it is assumed typical and min/max refer to 1σ and 3σ values, where 1σ refers to 1 standard deviation. A symmetric distribution is implied by the datasheet using only a single value, and without conducting one's own testing there little choice but to assume one of these is the case. If the manufacturers are

testing and either rejecting significant numbers of sensors or selling the better ones as a higher grade, i.e. 'cherry-picking', then other error types of distributions are more likely.

One example when the manufacturer assumes a normal distribution is [50], where they also use the definition of typical and min/max as above.

The real stochastic model may be more complex, see [43, 44, 54, 55] although higher order terms are frequently not modelled in integration filters (see Section 2.4.1) for low cost sensors, replaced by exaggerating (overmodelling) the amount of noise [1].

While it is assumed, particularly in Chapters 3 and 4 that this error model is a viable assumption and the coefficients are distributed with zero-mean Gaussian distributions, there is no way of verifying this without large-scale tests which are beyond the means of this project. In Chapter 6 ideas will be presented which can improve the performance of IMU arrays when this simple symmetric distribution of coefficients is not the case, and experiments test whether a particular sensor model has a simple symmetric error distribution.

2.3 The INS as part of a multi-sensor navigation system

In the previous sections, inertial navigation has been introduced. The problems of inertial navigation have been discussed, primarily solution drift with time caused by the accumulation of IMU errors [1].

All navigation technologies have both strengths and weaknesses. There may be situations where a particular technology does not work at all (e.g. GNSS in a tunnel), or suddenly becomes unreliable for a few seconds (e.g. compasses/magnetometers near concentrations of ferrous metal) [59]. However, by combining several navigation technologies into a multi-sensor navigation system, the strengths of one technique can ameliorate the weaknesses of another. When done correctly this can lead to better navigation performance than any of the systems can achieve alone.

In general, an integrated navigation system has better performance the more complementary navigation technologies are added to it. Even technologies which only measure one part of the navigation solution, such as a speed-sensor, magnetometer or barometer, can aid an integrated navigation system and increase overall performance. However, if no new measurements are given to a navigation system it typically just coasts, assuming the user continues to move with their last known velocity. Thus, a navigation system will benefit greatly from including at least one system that is available at all times. Most dead-reckoning technologies can work well in this role. In vehicle applications a vehicle speed sensor (e.g. odometer or ship's log) and heading from a compass, yaw-gyroscope or gyrocompass can provide continuous two-dimensional position. Similarly pedestrian dead reckoning (see Section 2.3.2.3) with a heading sensor can achieve better performance than than standard inertial navigation [60], particularly if uncalibrated low-cost IMUs are used. However, in general, inertial navigation is an excellent fit for this application for several reasons: it provides a 3D PVA navigation solution; this solution is updated frequently (50–1000Hz); its availability does not depend on the operating environment; and, it can be used in virtually any navigation

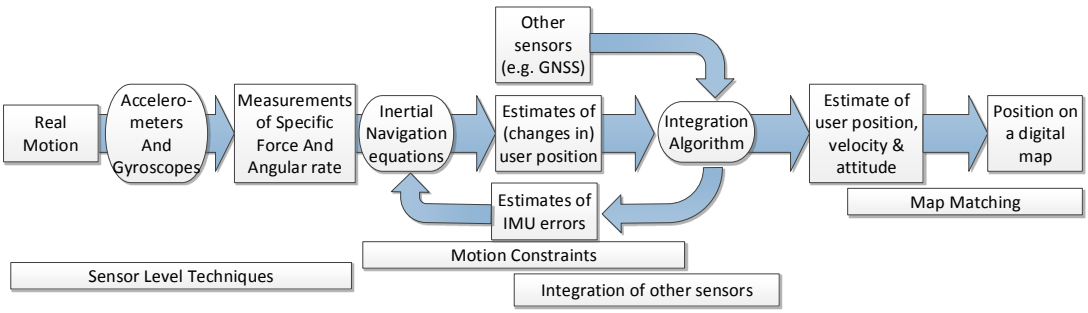


Figure 2.6: The navigation system processing chain. The positions in which different techniques to improve navigation performance fit are underneath.

application, provided suitable sensors are chosen.

One common integrated navigation system architecture is for one technology, in this case INS, to be at the heart of the system, providing a solution which is updated by the other sensors when a new measurement becomes available via some kind of integration algorithm (see Section 2.4). The process by which an INS-based integrated navigation system works is illustrated in Figure 2.6. It shows how the various techniques used to improve navigation performance fit into the chain. For example, the integration of GNSS (Section 2.3.1) or other sensors (Section 2.3.2) fits in between the IMU derived estimates of (changes in) user position and, by way of an integration algorithm, the information from the other sensors is used to come up with a combined estimate of position [1]. Motion constraints are discussed in Section 2.3.3 and map matching in Section 2.3.4. The sensor level techniques to improve IMU performance, IMU calibration and using arrays of IMUs, will be discussed in Section 2.5.

This section starts by considering integrating different sensor types into a navigation system. In evaluating any technology for inclusion in such a system there are a number of factors to consider. The main concern is whether an addition to the navigation system will provide enough improvement in performance to be worth the increase in size, weight, cost and computational complexity. Essentially an additional navigation technology must provide an increase in accuracy, an increase in availability or measure a new physical quantity. If the new sensor is measures the same thing more accurately and is more available (in all senses of these words), it might be a better choice to have it replace the previous technology rather than add to it.

The accuracy of the technique is often the simplest to understand. If two techniques measure the same thing, their accuracies can at least be quantified. For example, eLoran, which could provide positions accurate to 8–20 metres [61], is more accurate than the obsolete Loran-C which had an absolute accuracy from 0.1 NM (nautical miles) to 0.25 NM (165 – 463 metres) [62]. Note that the actual accuracy of any technique may vary between ideal conditions and what is actually achieved in use.

The availability part of the evaluation is more complicated, as there are different kinds of

availability. One type of availability is similar to bandwidth, i.e. how frequently is a new position available. For some applications, such as automatic guidance systems or auto-pilots, new PVA navigation solutions need to be available at high frequencies. A navigation system for this type of application, might well include a combination of a low-performance high-update-rate system, which could be corrected by an accurate but less frequently updating system, to create an accurate but high bandwidth system. It is important to note that some systems may be able to output measurements more quickly than their true bandwidth, for example the errors on sequential measurements may be significantly correlated. The frequency of truly new information is what is important for the navigation system.

Availability may also depend on operating environment, e.g. GNSS unavailable or degraded inside a building or urban canyon [63], or might depend on the motion, e.g. zero-velocity-updates (ZVUs), or perhaps both. If several technologies are to be combined to improve IMU performance, then the combination of their availabilities should be considered, e.g. WiFi and cell ID (GSM) positioning might not cover all possible operating environments. It is also likely that if, for example, several radio-positioning technologies are used the situations in which they are unavailable might well coincide. Another important consideration with some intermittently available technologies is, how reliably one can detect that it is available. For example, compasses/magnetometers can provide a heading measurement all the time. However, in the presence of strong magnetic fields or concentrations of metal, the derived heading is unreliable. Thus their usefulness to the navigation system relies on these instances being filtered out.

The other major consideration is the type of measurement provided by a particular technology or technique, and the frame of reference this is in respect to. A MEMS IMU measures angular rates and accelerations in a body reference frame. Many other dead-reckoning technologies also use the body reference frame, e.g. odometry. This means that the attitude must also be known to calculate a navigation solution in a global reference frame. For many position-fixing technologies the measurements are with reference to the position of the beacons, which are defined in a global reference frame, e.g. GNSS with relation to the network of satellites.

Generally speaking, the closer measurements are to the most important part of the navigation solution (often position) the better. If one cares about position then the fact than an INS drifts away from the truth (see Section 2.2.3) makes the INS less useful than a position fixing technology in the long term. On the other hand, if one wishes to calibrate an IMU, then measurements from sensors such as odometry and magnetometers, which provide speed (in the body frame) and heading, may make it easier to observe the IMU parameters than indirectly estimating them based on a sequence of positions in a global reference frame. Although even better than this for calibrating an IMU would be a higher-performance IMU to provide a higher accuracy version of the same measurements at all times. This might be in a laboratory calibration (see Section 2.5.1). However, note that while the low-quality IMUs errors could be estimated, there would be no performance improvement from including both in an integrated system. This is because the low-quality IMU would produce only a lower-quality measurement of the same quantity and so no useful new information, unless the intention is to use the lower-quality IMU independently after the calibration.

The next few sections present some of the sensors that can be added to a multi-sensor integrated navigation system, which is based on an INS. These sections concentrate on low-cost applications to remain relevant to this thesis. INSs are characterised by low short-term noise but a deviation from the true position, or drift, that increases through time. Combining an INS with another positioning technology whose errors behave differently can be very effective in increasing the performance of the overall system.

2.3.1 Global Navigation Satellite Systems (GNSS)

Global Navigation Satellite Systems [1, 64] refers collectively to several different satellite navigation systems: the US Global Positioning System (GPS), the Russian GLONASS system, the Chinese Beidou system and the European Union's Galileo system. While these systems are not identical, fundamentally, they operate in a similar manner:

A number of satellites, typically 20–30 per constellation, orbit the globe at medium Earth orbit (MEO, $\approx$ 20,000 km). Each satellite contains radio receivers (for receiving messages from the control stations), very accurate atomic clocks and powerful broadcast radio equipment, in addition to the necessary power and guidance systems. These satellites' positions and orbits are monitored by a network of reference stations whose positions are accurately surveyed. This includes the control segment which uploads the broadcast message to the satellites (space segment). The satellites then each broadcast a message down to the Earth, where it can be received by user-equipment. This message contains the current time and information about the satellite's orbital path.

The users of the GNSS system each have a relatively cheap radio receiver and clock, this radio receiver de-codes the message that the satellite has broadcast. The message contains both a ranging code and information about the satellite's orbital path. Then the receiver can work out where the satellite was at the time that the message was broadcast and thus knowing the time that the message was received and the speed-of-light-in-a-vacuum works out a *pseudo*range to the satellite. This pseudorange is corrected for a number of factors including the delay added by travelling through the atmosphere to make an estimate of range to the satellite.

If the receiver's clock was perfect, the range to three satellites would be sufficient to fix the user's position. A range to one known point gives the position as being somewhere on the surface of a sphere, two known points gives the intersection of two spheres: a circle, and a third range locks the position to two points (one of which can usually be eliminated as being in space or deep inside the Earth). However the receivers usually use cheap and inaccurate crystal oscillators. This means that there is a further unknown: the receiver-clock-offset. This must be determined by a *fourth* ranging measurement. As there are four unknowns, four measurements are required to determine them.

The quality of the position-fix from GNSS is improved by more ranging measurements (i.e. receiving signals from more satellites) and affected by many factors including the antenna used, the local environment, the weather and the quality of the orbit model that the satellite broadcasts. However, it is always possible (assuming at least 4 satellite signals

can be received) to calculate a position based *only on the instantaneous measurements*, this means that while the short-term position uncertainty may be several metres, this does not increase with time. In this way, GNSS positioning is very different to a dead-reckoning technique such as inertial navigation.

There are a number of factors which influence GNSS performance including: signal environment; antenna design; filter algorithm design (in standalone GNSS use); and signal geometry [64]. Poor signal geometry is when the signals the user receives are distributed unevenly, such as mostly from one direction or azimuth, this poor geometry creates an uneven distribution of position uncertainty in three-dimensional space. For example, GNSS is always, except in orbit, less accurate in height than in latitude and longitude because all the satellites that can be received are above the horizon.

It should be noted that GNSS can have much worse performance in urban areas where signals may be blocked, reflected or refracted, and the signal geometry may well be less than ideal [63]. In addition the signals are quite weak which makes GNSS vulnerable to accidental interference or deliberate jamming [65, 66], and it has also been shown to be vulnerable to spoofing [67].

Advanced GNSS positioning techniques can exploit corrections calculated from a network of reference stations, which are then transmitted to the user via radio signals or the internet. The density of the reference station network determines, to a large extent, the accuracy of the corrections [1]. Delay from the signal passing though the Ionosphere can be corrected by exploiting the different propagation characteristics of GNSS signals at different frequencies, these can be used by dual-frequency GNSS receivers to create an isonosphere-free observable. The GNSS carrier phase can be used to smooth the range measurements. However, these provide an ambiguous ranging measurement, which must be resolved by ambiguity resolution techniques (e.g. [68]). These can be used for high precision static positioning, see e.g. [69], but that is out of the scope of this thesis. They can also be combined for accurate kinematic positioning [70,71]. These advanced GNSS positioning techniques are not considered in this thesis, mostly because consumer-grade GNSS user equipment is single frequency.

In general, GNSS provides measurements of the user's velocity and position relative to a global reference frame. However, it is possible to use a multiple-antenna GNSS receiver to calculate differences in arrival times of GNSS signals at the different antennae. From this measurement, a fairly noisy but non-drifting attitude solution can be calculated. Three antennae are needed for a full attitude solution but a two antennae system would still be able to aid a navigation system. This can be used for INS calibration as in [72]. It is important to note that while a full GNSS position fix requires signals from 4 satellites, a GNSS attitude fix requires only two satellites, although more signals improve its accuracy. Higher precision can be obtained by increasing the antenna separation but this is limited by how far apart the antenna can be while remaining sufficiently rigid. GNSS attitude can be used to compare the measured angle of arrival of GNSS signals with that expected from the approximate position and the satellite orbits. This has been demonstrated as a spoofing countermeasure [73], and could also be used for detecting non-line-of-sight (NLOS) signals.

GNSS is frequently integrated with inertial navigation as it is an effective complementary system to an INS. This means that even intermittent GNSS position fixes can greatly improve the longer term navigation performance by removing accumulated position drift. There are different ways in which the GNSS data can be fused with the IMU data for an integrated solution. In most cases, the earlier in the processing chain the integration happens the greater the potential benefits but the more complex the scheme is to design and more computationally expensive to run. Methods use to combine GNSS and INS measurements are classified (here) according the domain in which the measurements of the two systems are combined following the naming conventions of [1].

In a loosely-coupled integration scheme the data is fused in the position-velocity domain. This means that without using the IMU data, the GNSS receiver determines a position fix at each epoch. This is then compared to the position calculated by an integration of all the IMU data compiled since the epoch of the previous GNSS position fix. This scheme is the simplest to design and the most modular, so if there are several different navigation technologies which might be turned off and on, integrating these different technologies is easier. It is also the only suitable option if one has limited access to the measurements made by the GNSS chip (for example if pseudoranges are not output), as it can run using only the position data. However a disadvantage of this scheme is that if less than four satellites can be received, then GNSS cannot generally be used to aid the navigation.

One of the alternative integration schemes is the tightly-coupled integration scheme. In this scheme the measurements from different position fixing beacons are considered separately, for example as ranges or lines of position. This means that GNSS pseudorange and pseudorange-rate measurements from each satellite are combined with the IMU measurements before the GNSS measurements are used to compute positions. This is more complicated to program but has greater potential performance, as even with a reduced number of satellites, corrections can still to be made to the INS solution, albeit less effectively than when there are the minimum number for a single epoch position fix (that is, 4). It also allows in-homogeneity in the position uncertainty from GNSS (e.g. from poor signal geometry) to be reflected in the integrated position uncertainty.

Another possibility is deeply-coupled integration. In this case the INS integration and the signal tracking are performed by the same algorithm. This has a much more complicated system architecture because it works with the GNSS signal correlator outputs instead of with pseudoranges [74]. The advantage of deep integration is that the sensitivity of the tracking loop can be increased to allow operation in poorer reception environments without losing the ability to respond to movement. It can also help signal reacquisition [75], as the INS helps compensate for the dynamics of the antenna. Potential disadvantages of deep-coupling include this not being possible with most low-cost commercial GNSS chipsets and being even more complex design than tightly-coupled integration. Also, without careful algorithm design, poor IMU performance (e.g. from vibration conditions) could potentially cause poor GNSS signal tracking.

The actual algorithm used to integrate the different sets of measurements is usually a Kalman filter variant. This shall be described in Section 2.4.

2.3.2 Integration of other sensors

While INS/GNSS is a classic example of a multi-sensor integrated navigation system, there are many other navigation technologies that could be used.

There is an inherent trade-off with integrating other sensors. Each sensor that is added to the package adds some size, weight, power consumption and cost. There is an additional processing load to integrate all the information, and most sensor technologies will require some sort of calibration. Therefore the type of sensors worth including in a given system, depends on the performance improvement they can provide in the specific application.

2.3.2.1 Magnetometer integration

A magnetometer is a sensor which detects magnetic fields. In navigation it is usually used to detect the Earth's magnetic field (which has a strength of 0.5–0.6 Gauss according to [76]), a 3-axis sensor is required in nearly all cases. As they can be used to determine heading, from the Earth's magnetic field, they are sometimes called electronic compasses. MEMS-scale 3-axis magnetometers are available at a low cost, both standalone [77], and as part of so-called 9-axis IMUs [23, 25, 45].

Integrating a magnetometer with an inertial navigation system is helpful because it provides a non-drifting measure of heading, unlike a rate-gyroscope which keeps track of attitude by dead-reckoning and therefore accumulates errors during operation. In the short-term a typical magnetometer-derived heading is more noisy than that from a typical MEMS gyroscope, thus in combination they are complementary: the magnetometer can correct the drift of the gyroscope.

In order to use a magnetometer is necessary to tilt-compensate it. During tilt compensation the 3-axis readings from a 3-axis magnetometer are converted into readings in the horizontal plane. This means that at least an approximate value for the pitch and roll parts of the attitude solution is needed. As the direction of the local vertical is approximately observable from the specific force output of a 3-axis accelerometer. MEMS accelerometers are often used for this, and so the two are sometimes available as a single component [78]. However, if accelerometers are used then large accelerations can corrupt the calculated heading.

Additionally, magnetometers are subject to biases caused by ferrous metals or current flows in the IMU and its housing. Although these can be compensated for by a method detailed in [76], where they are split into hard-iron (fixed offsets) and soft-iron (heading dependent offsets). This calibration need only be carried out once, and is simple enough to be done by the user (for a hand-held device), as it only requires that the magnetometer be rotated a full 360 degrees.

More significantly, large ferrous metal objects (such as filing cabinets or lampposts) can cause significant distortion of the Earth's magnetic field nearby (1 to 5 metres). Therefore when using a magnetometer for heading determination, it is always necessary to run an algorithm to determine if there is a magnetic anomaly. At times the magnetometer heading

will be unusable (when an error is detected) and even when available, it will not always be completely reliable (when an anomaly is missed by the detection algorithm).

An Attitude and Heading Reference System (AHRS) is similar to an INS combined with a magnetometer. However it only keeps track of attitude, rather than attitude, position and velocity, and as such only needs to carry out the first attitude-update part of the inertial navigation equations. It still needs the accelerometers to tilt-compensate the magnetometers. These have existed for several decades in nautical and aeronautical use and can also be used for correcting gyroscope drift in an INS as in [79].

2.3.2.2 Integration of Barometric Altimeter

Barometers measure air pressure, from this measurement height can be inferred. However, this exhibits a weather-dependent bias (and scale factor error). This bias can either be calibrated with reference to, for example online weather reports, or it can be used as a measure of change in height. Barometer-derived height measurements also exhibit short term noise (at around $\pm$1m)in addition to the bias described above and this needs to be accounted for in the integration filter.

A further problem with using barometers is that, because the estimated height is also affected by temperature and humidity, in applications when they might travel from indoors to outdoors and back (especially in winter), the behaviour of the altimeter becomes erratic [80]. It would be possible to store two (or more) sets of barometer calibration parameters which could be switched between, if a change of operating environment was detected by an algorithm. This would allow a rapid return to stable values if the user is moving between, say indoors and outdoors, although the context detection adds additional complexity [10].

Barometers are also built into many pedestrian navigation systems and activity monitors (pedometers), where they are used for detecting floor changes. They are also used in sports GPS watches, e.g. Suunto Ambit3 vertical [81], where they are used to smooth the higher vertical position uncertainty of GNSS.

2.3.2.3 Vehicle odometry and other measures of speed

There are various kinds of sensors and techniques which can provide measures of speed or velocity, these include vehicle odometers, ship's logs and radio frequency (RF) techniques which give range-rates (e.g. [82]).

There are several different kinds of measurements that might be provided such as speed (i.e. the magnitude of total velocity), velocity in a body reference frame, velocity in an external reference frame, or a range-rate, that is an increase or decrease in the range to a radio beacon, which is a measure of speed towards or away from a particular point. Any of these techniques could be 2D or 3D.

In general, a speed measurement alone is not sufficient to position a user, but if other sensors such as an IMU provide a PVA solution at all times this is not such a concern. The measurement of speed is in one way superior to the acceleration derived from accelerometer

readings because it only needs to be integrated once, rather than twice, to provide position information, and thus does not drift from the truth so quickly.

All road vehicles have an odometer in order to measure the distance travelled and current speed of the vehicle for the driver. Aircraft have air-speed sensors, and watercraft have ship's logs. All of these provide similar information, and all may require some sort of calibration, but this is, certainly in the case of road vehicles simple enough to be carried out in-run. However, there is the additional issue for water and air-craft that the fluid medium through which they travel is frequently itself moving, in a manner which will not be known perfectly.

Odometry combined with even only a yaw gyroscope (and/or magnetometer, see Section 2.3.2.1) has been used to run a a short term dead-reckoning navigation solution to bridge occasional GNSS outages (e.g. bridges or tunnels) with reasonable accuracy [21, 83].

There are a number of radio frequency (RF) techniques which provide measures of speed which may be useful for calibrating or removing drift from the INS, most of which provide similar information to GNSS (which also provides Doppler or phase-range-rate measurements). For specific applications, contexts or locations some may well be superior to GNSS. For example, Doppler-derived range-rate measurements to RF beacons may well be available significantly more often in urban or indoor locations than GNSS (e.g. [82]). However, each needs to provide better information than GNSS, in the sense of more available or more accurate. In general most of these will be more noisy than odometry but that only applies to some vehicles.

An equivalent measure of speed can also be applied to pedestrian motion, step detection with an IMU rather than (or sometimes in addition to) a full set of inertial navigation equations [60, 84, 85]. This is sometimes known as pedestrian dead reckoning (PDR) although this also requires a gyroscope and/or magnetometer to determine heading. This has a number of advantages over using inertial navigation, the main one being reduced accelerometer accuracy requirements, as the step detection algorithms are not so affected by, for instance, biases and scale factor errors. However, the step detection algorithm's accuracy may depend on the navigation IMU's position on the body and the step length may vary slightly from step-to-step, between people and between different kinds of motion, e.g. walking, running, climbing stairs or crawling. This means that for it to be accurate also requires a context detection algorithm [10, 86, 87].

2.3.2.4 Camera Integration

Combining a camera into a navigation system is an option as relatively inexpensive cameras have improved dramatically in performance in the last few years, as they are included in every smartphone. It is interesting to note that one of the other main applications of MEMS gyroscopes is image stabilisation in cameras.

Existing image based navigation systems include both dead-reckoning and feature recognition techniques [3]. These techniques can add considerable additional information to an integrated navigation system.

For dead-reckoning, the idea here is that a video feed (effectively a high frequency sample of still images) can be used to estimate the change in position of the video camera between epochs [88]. This means that they are vulnerable to the same kinds of errors as other dead-reckoning technologies, but the sources of their errors are not likely to be coincidental. For a start, cameras are better at tracking angles or angle-changes than linear displacements. This contrasts with INSs where the long-term performance is typically dominated by the gyroscopes. Therefore an INS and camera are a complementary combination. An an INS's gyroscope could be calibrated in-run and could then bridge instances where useful features cannot be seen by the camera.

There is a problem of scale-ambiguity when using only the information provided by a single camera. There are several ways to compensate for this, which include: accurately knowing the distance from the ground and orientation of the camera (in the body frame) which could work for vehicle applications; recognising the size of some objects (stadiametric ranging); and having two cameras a fixed distance apart (stereoscopic ranging) [3]. Alternatively the ambiguities can be resolved by integration of other sensors, e.g. [89, 90].

There are two main techniques used to perform dead reckoning with cameras; these are optical flow and feature tracking algorithms [1, 3].

Optical flow involves tracking the light intensity over a grid and trying to match this intensity to a particular amount of movement (rotation and translation), using a correlation function, between frames. The results of which can be interpreted as a position and attitude change (see [91]).

Feature matching techniques detect features of each image using one of several algorithms, these features can be lines or (ideally) corners (features which show considerable variation both horizontally and vertically) then these features can be tracked across several frames to determine relative movement between frames.

Optical flow algorithms are in general less computationally intensive and less robust. However the reduced computational load allows the images to be processed more frequently (for a given computational resource) and so the movement between frames will be smaller, which slightly reduces the robustness penalty. However, for any given algorithm there is a trade-off between the robustness (i.e. sensitivity to false readings) and the computational load. This can be reduced by good compensation for camera effects (such as flare).

There is potential to reduce long term INS attitude drift by using line recognition algorithms and exploiting the fact that in the (especially indoor) environment most lines are horizontal or vertical. [3] noted that considerable improvement can be made to optical dead reckoning systems through sensor fusion with a low-cost MEMS IMU because the approximate position change reduces the number of possible combinations that need to be searched in either optical flow or feature-matching algorithms, in a tightly coupled system, and reduces the ambiguity problem.

Camera integration can also be used to recognise landmarks, providing a way to correct a drifted position, see [92, 93]. A camera can be used to build up a simultaneous localisation and mapping (SLAM) database of particularly distinctive optical features matched to

locations. This map can then be used to correct the position drift of any dead-reckoning navigation system by loop-closures, that is by recognising when the user has returned to a previously visited location. Clearly, this is only useful in applications where the user returns to previously mapped locations.

However, there are two main performance limitations using cameras.

The first is hardware. The camera has a finite number of pixels, this (in combination with the focal length) determines the smallest angle that can be read (angular resolution). A longer (effective) focal length allows finer angle resolution but a smaller field-of-view, and thus less potential to find features. There are optical limits as well whether the quality of the lens (and the autofocus) can reduce the ability to determine individual pixels.

The second is data. A video stream is a lot of data, certainly relative to the output produced by virtually any other navigation technology. Moving, processing and analysing that much data can a serious computational challenge. This leads to much higher power consumption and potentially more expensive computational hardware. For taking photos and recording video streams, application specific integrated circuits (ASICs) exist. That is, dedicated computational hardware which reduces the power consumption and computational overheads for those specific tasks. They do not exist for camera navigation techniques. In addition to any navigation algorithm there is a lot of processing that needs to be done to create a good video stream, such as compensating for changing light conditions.

It should be noted that camera integration is not available all the time. There must be sufficient light and visual variation in the camera's field-of-view. Thus it is best used as part of an integrated navigation system.

2.3.2.5 Other radio frequency technologies

There are several ways other radio-positioning technologies can be used for multi-sensor integrated navigation systems. Most of these fall into a few groups. There are positioning technologies that exist to provide a terrestrial-based back-up to GNSS in safety critical applications, such as eLoran [61] or DME [94]. These technologies generally provide a less accurate position than GNSS but may be available in situations where GNSS is not. However the user-equipment to position using one of these technologies is often prohibitively large for applications that use low-cost MEMS IMUs.

There are positioning techniques that use signals broadcast for purposes other than navigation, these can be collectively described as signals of opportunity. There are some of these from which the user can be positioned, such as AM-radio [95] and airliners' beacons that broadcast their position [96].

There are also techniques based on short-range beacons such as bluetooth, RFID or WiFi [97, 98]. The most common implementation is a simple proximity measurement compared to a database of beacon locations although received signal strength may also be used. Fundamentally these methods give a position measurement, rather than velocity. This makes them not particularly suited to calibrating IMUs in-run, although they will clearly aid an INS if other better positioning technologies (i.e. GNSS) are unavailable.

2.3.3 Motion constraints

Motion constraints are application-specific restrictions on the way the navigation solution can change, based on knowledge of how the user or vehicle is capable of moving. The intention of these is to reduce navigation solution drift by removing impossible motion.

Motion constraints typically split into two different types. The first are actual constraints on the motion, usually for a specific vehicle, for example for a train: no upwards (body frame) motion, and no sideways motion. The second type of motion constraints are known as zero-velocity updates. These detect when the IMU is not moving, which can be useful because when the IMU is not moving, any non-zero motion sensed is an error. This means that in-run calibration can be performed of the accelerometer and gyroscope biases, as well as correcting the navigation solution.

Consider a general-purpose set of navigation equations (or integrated solution), which are designed to give reasonable performance in a wide variety of navigation applications. This kind of algorithm would be included in many commercial off-the-shelf (COTS) products such as GNSS user-equipment chips, and used in a wide variety of products such as smartphones. For these, motion is allowed in all directions without restriction and change of attitude solution (i.e. turning) is allowed at any speed. When the designer has no knowledge of the application this must be allowed, although in practice many filtering techniques place some restriction on the maximum acceleration possible.

However, if the designer knows the application, they can put in place motion constraints on the integrated navigation solution. The designer can use knowledge of how the vehicle or system is expected to move and so reduce the effect of erroneous measurements. This includes restraints on maximum and/or minimum velocity, speed and/or turning rate dependent on the vehicle.

The more detailed the designer's knowledge the better the motion constraints can be given. Consider designing a navigation system for use in road vehicles, as an aftermarket addition, e.g. a typical car 'sat-nav'. In this case one knows the kind of speeds that road-vehicles can typically reach, and one could restrict the navigation system to certain maximum speeds. Cars cannot move quickly in a vertical direction, e.g. up hills steeper than 1 in 3. These restraints would increase positioning accuracy particularly in the up direction, and could detect gross errors (e.g. car moving at 500 km/h).

Gross errors can be detected by considering the minimum and maximum speeds that a vehicle is capable of attaining. An aircraft has both a stall speed and a maximum airspeed. In addition road vehicles cannot travel vertically and most fixed wing aircraft are limited in the speeds that they can attain perpendicular to their direction of thrust.

However, with more specific knowledge of the application better constraints could be used. If an INS is built into a particular vehicle and thus fixed in the vehicle's body frame, much more specific and effective restraints can be used. For example, if it is known that the vehicle cannot move sideways (e.g. train), any sideways motion is an error. The vertical constraint can also be improved as now it is known that the vehicle cannot move directly in the body frame up. If specific performance data is known then the fastest that the vehicle

can turn (at different speeds) can be used as a restraint on the maximum yaw speed.

Another type of motion constraint is a zero-velocity update (ZVU). This has two steps. First, the fact that the system is currently stationary must be detected, using an application-specific detection algorithm. Then the navigation system can use the information that the user is currently stationary. At the least, this can stop the position solution drifting further during the stationary period. It can also be used to calibrate sensor errors, and — in the case of an INS-based system — improve the attitude estimate.

In some applications, this can also be a zero-angular-rate update (ZARU). Returning to the road vehicle example, in order to turn it must be moving forwards or backwards, that is they are not capable of rotating on the spot. This means that when the IMU's accelerometers (and possibly the vehicle's odometer) detect that the car is not moving forward or backward any sensed angular rate on the gyroscope is an error, and observing this error could allow a gyroscope bias to be filtered from the IMU's output. These are even more effective for calibrating gyroscope biases.

ZVUs are a subset of measures of speed (see Section 2.3.2.3). However, unlike many other speed measurements ZVUs are necessarily only available part of the time (i.e. when stationary), which may be infrequent. This means when they do occur they need to be very accurately known to be helpful, and most of their utility (from an INS calibration point of view) comes from the ZARU [3].

An established and very effective technique for improving the performance of an IMU for pedestrian applications is to mount an IMU in a footpod which can be attached to a user's shoe (see e.g. [99, 100]). This technique takes advantage of the fact that when walking, we move one foot at a time so that for about 50% of the time the IMU is approximately stationary. This allows a ZVU to be performed every 1–2 seconds. If implemented correctly ZVUs can compensate for accelerometer biases and even greatly correct for a bias which changes through time. As such, this technique is very effective, and can significantly reduce INS position drift (particularly in the direction of travel) [1, 60, 99, 101].

However, there are limitations on this technique as for many applications a footpod is not desirable. It means that at least two separate pieces of hardware must be present, powered and communicate with each other, as to read a display mounted on a shoe would be undesirable. For most applications it would be impractical to make this connection a wired one, and so this would require separate batteries and a short-range radio link. Therefore this method would be undesirable for military applications due to the RF stealth implications and possibility for signal drop-outs. Also when mounted on the foot, a device of this kind is more exposed to accidental damage. Aside from some specialist users such as firefighters or runners (who often use footpods currently[4], but only for speed and distance not for navigation) a footpod is also undesirable for civilian applications also. A separate piece of hardware which would need to be remembered, transferred from one pair of shoes

[3] Consider that one could drive around a sweeping corner without slowing down, but not rotate a car while stationary.

[4] These commonly work with a sports watch, such as Polar's S1 FootPod (or more recent S3 stride sensor) Garmin's Footpod or the Nike+ sensor (which is not strictly an IMU see [102] and [103]).

to another, wirelessly paired with the device in use and might spoil the aesthetic of and/or limit the choice of footwear, is not ideal.

Despite these problems, in specialist applications, the footpod mounted IMU is a practical proposition. In applications where a map is constructed using simultaneous localisation and mapping (SLAM) techniques (such as [104, 105]), which might then be used by other users, a footpod (or a pair) is a good way for the mapper to increase their accuracy even with relatively low cost MEMS IMUs.

2.3.4 Map and feature matching

Once the navigation system has estimated a position, a geographically referenced database can be used to improve the position estimate [1].

A common example of this is map matching [106, 107]. For example, a typical car GPS sat-nav will move the calculated position onto the nearest road on the digital map. This will generally give a more useful navigation performance for the specific application, but when used out of context map-matching can make the position misleading. Good implementations of map-matching use the history of positions to eliminate impossible road changes (e.g. on/off a flyover mid-span). However, many users have experienced this going wrong, such as passing a motorway junction and being given directions that correspond to approaching the roundabout that one is driving above. Map matching is often done right at the end of the processing chain, e.g. within the sat-nav app on a phone which takes input from the phone's navigation module. However, integrating this into the navigation processing could be more effective, with the map-matching providing feedback to the integration algorithm.

As this technique relies on a map (i.e. database of position-referenced information) there will always be issues relating to the accuracy of the map. In the previous street-map example, the algorithm cannot be as simple as moving the calculated position to the nearest road, as inevitably there are some situations when the user will not be on a road or will be on a road not on the map, and so an off-road detection algorithm is required.

Techniques like map-matching can be more generally referred to as feature-matching techniques. There are both a wide variety of features that could be mapped and several different ways to construct the map (database). Note that while feature-matching can aid a navigation system, the pre-comparison position estimate needs to be reasonably accurate to reduce the area of the map that needs to be searched, otherwise potentially ambiguous matches can result.

There are a large number of items that could be included in a feature database for any particular application, and several considerations in choosing them [108]. First, there must be variation in the feature throughout the area of operation. Second, the measurement should be easy to measure, sometimes they might be already being measured (e.g. magnetic anomalies when using a magnetometer), or be easy to add with minimal extra hardware (e.g. smartphones already have WiFi chips so WiFi signal strength is easy to add). The temporal validity of the features being mapped is a major consideration, that is, how quickly the map goes out of date.

Street maps have already been discussed, but if 3D city models are available then this additional information can be used. For example, in cities the received GNSS signals can be compared with those that the 3D model would expect to receive at nearby points, as each satellite's elevation and azimuth is known. This is effectively creating a feature-map of each satellite in the constellation's availability or non-availability, and is called shadow-matching [109, 110].

Another use of a 3D city model would be to have a camera pointing upwards at an elevation of around 45 degrees (as used in [111–113]) and match the skyline to the 3D map.

Terrain height maps (i.e. maps with contour lines) can also be used for map-matching and are readily available, e.g. UK Ordinance Survey and NATO global databases. If one had a good vertical position, e.g. through barometer integration, in an area of significant height variation (e.g. canyons, mountains), then the height map could improve the horizontal position [114]. This is terrain referenced navigation (TRN) and has been used in aircraft for a long time [1, 115, 116]. In this application it is particularly effective, because hardware for measuring height above the ground, such as downward facing radar, is already available for other reasons (i.e. landing safely), and the terrain height varies more quickly below them as they move fast.

One possible implementation is to consider the height map as a range measurement from the centre of the earth. This can, for example improve the signal geometry of a GNSS positioning, and a comparison between position calculated with and without can aid identification of poor GNSS measurements. Note that, unlike TRN, this does not require there to be significant variation in terrain height. This height aided GNSS approach has been the subject of recent research [117–120], although the approach has been around for some time [1].

For ships there is an equivalent system which can be made using bathymetry sensors. These sensors can be used to measure the sea depth, and comparing it to a sea depth map [121], although this is limited to littoral areas. Additionally, as a ship's height above the geoid can be accurately predicted with tide-tables, the height-aided GNSS techniques can be used effectively.

Neither of these terrain height or sea depth maps are likely to go out of date, and street-maps and 3D city models will only change over months and years, and thus are good candidates for being kept up to date in a crowd-sourced manner (at least the street-maps are, see [122]) and periodically downloaded by the navigation system.

However, other possible features, which may change more rapidly need to be treated differently. For example, in a typical urban area there are a lots of signals transmitted on a range of radio spectrum frequencies. The strength of received signals varies quite a lot over a small distance, as signals are attenuated differently by physical barriers. This means that one potential approach for pattern-matching is to build up a map of received signal strength over an area, on several different commonly occurring frequencies (e.g. GSM, FM or DAB radio, television, WiFi) and then use this map to pattern-match. However, these RF sources ("signals of opportunity") could be moved, switched off, or added without warning and the physical barriers blocking signals could easily move. This means that the signal map must

be updated frequently to retain reliable performance. To do this a simultaneous localisation and mapping (SLAM) approach can be used, e.g. [80], which also uses magnetic anomalies as features. Note the use in [80] of a particle filter (see Section 2.4.4) to track multiple hypotheses. Other filtering techniques can also be used for SLAM [123].

2.4 Integration algorithms

In a multi-sensor navigation system there will be several different sensors providing measurements. Each of these sensors will have different error characteristics and may well measure different quantities.

A naive approach would be to calculate independent (parts of) navigation solutions from different sensors and then to use that from the sensor expected to be more accurate ignoring the other solution. For example, when adding a barometer to a GNSS system, throw away the (noisier) GNSS vertical position and just use the barometer's estimate.

This approach exists in some consumer devices, e.g. sportswatches, where accurate positioning is not critical to the function of the device. For instance, in the sports file exchange format [124], there are separate lines for "wheel speed" (from odometry) and "GPS speed" (i.e. magnitude of GPS velocity), the assumption being that the wheel speed line will be used when available. However, this approach is discarding data that could be used by the system to improve its performance, e.g. to calibrate the odometer in this case.

To use all the information some kind of integration filter is needed to combine the multiple measurements and weight them correctly. This is sometimes called sensor fusion.

This section presents some background on various different filter types, including Kalman filters, more advanced filters related to them and briefly discusses alternatives such as the particle filter and machine-learning approaches.

2.4.1 The Kalman filter

A Kalman filter (KF) [125] is an algorithm to estimate a time-varying series of states from a series of measurements. Kalman filters are frequently used for sensor integration, and their application for INS/GNSS integration is well-established [1, 18, 125, 126]. This enables the IMU's calibration to be frequently updated without any effort by the user, whenever another navigation technology is available. This improves performance both during integrated use and if the INS is used to bridge short outages.

A KF estimates each estimated parameter as a state. For example, 3 states are needed for each of position, velocity and attitude (for 3D navigation). States can also be used to estimate calibration parameters e.g. sensor biases.

When measurements are added to the filter, first the current state estimates and covariances are projected to the current epoch. The measurements predicted using these are compared with the actual measurements and the combination, weighted according to the relative uncertainty of the measurements and the predicted state estimates, is used to update the

state estimates and covariances [1]. The filter is recursive, in that it does not store the old measurements; rather the state estimates aggregate this information in combination with the state error covariance matrix (the P-matrix), which represents the estimation accuracy of the filter's state estimates. More information about Kalman filtering can be found in standard texts such as [127, 128].

To obtain optimal performance, the best possible initial estimates of the states should be used and the state error covariance matrix should be correctly initialised with the initial uncertainty of each state. The filter also needs to know how much the states are expected to change through time, and how accurate the measurements are. Choosing the correct level for all of these settings is known as tuning, and often requires an element of trial and error.

The Kalman filter is based on the following assumptions [127]:

- the statistical distribution of errors are unimodal and can be described using only a mean and covariances;
- the noise terms are white-noise (not correlated with time); and,
- most importantly, that the system propagation and measurements are linear combinations of states.

However, real systems, such as INS and GNSS, do not obey these rules. When attempting to calibrate larger errors, the KF-integration starts to break because linearity and small angle approximations made within its system model are not valid. Linearising approximations typically made include the small angle approximation and the assumption that the products of state estimate errors are negligible. The white noise assumption inherent in Kalman filtering can be partially circumvented by telling the filter that the noise variance is greater than it really is (over-bounding) in order to model noise that is time correlated over a few successive epochs [1], or by use of a shaping filter [19].

If these assumptions are not met or circumvented and/or the tuning is wrong, the Kalman filter will not behave as expected, e.g. estimates will not converge [19]. This situation typically causes the covariance estimates of the KF become inconsistent with the actual covariance of the state-errors.

As Kalman filters assume that the measurements and system propagation are linear combinations of the states; if the system being modelled is not linear then one common approach is to linearise the system. An Extended Kalman filter (EKF) is a KF where the system and/or measurement model are linearised [129]. A filter equivalent to an EKF can be formed by feeding back corrections of some state estimates, like the bias, without significant increase to the computational load [1]. An EKFs work well for in some circumstances, such as linearising GNSS measurement models, but not in others, particularly where there are large angular uncertainties. However, EKFs use the covariance propagation and updates as KFs, which means that the assumption that the products of state estimation errors are negligible still must be valid.

It has been frequently established that INS/GNSS integration is possible, and effective with tactical grade or better sensors [1]. However, with un-calibrated consumer-grade MEMS, the errors are too large for the filter to remain stable. However, it remains an open question

as to when they are too large. Determining the point where the combination of filter errors is small enough for stable operation is one of the research contributions of this thesis as described in Chapter 3.

2.4.2 Advanced Kalman filter variants

There are a number of alternatives and improvements to a standard (or extended) Kalman filter which can outperform a KF or EKF in some navigation applications.

One of these is the Unscented Kalman filter (UKF) [130], also known as a sigma point filter. This parametrises the state covariance in a different way, by using a small number of sigma points. These points can then be put through non-linear system and measurement models and the uncertainty of the transformed distribution reconstructed. It can avoid the majority of non-linear effects of the measurement and system model because it does not linearise them [131]. The transformed distribution matches higher-order Taylor series terms in the UKF than by the EKF, which is often more accurate and so can reduce the instability, as less over-bounding is required. This effectively removes the KFs linear-state-combination assumption, but the noise-is-white assumption still applies.

Another alternative integration filter is adaptive filtering [132–135]. This adjusts the value of the system (Q) and/or measurement (R) noise covariance matrices, based on either varying R and/or Q and examining the sequence of measurement innovations (innovation adaptive filtering, IAE) or by running a bank of multiple models and picking the best one (multiple-model-based adaptive estimation, MMAE). These approaches primarily help in situations where an appropriate starting values of R and/or Q are not known, for instance where the quality of the measurements from a particular sensor are highly dependent on the environmental conditions which are not known. MMAEs have also been applied to multi-modal problems, that is there the first assumption does not apply, although particle filtering is a more common approaches to that problem, see Section 2.4.4. It is possible to combine adaptive and unscented filtering, e.g. [136].

Another approach to improve filter performance is to run an automatic tuning algorithm [137]. This seeks to optimise the filter's tuning for a particular sensor. This is particularly important as the performance of different sensors of the same model can vary quite considerably (see [8, 137] and Chapter 5).

2.4.3 Improved error models

An INS/GNSS producing stable performance depends on, in addition to the 3 points mentioned in Section 2.4.1, the system (INS) being adequately modelled by the system model. In theory an INS/GNSS KF could be modelled with only position and velocity states, as only these are measured by GNSS, but this is not practical for anything other than the best performing INS [1]. Thus the standard method is to model the attitude and the biases, which works with sensors down to tactical grade. This filter will have 15 states: position, velocity, attitude, accelerometer bias and gyroscope bias all for x, y and z axes.

A common mistake, according to [1], is to model only scale factors, rather than both scale factor and cross coupling errors, because both are of similar magnitude for most low-cost MEMS and thus are approximately equally significant for navigation performance. However, as shall be presented in Chapter 5, this is not necessarily the case for post-calibration residual errors. Scale factor-only modelling requires 3 extra states for each of the gyroscope and accelerometer, whereas scale-factor and cross-coupling requires 9 extra states for each. A significant amount of the computational time is spent performing matrix multiplications and inversions both of which are approximately $O(n^3)$, where the n is the number of states[5], so keeping the number of states to the minimum needed is important.

It would be possible to include 2$^{\text{nd}}$-order or higher errors in the INS model, at the cost of further increased computational load. However, in order for an IMU error state to be observable in the INS/GNSS integration it must have a significant impact on the navigation solution [1], that is its effect must be distinguishable from the inertial sensor noise. Which errors have a greater impact on the solution than the noise depends on the system dynamics, i.e. are the errors excited, as well as the magnitude of the sensor noise. The observability of states can be tested by examining state covariance with representative recorded test data [1]. However, due to the large numbers of additional states that would be required to add 2$^{\text{nd}}$-order errors, approaches from Section 2.4.2 such as the UKF [130], should be tested first.

2.4.4 Alternative filters

There are potential methods that can be used for an integration algorithm that are not based on Kalman filters.

A particle filter [140] is a form of sequential Monte Carlo filter [141]. The basic idea is that a cloud of points (particles) represent possible values for the states being estimated, and as new measurements are added, particles which the measurements show to be unlikely are removed and more particles are added near those with high probabilities. Particle filters do not require (or assume) that the states have a Gaussian distribution. So they are well-suited to situations where a Gaussian distribution does not describe the position uncertainty well. A classic example of when a particle filter is better than a KF-derivative filter is for feature matching techniques [80, 142]. In these situations there are typically several discrete positions which match the map well. For example, the user may be either on one street or on the parallel one. However, an error ellipse covering both of these scenarios would show that the user was positioned between the two streets with a large uncertainty, whereas particles could represent the true position distribution better.

The performance of a particle filter can be improved by using more particles, but this increases the computational cost, so performance is a trade-off [143]. This is a particular problem for where a large number of states need to be modelled as the particle cloud is distributed across more dimensions of state space, requiring more particles for reasonable performance. This makes particle filters better suited to 2D navigation (e.g. [142]) than to 3D navigation. A particle filter also needs tuning to improve the performance for the partic-

[5] Matrix inversion can in fact be reduced to $O(n^{2.807})$ [138] and in some cases to $O(n^{2.376})$ [139].

ular application, including, for example, how quickly to remove low-probability particles. In situations where a KF-derivative filter is well suited, e.g. GNSS-only positioning in an open area, the computational cost of a particle filter is generally higher than a KF-derivative or equivalent performance.

It is also possible to use a machine-learning approach for integrating navigation sensors. Artificial Neural Networks (ANN) approaches have been used for GNSS/INS integration [144], learning for example position errors [145]. These are only one class of many possible machine-learning techniques. However, most machine-learning works in a similar way.

They begin with a learning phase during which the algorithm treats each sensor input as a black box and tries to replicate position output by varying the weight given to connections between inputs and outputs. The approach essentially assumes no knowledge prior to the learning phase, although some approaches also combine this with a Bayesian estimator. Then the ANN-derived integration algorithm can be used for normal navigation.

However, generally the performance is only good in situations similar to those included during the learning phase. For example the response on the ANN INS algorithm to a roundabout will not be good unless one was included in the learning phase. Thus, collecting representative training data is critical for their success. Additionally, these generally assume no prior knowledge of sensors' output in relation to the motion, which is unnecessarily general as it known that the accelerometers should strongly respond to acceleration. This can make the ANN algorithm potentially performing poorly in some situations (i.e. unfamiliar ones) and difficult (and computationally expensive) to set up in the first place, as relationships between inputs and outputs which could have been predetermined are determined through a learning process at large computational expense.

There are some hybrid approaches, such as [146], which mitigate some of the computational problems of ANN approaches by using them in combination with Kalman filters.

All of the options, presented in Sections 2.4.2, 2.4.3 and here, come at the cost of increased design complexity and computational load. So the question is: when are they necessary? This is a question that shall be addressed in Chapter 3.

2.5 Improving the performance of an INS

For all the integrated navigation systems considered in this thesis, only those that include an INS are considered. As INS measurements are available at all times and have frequent updates, a common integration algorithm design, e.g. for INS/GNSS [1], is to run inertial navigation equations all the time and when a measurement from another sensor becomes available to run a KF update to correct the inertial solution and estimate INS errors. Thus the INS provides the basis with which other sensors are integrated.

The techniques discussed so far improve performance by including other sensors, which can estimate the position, velocity and/or attitude more accurately when they are available and use this to calibrate the IMU 'in-run'.

However, there are also techniques that can improve the performance of a low-cost inertial navigation system which do not use other sensors. These include: sensor calibration by recording output during a procedure conducted specifically for the purpose of calibration (Section 2.5.1) and techniques using arrays of IMUs to improve the measured signals (Section 2.5.2). These could be considered sensor-level techniques, that is, techniques which use the sensors' output signals to improve the estimates of specific force and angular rate, rather than any other input. The critical difference between the techniques in this section and the in-run calibration techniques discussed earlier is that their effect is earlier in the system processing chain (Figure 2.6), before multi-sensor integration.

2.5.1 Calibration

The out-of-the-box (starting) performance of the consumer-grade MEMS sensors is so low, that it is difficult to estimate their values during use using a standard EKF [19]. Thus as discussed in Section 2.4 a more complex, sensor integration algorithm is required unless the effect of these errors can be reduced. This makes it potentially beneficial to record data specifically for the purpose of improving IMU performance. The recorded output, combined with assumptions based on how it was collected or even measurements from other equipment during its collection, can be used to estimate the IMUs error parameters (see Section 2.2.5) and thus to compensate for their effects.

This section will begin by discussing characterising an IMU's stochastic errors, then discuss systematic error estimation, split into procedures which require expensive laboratory equipment and those that do not require this equipment, or at least only require a subset of it.

This thesis also uses the term user calibration for one carried out by the end-user of the navigation system, contrasting with a factory calibration or laboratory calibration conducted by experts such as the manufacturer of the sensor or of the navigation system. This thesis assumes that the end-user will not have access to any expensive calibration equipment, such as temperature controlled chambers or high-precision rate tables.

2.5.1.1 Stochastic error characterisation

As discussed in Section 2.2.5, the effect of stochastic errors cannot be removed by calibration. However, if the stochastic behaviour is well known then its effect better modelled an integration algorithm potentially improving the algorithms performance.

Stochastic error characterisation generally involves collecting long samples of static data to examine the noise sequence. The standard method for this is Allan variance analysis [5, 147, 148]. However several other newer methods exist based on e.g. wavelet variance [43]. The output of these techniques is a set of stochastic parameters that will define the sensor noise as, e.g. a combination of white noise and two Markov processes.

For low-cost IMUs, the white noise term will typically be the most important stochastic error term, dominant at least over the time period of a few minutes. So for many simple

KF integrations higher order noise terms are not included in the system model. This thesis concentrates on calibration of the systematic errors.

2.5.1.2 Laboratory calibration

IMU systematic error calibration is most accurately conducted using specialist equipment [20]. Example of the various procedures that can be used to perform a laboratory calibration on a MEMS IMU are outlined in [149]. At the simplest level calibration may be static, which can allow the stochastic parameter estimation and gyroscope bias to be determined. As the calibration procedure's complexity, and thus cost, increases the testing can take place on a one- two- or three-axis motion simulator, which might also include a temperature controlled oven in which the IMU will be placed to allow temperature dependent terms to be determined, see also [150]. According to [149], the motions must be measured with 5 to 10 times greater accuracy than that required of the IMUs which are to be calibrated and that five types of test are commonly carried out. These are Hysteresis, Step Rate, Scale Factor Linearity, Bandwidth, and Stability. Hysteresis testing measures the internal friction in the system. The input signal is increased and then decreased and the difference between the measurements on the 'way up' and 'way down' is the hysteresis. The step rate test, subjects the IMU to inputs signals at given step values, the time to stabilise on the new value (settling time) and the angular rate accuracy at each step are measured. Scale Factor Linearity testing, measures the sensor's response to input through its whole measurement range, the output being the deviation from a linear response. Bandwidth testing measures the sensor's response to input signals at various frequencies. Stability testing is the name given in [149] to analysis of the residual noise, i.e. stochastic error characterisation. [149] also mentions that all of these tests may be carried out at a range of different ambient temperatures. As noted by many authors [1, 4, 20, 55] the temperature can have a significant effect on the systematic errors of automotive-grade MEMS so this step is important.

Several production INS or GNSS/INS systems use consumer-grade MEMS IMUs, which have near tactical grade systematic error performance, because the sensors have been factory calibrated [58, 151]. However, they still have considerably worse noise performance than a true tactical grade MEMS IMU, such as [152]. In some cases different prices are given for differing qualities of calibration, e.g. VectorNav [26] offer products with calibrations valid at 25°C which can be upgraded to being valid from -40 to 85°C for an extra $300.

2.5.1.3 User-conducted calibration

Factory calibration adds significant cost as mentioned above so there is incentive to develop simpler calibration procedures that can improve the calibration of the sensors, without requiring specialist equipment, and thus could be conducted by the end-user of the device to reduce cost. There are a variety of techniques presented in the literature for user-calibration, and their authors also significantly differ on what equipment an end-user might be expected to have access to. Some authors require a turn-table [153], some assume that it is not high-precision [154]. Others require specially made equipment, such as a specially shaped box [155] or a hinged surface [156] which is less expensive than full laboratory calibration

equipment . The calibration techniques that will be presented in this thesis do not require special equipment.

The accelerometers' biases, scale-factors and cross coupling can be calibrated from static tests, because a vertical specific force of approximately 9.8ms^{-2} will apply to the static sensors, as can gyroscope biases (and g-dependent error). However, gyroscope scale factor and cross coupling cannot be observed without rotation, so a dynamic calibration is necessary.

The static calibration is often performed by means of a six-position test. That is, by mounting the sensors in a cuboid box and recording a sample of static data on each of the faces of this box, allows estimates to be made of accelerometer bias, scale factor and cross coupling, as well as gyroscope bias and g-dependent error. Many authors have published papers including a version of this procedure [8, 153, 154, 157–159]. According to [149], in combination with a granite table and "a precision fixture" (a cuboid), this represents the lowest cost version of factory IMU testing. There are different ways in which the biases and alignments can be estimated from this data, which is not always specified by the authors, two alternatives will be described in Section 4.5.1, although others approaches are possible, e.g. [157].

It should be noted that for some high-performance IMUs the Earth's rotation rate can be used to calibrate the gyroscopes as part of static six-position tests, see [160]. This thesis' focus is on low-cost sensors which are not accurate enough to use the Earth's rotation rate ($\approx 0.0041\,^\circ/\text{s}$) for calibration.

An interesting variation of this 6-position test is presented in [155] where instead of a cuboid box an icosahedron (20 faces) is used for accelerometer calibration. This theoretically allows some accelerometer non-linearity terms to be estimated, as specific forces between 0 ms^{-2} and 9.8ms^{-2} are included. Although this is not done in [155], the extra information is used instead to relax the requirements for construction accuracy. However, a disadvantage of this is that while a cube-based procedure can use the navigation system's case, this requires a separate special calibration box, which reduces its utility as a user-conducted procedure. One strength of their method is that increasing the number of faces makes the procedure less sensitive to one or two poorly aligned faces. They reduce the cost of constructing this icosahedron by 3D printing it. This is not ideal as most 3D printers, particularly the extrusion-based ones that it is clear they have used from the photographs in the paper, have separate x- y- and z-axis motors and position-control. This means that their manufacturing errors in the resulting 3D printed icosahedron will be correlated to the 3 axes of the printer. This correlation removes this potential benefit of their method.

Some authors, including [161], adapt this 6-position test to 12 or 18 position test by resting the cube on (some or all of) its edges at "approximately 45°" from the level table, although the reason for doing this not made clear in their paper beyond stating that it is "optimal". It appears that they are under the mistaken impression that calibrating the 12 coefficients of the $\mathbf{A_a}$ and $\boldsymbol{b_a}$ requires data to be recorded at 12 positions, when, due to the fact that 3 sensor outputs are recorded at each position; the standard 6-position test is already overdetermined. In [154] a similar statement is made. The authors note that there are 12 coefficients which are being calibrated (the entries of $\mathbf{A_a}$ and $\boldsymbol{b_a}$ in this thesis' nomenclature).

They then state:
"Then, through collect n ($n \geq 12$) sets of different positions measurement data (u_1, u_2, ... , u_n), use the non-linear least squares method to determine the above twelve unknown parameters." (direct quotation from [154])

Generally these methods, aside from the above many-position techniques, assume that the surface upon which the sensor box is placed is level, and also implicitly assume that the 'box' is exactly the shape that it is meant to be, that is, entirely without manufacturing errors.

When it comes to dynamic calibration, there is little agreement on what equipment it is reasonable to expect the end-user to have. Some authors describe procedures which use rate-tables as user-calibration [153], assuming the user has access to a low-precision rate table, perhaps intended to be a vinyl record player, where the angular rate is not known but is assumed to be fixed [154, 162]. Some authors use a specially constructed piece of hardware for their gyroscope calibration, e.g. the "rotation hinge" of [156]. None of these would qualify under this thesis' definition as user-calibration, except perhaps the last one.

There are some studies that suggest methods to calibrate the gyroscopes without using very precisely known (or measured) rotations. [154] presents a method for calibrating an IMU without a rate-table, this involves calibrating the accelerometers using a level surface, then rotating the IMU on an angled surface and using the accelerometers to provide an attitude reference by tilt-sensing (note that this only provides the roll and pitch parts of the attitude solution).

However, their actual experimental data is collected on a high-precision turntable and they process it twice, once through their algorithm and once using the output of the high-precision turntable. If they had used hand rotations to test their algorithm variations of speed, slight variations in the rotation axis and not-precisely stopping at 360° would have been tested. This means that their procedure is flattered by comparison with how it would be conducted in-the-field, so their test does not represent a realistic use case.

Other authors use machine learning to calibrate gyroscopes [161]. They conduct a static calibration of the accelerometers, and use standard magnetometer calibration techniques (see [76]). They then perform a large number of rotations, about many different rotation axes, by hand with static periods in between, recording the gyroscope and magnetometer output. This data is used to calibrate the gyroscope by using machine-learning to pick the gyroscope $\mathbf{A_g}$ parameters that maximise the agreement of the magnetic heading with the gyroscope-derived heading. This is not a very tractable problem for an analytic solution, so one can see why machine-learning techniques are employed. It is not clear whether this approach would produce superior results to a normal KF (or KF-derivative) sensor integration which could be used for the same task, given the frequent static periods effective ZARUs could be performed. There is potential for machine-learning algorithms (in general) to produce solutions that fit the training data, but do not work in practice [1]. A similar approach is presented by [156], although instead of magnetometer output, they use an accelerometer-derived tilt-angle, and their procedure seems to depend even more strongly

on detecting ZVUs. It does not produce very repeatable estimates, particularly of the off-diagonal elements of $\mathbf{A_a}$ and diagonal elements $\mathbf{A_g}$.

In [159] a method for gyroscope calibration using manual rotation of (in their case) an IMU inside a "right-angle iron" which has "six surfaces that are square and parallel to each other within $0.002''$ per $6''$ " [159], in other words a highly precise[6] calibration fixture. In this calibration process, biases are estimated by a six-position calibration similar to those described above, then the IMU is "rotated by hand on a surface contact with a clean flat bed" [159] three times with a different (mutually orthogonal) face in contact with the bed. After this there is a single 90° rotation using a straight edge (a ruler) attached to the table. The measurements during the 3 rotations are used to estimate the gyroscope scale factor and cross coupling up to a scale ambiguity. This ambiguity is due to not knowing the true values of the speed or total angle of the rotations. The final 90° rotation is used to solve this ambiguity. Note that while the actual manoeuvres conducted are similar to those that will be presented in Chapter 4, there are important differences and the calculation method is not the same.

It is not clear how much the technique presented in [159] depends on the high degree of angular accuracy of their calibration fixture, however as all the scale factors depend on the final 90° rotation, it is likely that the accuracy of this rotation has a large impact on the accuracy of the error estimation. If the estimates' dependence on the accuracy of the calibration fixture was low, then this would be a workable calibration procedure. They present the results of calibrating a real set of gyroscopes, but no information on the accuracy of this estimate other than their (least-squares) parameter-fitting residuals. They also present a simulation, but this does not apparently contain errors other than IMU sensor noise, which judging by the order of magnitude smaller parameter-fitting residuals, may not adequately capture all the estimate errors.

A recurring problem for these kinds of calibration procedures is how to determine the accuracy of the calibration. This is a harder question to answer than one might expect at first. The issue is that the true values of the calibration parameters for real sensors are not known. Ideally, the calibration procedure's estimates for the calibration parameters would be compared with a known-good calibration procedure, such as that conducted by a high-performance factory calibration. However, the (usually) academic groups that publish these kind of procedures do not often have access to the equipment needed for a factory calibration. The study [154] comes close, but they did not actually present the hand-rotated results they claim in the abstract.

A more common approach is to simply repeat the procedure a number of times and present information on the repeatability of its estimates. For example, [155] simply repeats their calibration procedure ten times on the same array of 14 sensors. This demonstrates that their procedure is insensitive to human error[7] and to noise[8]. However, if there were a systematic mis-shaping of their calibration icosahedron, this would create an error, but it

[6] Angular error of about 5.8×10^{-7} °, by trigonometry

[7] Which is not surprising given that it is based entirely on data at static positions.

[8] The expected error from noise on the mean value of a static dataset is readily calculable, see Equation 4.99.

would manifest identically over the repeats, and thus would not affect the repeatability. This is just one example why repeatability is a necessary but not sufficient condition for accurate calibration. That is not to say that it is an easy condition to meet. In [156] a gyroscope calibration procedure is demonstrated by repeating it 4 times, as is shown in their Table 3, this their estimates of $\mathbf{A_g}$ matrix components have a standard deviation of up to 0.008, i.e. just under 1%. Five years ago, when typical gyroscope scale factor errors were ±6% [53] or worse, this might have been a useful user-conducted calibration. However, newer sensors come from the factory with scale factors are ±1% at 25°C [45] or ±2% across the entire operating temperature range [46], so this procedure barely represents an improvement.

One possible option is to do a semi-simulated test. This starts by recording a real calibration procedure with a one of the near-tactical performance factory-calibrated IMUs like Xsens, or a higher performance IMU if available, which hopefully measure the calibration procedure with very little error. Then artificial errors (e.g. biases) can be added to the recorded inertial output, and one can see whether these artificial parameters can be re-constructed. This kind of approach is often done with filter-estimated parameters, e.g. in [56], but there is no reason why it cannot be extended to deterministic or machine-learning procedures. Of course, the initial sensor calibration has some small errors, so they may account for some of the error in re-constructing the artificial errors.

The other option is a full simulation, as is presented in Chapter 4 of this thesis. Of course, the challenge then is validating it.

The final option is to go fully empirical. A navigation solution can be calculated from a period of recorded data, with and without calibration, or with differently calculated calibration parameters can be shown. Examples of this approach will be presented in Sections 5.6 and 6.6.3. However, this only can give an idea of the overall calibration performance, rather than the accuracy of any individual error sources.

2.5.2 Redundant IMU configurations

One way in which the performance of the inertial navigation system can be improved is by using more than the minimum of 3 gyroscopes and accelerometers. Different authors use different terms for this, e.g. redundant IMU [163] [9] or multi-IMU [164] [10] but in this thesis the term IMU array or array of IMUs will be used, from here onwards. This terminology is also used in existing literature, such as [166].

10 years ago, a low-cost IMU would have been constructed from several separate chips, possibly on circuit boards fixed in a frame at 90° to one-another, e.g. [157]. As single-chip MEMS IMUs are available now, the additional expense and effort to construct one from several chips is unlikely to be worthwhile. At the most there will be a 3-axis gyroscope chip

[9] This unfortunately has the same acronym as reduced IMU which is rather the opposite thing.

[10] The choice of the term MIMU to refer to their array is poorly chosen as this is used by other authors, e.g. [154], to refer to a single MEMS IMU, as Micro Inertial Measurement Unit (MIMU), and by Honeywell to refer to one of their high performance inertial navigation units "miniature inertial measurement unit" (MIMU) used on some space missions [165], although this not miniature by MEMS standards.

and a 3-axis accelerometer chip. This means that for any given array, it will be constructed of sets of orthogonal triads rather than by positioning individual sensors.

It is generally accepted that a larger array (i.e. more IMUs) will perform better, e.g. [164, 166], with the trade off of additional size, cost and power consumption. Most authors mention the improvement in noise performance that comes about from averaging the IMU output

The optimal configuration for an array is an open question, and generally depends on the application and characteristics of the sensors. Aside from the number of sensors in the array, there are a number of other parameters that can be varied. The array can have various physical arrangements: the sensors might be arranged with their sensitive axes pointing in various directions, the separation between sensors might be minimised or made greater. Additionally, the sensors within the array might be of different models or their configuration may be varied to measure the specific force or angular rate signal with different errors. This thesis uses the term aligned array to refer to an array where all the IMUs share the same orientation, i.e. where they are all arranged side-by-side with each of the sensitive axes of all the sensors pointing in the same direction.

The physical separation of sensors within an IMU is sometimes referred to as the lever-arm. In most sensor designs, it is not possible for the sensitive axes of all the sensors in the triad to be co-located. So, if an IMU's measurements are considered as the motion of a particular point in space, an angular acceleration about this reference point can be erroneously recorded as a small linear acceleration by the IMU. If the displacement is known this can be mathematically compensated for [1,4], if one assumes that the lever-arm is fixed, i.e. that the mounting is arbitrarily stiff. On a MEMS scale device, the distances involved are likely considerably less than a millimetre, so it is possible that some (at least low-cost) MEMS IMUs ignore the lever arm between axes. However, once an array of MEMS sensors is being considered the distances become less trivial. Some array designs seek to place the sensors as close together as possible which minimises this effect (e.g. [164]).

However, it is possible to use this lever-arm effect to measure attitude change and estimate attitude using dead-reckoning [167]. Some authors suggest all accelerometer IMUs using this effect [167–170], where the attitude change is tracked by deliberately separated accelerometers. An important point to note is that greater IMU separation increases the sensitivity to this effect, but the lever-arm must be sufficiently stiff that its deformation can be assumed negligible. This would make placing IMUs on the wing-tips of an aeroplane less effective than on opposite ends of the fuselage, for example. This angular measurement from the accelerometers could be combined with the attitude signal from a gyroscope, if desired. For example, it can be used to reconstruct the angular rate signal when the gyroscope has clipped [171].

Some authors arrange their IMU array so that their IMUs are skewed from each other, e.g. [163]. The idea of this is that none of the sensors share a sensitive axis. For example, in [163] 4 Xsens factory-calibrated IMUs are arranged with one on each face of a tetrahedron. [163] also discusses other possible orientations, including other platonic solids and around the sides of a cone, whose half-angle could be varied to change the vertical sensitivity, this

arrangement might be useful for vehicles which have different performance requirements in the vertical and horizontal, but only applies to positioning individual sensors rather than triads.

This skewed arrangement makes sense when one considers running a fault detection and exclusion algorithm over the whole array. Firstly, there are many more sensors that pick up (part of) a signal in any particular direction for consistency checking, as compared to an aligned array with the same number of sensors. Secondly, when some of the sensors are excluded from the solution, it is not possible that two sensors sensitive in the same direction would be removed, so the ability to measure inertial signals in all directions will be maintained. Additionally, if there are significant performance differences between the different sensors in the triad, aligned arrays retain these differences, but other arrangements need not, see Section 6.3

The main drawbacks of 3D skewed arrangements is that they are much more complex to construct, require more space and cannot in general be built onto a single PCB. This complex construction makes it more likely to be calibration issues as the spatial relationship between the sensors depends on the mounting which may well change shape with temperature and/or be more sensitive to accidental damage, than if the sensors were all fixed to a single PCB. The advantages of skewed arrays are highlighted by the fact that [163] uses the term "skew-redundant IMU", that is a skewed arrangement is good for redundancy but if all the sensors are going to be used with equal weighting, there is little benefit to offset the complexity of a three dimensional arrangement.

Another option for arranging the IMUs is what this thesis will term the anti-parallel arrangement. In this arrangement the positive x-direction of one sensor is aligned with the negative x-direction of another, similarly for y- and z-axes. It is not mathematically possible to have two IMUs arranged to form a fully opposed direction array (in x, y and z). However, it is possible to arrange 4 sensors so that there are two sensors in each direction. It is also possible to arrange that the positive x is opposed by the negative y, which allows two IMUs on a single PCB to form an anti-parallel arrangement. The output of the sensors can then be sent through a very simple rotation matrix to get them at the same orientation before averaging. This arrangement of sensitive axes was suggested in [55] as a way to reduce the effect of temperature-related bias drift. The observation was made that a particular gyroscope sensor's bias appeared to increase with increasing temperature for several samples of the same model. The idea of the orientation is that without actually calibrating the temperature-related drift the majority of its effect would be removed by the fact that it caused a spurious positive rotation on one sensor, but a negative spurious rotation on the other. This idea was proposed but not tested in [55] and it has been further tested and slightly extended in this thesis, see Section 6.2 and in [8].

The sensors used in the an array need not necessarily all be the same. It would be possible to have an array of sensors with different error characteristics. Sensors might have different maximum ranges, differing performance characteristics, in terms of noise, temperature stability etc. In [56], a navigation system based on two gyroscope triads and two accelerometer triads with "dissimilar" error characteristics is constructed. They then use the four possible gyroscope-accelerometer combinations from this for fault-detection. This is demonstrated

by artificially adding faults to the data.

In very high performance applications, arrays have been demonstrated to allow a very high measurement range, while keeping good precision for smaller forces. In [172] arrays are constructed with a combination of accelerometers with 10g, 100g and 1000g ranges, with a weighting scheme. This array is constructed on a single piece of silicon, i.e. it is an internal design of a MEMS sensor within the first-level-package (FLP), so it is not an option for low-cost applications. However, it is worth noting that sensor arrays are also being used for high-performance sensors.

2.6 Topics requiring further investigation

This chapter presented the main issue for low-cost inertial navigation; that of position drift as errors accumulate with time, made worse by the inaccurate initial calibration.

One way to improve navigation performance is to integrate another navigation technology (Section 2.3), which can be used to remove accumulated navigation solution drift and to simultaneously calibrate the IMU's errors. To do this requires an integration algorithm. In Section 2.4, these integration algorithms were introduced and their inability to reliably integrate low-cost INS due to the size of the sensors' errors was described. In order to know when a more computationally expensive alternative (e.g. [130]) needs to be used the level of error for which a standard KF remains stable needs to be established. This will be the focus of Chapter 3 of this thesis.

Once the target specification has been established, the techniques in Section 2.5 can be used to improve the calibration of the sensors.

Pre-calibration where output recorded during a specific set of manoeuvres is used to estimate the INS's errors is one such option (Section 2.5.1), but laboratory conducted procedures vastly increase the navigation system's cost. Ideally, the end-user would conducted their own calibration reducing the unit cost considerably. Existing approaches to user-conducted calibration differ on what equipment an end-user might have access to, they also very rarely make a thorough assessment of the calibration procedure's accuracy, in particular the effect of different errors on the overall performance. To this end in Chapter 4 a calibration procedure based on the 6-position test, but with a number of improvements that increase its accuracy, and also a gyroscope calibration procedure, which uses only hand-rotations, are presented. The sensitivity of these procedures to a wide range of procedural and IMU specification inaccuracies is assessed through simulation, unlike most of the procedures presented in the literature, and the procedure is tested with physical experiments (Chapter 5).

The array techniques reviewed in Section 2.5.2 are assessed further and some are tested experimentally in Chapter 6. The underlying assumptions for these techniques are examined and their validity assessed for one model of MEMS IMU.

Chapter 3

Kalman Filter Stability

This chapter assesses use of the Kalman filter (KF) to integrate the navigation solution of inertial navigation system and another positioning technology, such as GNSS, and by doing so estimate, and hence calibrate, the systematic errors of the inertial sensors in-run. It identifies the assumptions that need to be made and when these assumptions break down. These assumptions are not typically valid when the inertial sensors begin with a very poor calibration so for sensors whose initial calibration is outside a certain limit, such as consumer-grade MEMS, the KF frequently becomes unstable.

In this chapter, the extent of these limits are established. First, some relevant parts of the background are briefly recapped, (Section 3.1). Then, well-defined filter failure criteria are established (Section 3.2). These are crucial for determining when the current filter is inadequate and a more complex filter is needed.

Having established suitable failure and success criteria, Monte-Carlo simulations are performed with a range of different sensor specifications in order to determine the maximum tolerable sensor errors. The approach to computing these simulations is presented in Section 3.3 and the results of these simulations with a car and quadcopter motion profiles are shown in Section 3.4. The implications of this chapter's results for the rest of the thesis are discussed in Section 3.5.

A preliminary version of this study was presented at the 2014 Institute of Navigation (ION) GNSS+ conference [7]. Note that the results presented here show a considerable improvement from the preliminary results presented in [7] because of the improved filter tuning, described in Section 3.3.4. The research presented in this chapter published in ION's journal NAVIGATION [6].

3.1 Background

In Chapter 2 background was presented on inertial navigation systems and their errors (Section 2.2). The most salient points for this chapter are that INS errors are split into stochastic (noise-like) and systematic errors. Only the latter of these can be calibrated, and for low-cost MEMS IMUs they contribute a significant part of the error in the navigation solution. Thus, calibrating the systematic error sources represents low-hanging fruit for improving low-cost INS performance.

This calibration could be performed in several ways. As was presented in Section 2.5.1, an accurate result it could be achieved by a laboratory calibration procedure, but it would be too expensive for the low-cost applications considered in this thesis. This leaves conducting a simpler instruction-based calibration procedure that might be accomplished by the end user, as was discussed in Section 2.5.1.3 and will be investigated further in Chapter 4, and calibration in-run. That is where the inertial sensors are calibrated during the normal use of the navigation system, when a second positioning technology is available.

Many different navigation technologies could be used to aid/calibrate an INS. Maximum robustness is achieved by combining many different sensors [108], but complex multisensor navigation brings many challenges [9]. In this chapter, the INS is assumed to be aided/calibrated using global satellite navigation systems (GNSS). However, the technique presented here could be used with other aiding sources (see Section 2.3.2). GNSS is commonly used due to the low cost of the user equipment, free to use infrastructure and fairly high accuracy and availability. INS/GNSS integration is a well-established technique [1, 126].

3.2 Kalman filter failure

As discussed in Section 2.4.1, the Kalman filter (KF) [125] is an estimation algorithm that is linear; if the system is not it must be linearised. In addition, correct filter tuning is very important for the filter to operate in the manner intended by the designer.

The Kalman filter is based on the following assumptions [127]:

- the noise terms are white-noise (not correlated with time); and,
- most importantly, that the system propagation and measurements are linear combinations of states.

However, real systems, such as INS/GNSS, do not obey these rules. Approximations to allow the KF to be used for this application include the small angle approximation and the assumption that the products of state estimate errors are negligible. The white noise assumption inherent in Kalman filtering can be partially circumvented by telling the filter that the noise variance is greater than it really is (over-bounding) in order to model noise that is time correlated over a few successive epochs [1].

If the these assumptions are not met or circumvented and/or the tuning is wrong, the Kalman filter will not behave as expected, e.g. estimates will not converge. In short, the Kalman filter will break if the errors are too large.

To examine the limits of Kalman filter performance, criteria must be established for when a Kalman filter is performing unacceptably, inadequately or unstably, henceforth failing.

In order to decide when a particular Kalman filter fails, it is necessary to first examine how it should behave; so that failure can be detected when it has occurred.

A well-behaved KF should start with state uncertainties (covariance matrix entries) which slightly exceed the true uncertainties of the state-error distribution. This is done to aid filter stability when the true stochastic behaviour of the states diverges from the KF's

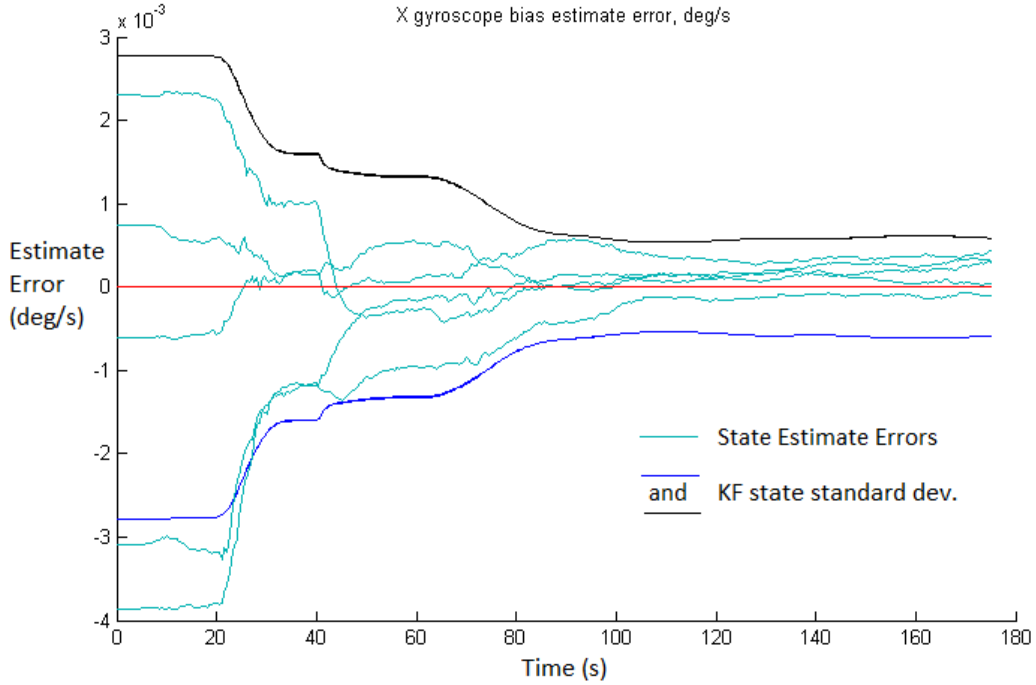


Figure 3.1: 5 runs of a Monte Carlo simulation illustrating typical state estimate behaviour when an INS/GNSS KF integration is working as intended. In this example the estimation errors of this state (X gyroscope bias) converge towards zero and their distribution is well described by the KF's state standard deviation.

assumptions. Then, as more aiding (e.g. GNSS) measurements are added, both the filter's state uncertainty and the real standard deviation (SD) of the error in the state estimates should reduce, as is illustrated in Figure 3.1, particularly when manoeuvres (such as turns) take place. Note that the accuracy of the estimates eventually plateaus at a level that depends on the accuracy of the aiding measurements, the IMU noise and the time variation of the systematic errors.

When the sensor errors are very large, the linearising assumptions, such as the small angle approximations, break very quickly. For example, GNSS integration can normally only correct attitude errors indirectly based on their effect on the position/velocity solution. The KF uses the off-diagonal elements of the P-matrix to infer the attitude and instrument errors from the position and velocity errors. When the gyroscope biases are large, the attitude error can grow more quickly than it can be corrected, breaching the small-angle approximation. When the small angle approximation ceases to be valid, the P-matrix no longer models the correlations between the states correctly, so the corrections to the state estimates from the measurements can be applied incorrectly.

As the errors become larger, other KF assumptions, such as the product of two state-errors being negligible even when the states are correlated, can also break down. This produces poor position, velocity and attitude estimates. It also produces similarly erratic estimates for the IMU error states, as shown in Figure 3.2. Clearly, if the results of a simulation are as erratic as in that example, detecting that this is a failure is straightforward.

However, when the IMU errors are smaller, the position, attitude and velocity errors can

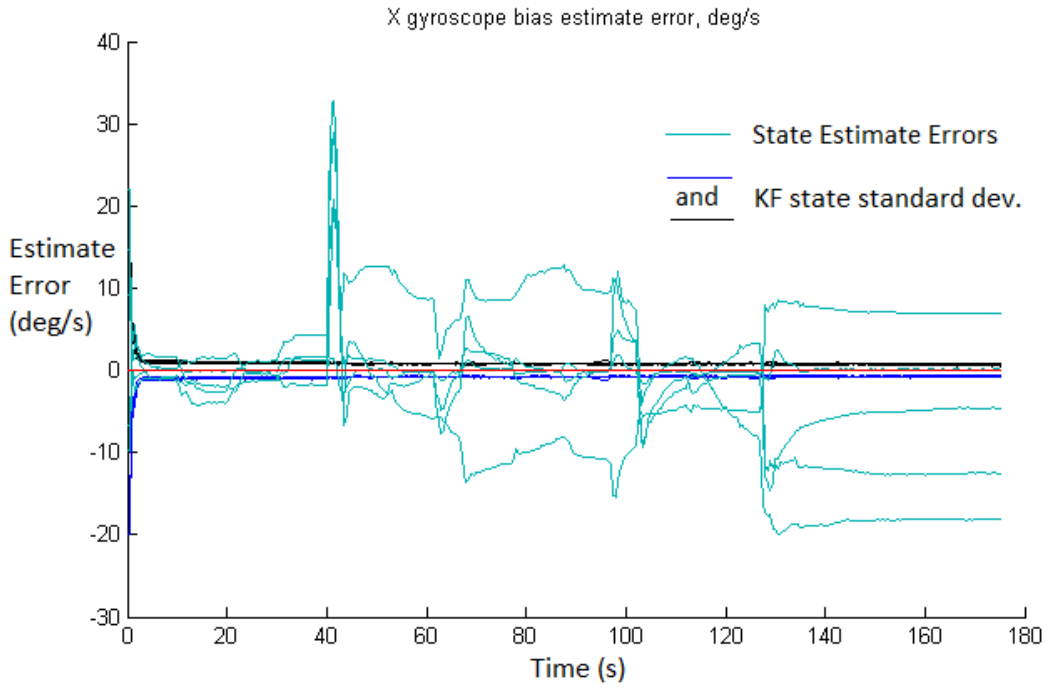


Figure 3.2: 5 runs of a Monte Carlo simulation illustrating typical state estimate behaviour when a KF integration is not working as intended. In this example the estimation errors of this state (X gyroscope bias) do not converge to zero and vary erratically, additionally their distribution is much greater than the KF's state standard deviation. Note that one would expect the SD bounds to be exceeded 35% of the time, but this illustrates 4 out of 5 being exceeding the bounds by greater than a factor of 3 which would only be expected 1% of the time.

remain apparently small, within a few degrees and tens of centimetres (per second), even though the IMU error estimates produced are inconsistent with the corresponding state uncertainties (Figure 3.3). This situation illustrates a problem when determining if a simulation has failed. For a given error distribution, sometimes the IMU errors are within the uncertainty bounds and sometimes they are not, depending on the individual IMU samples. The uncertainty is needed to represent the whole distribution not just individual samples. Thus while a particular sensor may happen to have a fortuitously small starting error, with the result that the GNSS integration works well, that is not sufficient. So, it is necessary to run a large number of different values of the starting errors, sampled from the same distribution, as part of a Monte Carlo simulation, and thus demonstrate that the uncertainty represents the whole distribution.

The intended use of an INS/GNSS integration KF is important when determining failure criteria. The aim considered here is to calibrate the inertial sensors in the IMU so that the INS can bridge a future GNSS signal outage. This means that it is required that the errors in KF state estimates corresponding to the IMU's errors, e.g. the accelerometer and gyroscope biases, finish with a smaller distribution than they started. The test could be: the simulation must end up with a smaller average magnitude of errors than it started with. However, in the real context some short time later the filter could became unstable and the error estimates go wildly wrong.

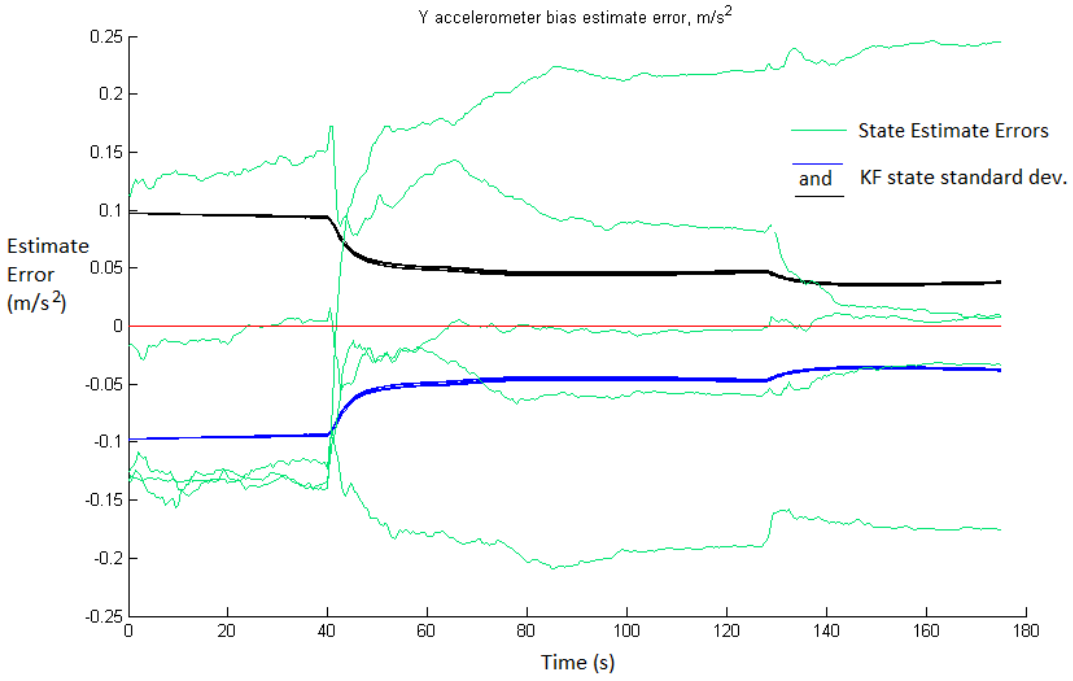


Figure 3.3: 5 runs of a Monte Carlo simulation illustrating a typical state estimate when an INS/GNSS KF integration is behaving inconsistently. In this example some of the estimation errors of this state (Y accelerometer bias) converge to zero but others do not, this spread of estimate errors is not well described by the KF's state standard deviation.

Another possible approach could be empirical. A simulated signal outage could be added and how well it is bridged by the calibrated solution could be examined. However, it is only possible simulate a trajectory a few minutes long before the computational cost of running the Monte Carlo simulation becomes very large. In some applications, there may be tens of minutes or even hours between outages, so choosing an arbitrary amount of time after which to test whether the resulting calibration is adequate is unsatisfactory. For instance, in this chapter only three minutes of motion are simulated, and the estimates of the states might continue to improve if the simulations was longer. Additionally, determining what is adequate performance for bridging a GNSS outage is not straightforward. For example, two INSs with different amounts of sensor noise would perform differently even if both had their systematic errors perfectly calibrated. Perhaps the GNSS-calibrated INS could be compared to an INS with identical noise parameters but zero systematic errors, although it is not clear how much worse one should expect an imperfectly calibrated INS to behave than a perfectly calibrated one, and still be judged adequate. Therefore, a success criteria based on this approach would be specific to the requirements of a particular application, and thus unsuited to drawing the more general conclusions intended for this chapter.

In this chapter, a different approach is taken to deciding whether an INS/GNSS integration Kalman filter is successful, which depends on the consistency of the filter's state-covariance matrix.

In general, a Kalman filter will behave consistently if all of the assumptions stated in Section 2.4.1 are valid and if the variances and covariances stored in its state error covariance matrix

(P-matrix) are an accurate reflection of the real errors of the estimates. Fortunately, if the former condition breaks, that latter tends to also. In a physical system, the true errors in these state estimates are not known. However, in simulations, the truth from which the estimates differ is known, which makes it possible to use criteria based on the consistency of real estimation-errors with the estimated covariance.

A necessary but not sufficient condition for the state error covariance matrix being consistent is that the filter is tuned correctly, as otherwise it is either wrong from the start or will quickly become so.

The stability condition that is tested here is the relationship between that filter's state uncertainty for every state at each epoch (averaged across the 100 runs in the Monte Carlo simulation) and the root-mean-squared-error (RMSE) across the set of 100 simulations of the filter's state estimate error. Attitude state errors are computed by comparing the true and estimated rotation matrix to find the error, after which this attitude error rotation matrix is converted to Euler angles. RMSE is only one of several possible techniques for comparing the true and estimated state uncertainty. The threshold for failure is that the worst state's RMSE does not exceed 2σ for more than 5% of the simulation time, where σ^2 is the corresponding state variance from the state error covariance matrix. It should be noted that this threshold is very generous. If the state uncertainty, σ, is correct then the RMSE over the 100 Monte Carlo runs should be very close in value to it, as the unbiased estimator of the distribution variance, σ^2, is $\sigma^2 = \frac{n}{n-1}s^2$ where s^2 is the sample variance and n is the number of samples. The probability that this condition would fail by chance is very small, because for a normal Gaussian distribution $P(s > 2\sigma | n = 100) \approx 3.8 \times 10^{-37}$. Thus, if this condition fails, it is reasonable to infer that the state variance is not accurately describing the real errors.

3.3 Simulation Approach

In this section the approach taken to run simulations to determine whether a particular IMU model is suitable for in-run calibration is described. First, the particular Kalman filter variant used is discussed (Section 3.3.1). This is followed by details of the simulation algorithm's design (Section 3.3.2), the filter tuning (Section 3.3.4), the motion scenario used (Section 3.3.3) and the GNSS parameters (Section 3.3.5). Finally, the way in which the set of inputs were searched is outlined in Section 3.3.6.

3.3.1 Basic Kalman Filter

For the simulations discussed here the intention is to use the most basic Kalman filter that might realistically be possible. The idea being to assess the limits of this basic filter and then determine when more complex filters or pre-calibration are required. This basic KF is both the most simple to program and has the lowest processing load.

The INS/GNSS integration Kalman filter is loosely-coupled, which means that the GNSS information is given to the filter in the form of GNSS position and velocity measurements,

rather than as, for example, GNSS-pseudoranges. This has the advantage that the results are applicable to other sources of aiding. It is also a standard Kalman filter rather than an extended Kalman filter (EKF) [127, 173] or unscented Kalman filter (UKF) [130]. This means that it has linear system and measurement models. However, it has closed-loop correction of the inertial sensor error states, which improves the stability of the filter if the magnitude of these states were to become large. This is a linearisation approximation and is equivalent to an EKF system propagation [1].

The states modelled by this basic Kalman filter are the minimum commonly used configuration. That is position, attitude and velocity (3×3), accelerometer bias (3) and gyroscope bias (3), a total of 15 states. Potentially better performance could be achieved by estimating the first-order IMU errors as well. However, an additional 27 states would be required to estimation all components of the accelerometer and gyroscope scale factor and cross-coupling errors and the gyroscope g-dependent errors. Modelling additional states makes the filter significantly slower as the majority of the simulation's time is spent performing matrix multiplications, which require computational power proportional to n^3, where the n is the number of states.This is particularly significant when dealing with the limited computational resources available in embedded hardware.

The KF is adapted from the example software provided open-source on the CD accompanying [1]. The transition ($\mathbf{\Phi}$), system noise ($\mathbf{Q}$) and measurement ($\mathbf{H}$) matrices used can be found in equations 14.50, 14.82 and 14.115 of [1], respectively. These are presented below.

$$\mathbf{\Phi}^{\mathbf{e}}_{\mathbf{INS}} = \begin{bmatrix} \mathbf{I_3} - \mathbf{\Omega}^{\mathbf{e}}_{\mathbf{ie}} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & \widehat{\mathbf{C}^{\mathbf{e}}_{\mathbf{b}}}\boldsymbol{\tau_s} \\ \mathbf{F}^{\mathbf{e}}_{\mathbf{21}} & \mathbf{I_3} - 2\mathbf{\Omega}^{\mathbf{e}}_{\mathbf{ie}} & \mathbf{F}^{\mathbf{e}}_{\mathbf{23}} & \widehat{\mathbf{C}^{\mathbf{e}}_{\mathbf{b}}}\boldsymbol{\tau_s} & \mathbf{0_3} \\ \mathbf{0_3} & \mathbf{I_3}\boldsymbol{\tau_s} & \mathbf{I_3} & \mathbf{0_3} & \mathbf{0_3} \\ \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{I_3} & \mathbf{0_3} \\ \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{I_3} \end{bmatrix} \tag{3.1}$$

where $\mathbf{\Omega}^{\mathbf{e}}_{\mathbf{ie}}$ expresses the rotation of ECEF in inertial space, $\boldsymbol{\tau_s}$ is the update interval, and

$$\mathbf{F}^{\mathbf{e}}_{\mathbf{21}} = \left[-(\widehat{\mathbf{C}^{\mathbf{e}}_{\mathbf{b}}})\wedge\right], \quad \mathbf{F}^{\mathbf{e}}_{\mathbf{23}} = \frac{2\widehat{\boldsymbol{\gamma}^{\boldsymbol{e}}_{\boldsymbol{ib}}}\widehat{\boldsymbol{r}^{\boldsymbol{e}}_{\boldsymbol{ib}}}^{T}}{\boldsymbol{r}^{e}_{eS}(\widehat{\mathbf{L}_{\mathbf{b}}})\left|\widehat{\boldsymbol{r}^{\boldsymbol{e}}_{\boldsymbol{ib}}}\right|}$$

$$\mathbf{Q}^{\mathbf{e}}_{\mathbf{INS}} = \begin{bmatrix} S_{rg}\mathbf{I_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} \\ \mathbf{0_3} & S_{ra}\mathbf{I_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} \\ \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} \\ \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & S_{bad}\mathbf{I_3} & \mathbf{0_3} \\ \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} & S_{bgd}\mathbf{I_3} \end{bmatrix} \boldsymbol{\tau_s} \tag{3.2}$$

where S_{rg}, S_{ra}, S_{bad} and S_{bgd} are the power spectral densities of the gyroscope noise, accelerometer noise, gyroscope bias variation and accelerometer bias variation, respectively.

$$\mathbf{H}^{\mathbf{e}}_{\mathbf{G,k}} = \begin{bmatrix} \mathbf{0_3} & \mathbf{0_3} & -\mathbf{I_3} & \mathbf{0_3} & \mathbf{0_3} \\ \mathbf{0_3} & -\mathbf{I_3} & \mathbf{0_3} & \mathbf{0_3} & \mathbf{0_3} \end{bmatrix}_k \tag{3.3}$$

The calculation is performed in an ECEF reference frame, the inertial sensors have an output rate of 100 Hz and GNSS (measurement) updates occur at 2 Hz.

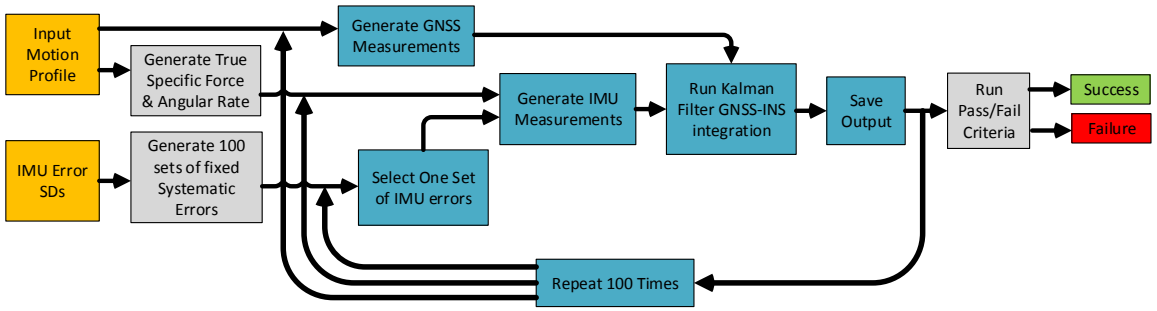


Figure 3.4: Flow Diagram of the Monte Carlo simulation algorithm. Inputs in yellow and Monte Carlo part in blue.

3.3.2 Algorithm Process

IMUs are modelled by adding errors of differing magnitudes to the output of a fictional perfect IMU to simulate IMUs of different grades. The values for each of the systematic errors are selected randomly for each run of a Monte-Carlo simulation and kept constant within that run. Thus, the only input needed to generate the set of systematic errors for all the runs of a Monte-Carlo simulation is the standard deviations of each systematic error. Noise sources are specified by their power-spectral-densities (PSDs) and assumed to be white.

IMU sensor noise is generated independently for each run of every test. Similarly, the GNSS measurement errors are also selected independently for each run of each Monte-Carlo simulation.

The process through which the program runs to make a Monte Carlo simulation of a single error distribution is shown in Figure 3.4. First the inputs listed in Table 3.1 are specified, noting that the reason these particular ranges were chosen is discussed in Section 3.3.6. The true motion profile is also an input, as described in Section 3.3.3, as are the GNSS parameters described in Section 3.3.5.

Parameter	Value or range used
number of runs in MC simulation	100
attitude initialisation error SD	0.5 deg (all axes)
Accel. Bias SD ($\boldsymbol{b_a}$)	1000 to 100,000 μg
Accel. Noise root-PSD ($\boldsymbol{w_a}$)	100 to 500 μg /$\sqrt{\text{Hz}}$
Accel. Scale factor error SD	0.06% to 3%
Accel. Cross axis sensitivity SD	0.025% to 1%
Accel. quantization level	0.01ms^{-2}
Gyro Bias SD ($\boldsymbol{b_g}$)	10 °/hr to 20 °/s
Gyro Noise root-PSD ($\boldsymbol{w_g}$)	0.01 to 1.8 deg/$\sqrt{\text{hour}}$
Gyro. Scale factor error SD	0.03% to 3%
Gyro. Cross axis sensitivity SD	0.02% to 2%
Gyro. g-dependent error SD	1 to 100 deg/hour/g
Gyro. quantization level	0.0002 rad/s

Table 3.1: The inputs to the Monte Carlo simulation and their values.

Next, values for the systematic IMU errors for each of the 100 simulation runs are chosen randomly from a zero-mean Gaussian distribution with standard deviations set to the parameter input values. For example, 100 sets of accelerometer biases are chosen and 100 3×3 gyroscope g-dependent error matrices. Then the Kalman filter tuning parameters are set using the input distribution SDs, in the manner described in Section 3.3.4.

The next step is to calculate the true specific force and angular rate measurements from the true motion profile and then use all the systematic and stochastic errors to create simulated IMU outputs. The measured specific force is calculated as,

$$\widetilde{\boldsymbol{f}_{\boldsymbol{ib}}^{\boldsymbol{b}}} = \boldsymbol{b_a} + \mathbf{A_a}\boldsymbol{f}_{\boldsymbol{ib}}^{\boldsymbol{b}} + \boldsymbol{w_a} \tag{3.4}$$

and the measured angular rate is calculated as,

$$\widetilde{\boldsymbol{\omega}_{\boldsymbol{ib}}^{\boldsymbol{b}}} = \boldsymbol{b_g} + \mathbf{A_g}\boldsymbol{\omega}_{\boldsymbol{ib}}^{\boldsymbol{b}} + \mathbf{G_g}\boldsymbol{f}_{\boldsymbol{ib}}^{\boldsymbol{b}} + \boldsymbol{w_g}. \tag{3.5}$$

Note that this neglects the non-linearity terms of Equations 2.1 and 2.2. Note also that the filter ignores the effects of $\mathbf{A_a}$, $\mathbf{A_g}$ and $\mathbf{G_g}$, effectively assuming that they are $\mathbf{I_3}$, $\mathbf{I_3}$ and $\mathbf{0_3}$, repectively. That is,

$$\widetilde{\boldsymbol{f}_{\boldsymbol{ib}}^{\boldsymbol{b}}} = \boldsymbol{b_a} + \boldsymbol{f}_{\boldsymbol{ib}}^{\boldsymbol{b}} + \boldsymbol{w_a} \tag{3.6}$$

and,

$$\widetilde{\boldsymbol{\omega}_{\boldsymbol{ib}}^{\boldsymbol{b}}} = \boldsymbol{b_g} + \boldsymbol{\omega}_{\boldsymbol{ib}}^{\boldsymbol{b}} + \boldsymbol{w_g}. \tag{3.7}$$

Simulated GNSS positions and velocities are also created from the true motion profile (see Section 3.3.5). Then inertial navigation equations (see Section 2.2.3) and the basic Kalman filter INS/GNSS integration algorithms (discussed in Section 3.3.1) are run for each of the 100 sets of simulated IMU and GNSS measurements. The results of all of the simulations are saved.

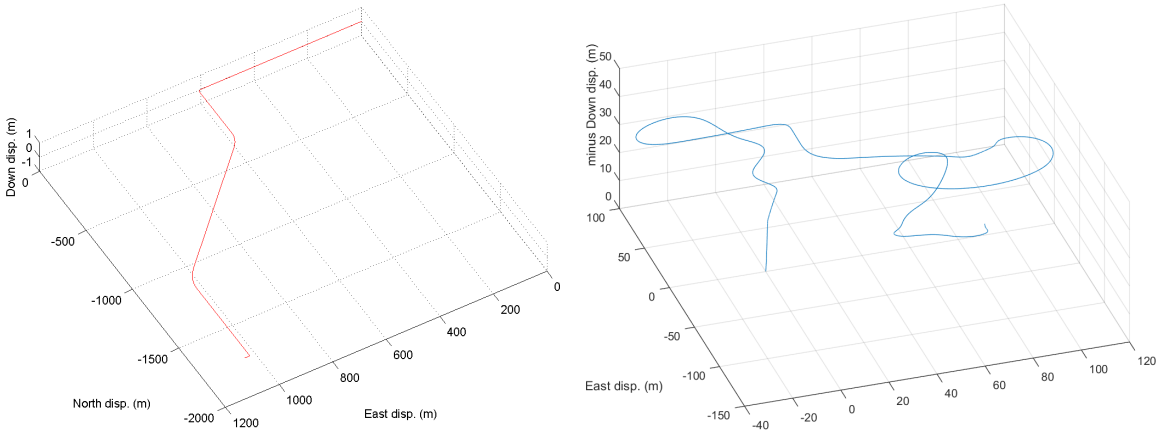
Finally, summary statistics are calculated for all 100 simulation runs of that distribution and comparisons between the KF's estimation errors and the uncertainty that it calculates for each state, as discussed in Section 3.2, are made. This gives the result of pass or fail for the particular error distribution.

3.3.3 Simulation Motion Scenarios

In the research presented in this chapter, two truth motion profiles for the Monte Carlo simulations are used.

The first consists of a typical car motion lasting three minutes and containing three turns. It is shown in Figure 3.5a. This motion profile is fairly representative of the navigation scenario that a typical consumer grade IMU might be used in. However, to avoid making the results specific to cars or land vehicles land-vehicle motion constraints [174] are not implemented. These would reduce the INS drift and could make the calibration of the IMU systematic errors easier. Additionally, the relatively limited number of different manoeuvres make it relatively difficult to separately observe the different IMU errors in this scenario.

The second motion scenario tested simulates a quadcopter. This was chosen to provide a contrasting example with much higher dynamics. Motion representative of a fighter aircraft

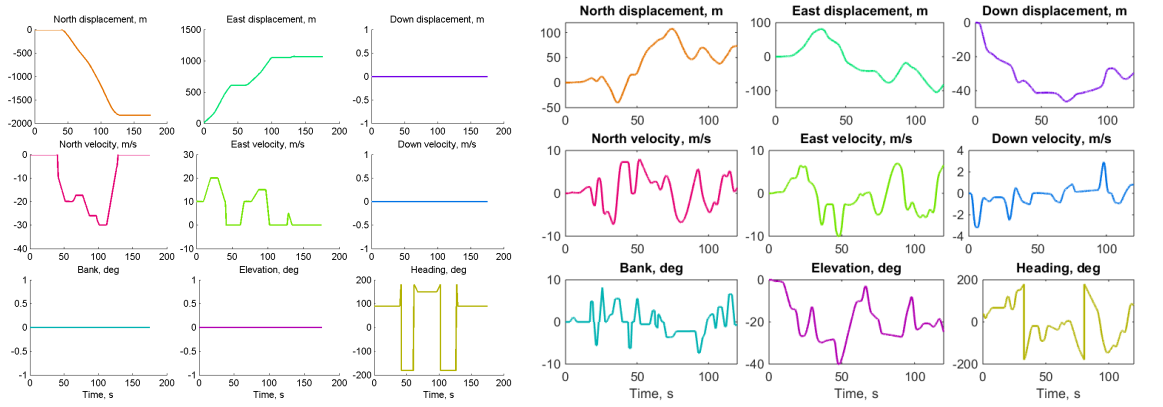


(a) Car (b) Quadcopter

Figure 3.5: 3D projection of the two truth motion profiles used for the Monte Carlo simulations.

or a missile could have been used, but neither of these are realistic applications for low-cost IMUs. As such, motion was generated representing a small quadcopter of the type available to hobbyists for a few hundred dollars, to carry small payloads such as video cameras.

The quadcopter simulated has a maximum speed of 10 m/s, an elevation angle dependent on its speed and achieves high bank angles when turning. These are the advertised performance characteristics of a market-leading model, the DJI Phantom [175]. The two minutes of 3D motion and resulting position, velocity and attitude are illustrated in Figures 3.5b and 3.6b, respectively. Note the frequent sharp turns and that the elevation angle depends on the speed, because the rotors do not pivot with respect to the quadcopter body.



(a) Car (b) Quadcopter

Figure 3.6: Components of the two simulated truth motion profiles used for the Monte Carlo simulations.

A pedestrian motion scenario has not been tested because IMUs are typically used for pedestrian dead reckoning (PDR) using step detection, which does not require sensor calibration [60, 79]. Inertial navigation using consumer-grade sensors is generally only viable using a foot-mounted IMU, for which zero-velocity updates (ZVUs) are used to calibrate

the IMU, requiring a different KF design. Representative pedestrian motion is also difficult to generate [176]. PDR was discussed in more detail in Section 2.3.2.3.

3.3.4 Filter Tuning

As the true distribution of the sensor errors is known, the KF could be tuned with precisely these values. However, as is standard practice, the error distributions and noise power spectral densities are over-bounded in the KF integration [1]. This helps maintain filter stability, as those error sources which are not estimated and the effects of any system non-linearities can appear as noise to the filter. The noise root-PSDs assumed within the KFs are set to double the noise root-PSDs, used to simulate the inertial sensors, plus an additional factor detailed below. Additionally, the initial position, velocity and attitude state uncertainties are overmodelled by a factor of 2. This level of noise over-bounding and initial state overmodelling is intended to replicate the level of compromise between performance and stability commonly used in industry.

The tuning of the sensor bias states could be use only the stochastic parameters of the biases themselves (as initially assumed in [7]). However, in initial tests it was observed that particular state estimates were frequently becoming unstable due to systematic errors that a basic 15-state Kalman filter does not directly model, such as scale factor and cross-coupling errors. Thus, consideration of these errors was included in the filter tuning for its bias states.

The basic Kalman filter (Section 3.3.1) assumes that all the sensor errors are a combination of bias and noise, so any other error types will manifest either in the estimates of bias or as additional noise. The effects of the higher-order errors on the bias state estimates are the average effect that the higher-order errors have on the measurements of specific force and angular rate. This is approximately equivalent to the effect of the higher-order errors on the average specific force and angular rate applied in the simulation.

Assuming that the IMU is, on average, level during the motion, the angular rates and horizontal specific forces average across the whole simulation to a number close to zero, while the vertical (Z) specific force averages to about 9.8 ms^{-2} due to reaction to the Earth's gravity. For example the average true values over the whole land-vehicle motion profile is $\overline{\boldsymbol{f}_{\boldsymbol{ib}}^{\boldsymbol{b}}} = [-0.0572, 0.0731, 9.8098]^T \text{ ms}^{-2}$ and $\overline{\boldsymbol{\omega}_{\boldsymbol{ib}}^{\boldsymbol{b}}} = [-0.2208, -0.3042, -0.5736]^T \times 10^{-4}$ rad/s. Consider the scale-factor-and-cross-coupling error matrix, $\mathbf{A_a}$, where, $s_{a,i}$ is the scale factor error of the i-axis sensor and $c_{a,ij}$ is the cross coupling error on the i-axis sensor of the j-axis specific force, so

$$\mathbf{A_a} = \begin{pmatrix} 1 + s_{a,x} & c_{a,xy} & c_{a,xz} \\ c_{a,yx} & 1 + s_{a,y} & c_{a,yz} \\ c_{a,zx} & c_{a,zy} & 1 + s_{a,z} \end{pmatrix}. \tag{3.8}$$

The entries in the third column are those depending on the vertical-axis specific force. As a result these errors appear (on average) to the simplified IMU model used in the basic Kalman filter (see Section 3.3.1) as an additional bias and noise on the accelerometer signals, whereas the other six errors only appear as noise. In the equivalent A-matrix for the gyroscope all 9 errors must be absorbed by the system noise. However, for the gyroscope g-dependent error matrix, which is a 3×3 matrix describing the effect of the specific force on the angular

rate measurements, the 3 components corresponding to the z-axis specific force appear to produce an additional bias, and all components appear to produce additional gyroscope noise.

In order to account for the bias in the tuning, the state uncertainties associated with the accelerometer biases are set equal to the root sum of squares (RSS) of the accelerometer bias SD and g times the greater of the scale factor and cross-coupling error SDs. This form is chosen, because it allows arbitrary orientation of the IMU. Similarly, the gyroscope bias uncertainties were set to the RSS of the gyroscope bias SD and the SD of the g-dependent bias associated with a specific force of g.

In order to maintain filter stability it is necessary to include additional system noise (represented by the system noise covariance matrix, $\mathbf{Q}$) to account for the noise-like effect of the higher-order systematic errors. The best way to tune the KF noise parameters to account for these errors depends both on the dynamics of the motion and the magnitude of the unestimated errors. These error sources add a large amount of noise to the sensor outputs during a manoeuvre and have little impact the rest of the time. In a real system it is likely that the designer would determine the amount of extra noise empirically, using trial-and-error until the system remains stable. Some authors also suggest using a system model that depends on the dynamics (see [1]), where the value of $\mathbf{Q}$ may be varied in real time depending on the dynamics.

The intention of the simulation in this chapter is to test the most basic KF integration possible so a constant amount of additional system noise noise covariance is added that is proportional to the variance of the unestimated systematic errors. This system noise is represented by a factor related to the level of sensor noise (double the root-PSD) and an additional factor that is proportional to the level of unestimated systematic errors (e.g. the scale-factor and cross-coupling) and which varies according to the motion profile, i.e. the level of dynamics expected. This second factor is intended to represent the designer adding small amounts of additional noise on a trial-and-error basis. These levels of additional noise were all tested numerically, prior to the full simulation. The system noise used in the KF to represent the behaviour of the accelerometers is given by,

$$S_a^{\mathrm{KF}} = 2^2 S_a^{\mathrm{IMU}} + (\max(\sigma_{c_a}, \sigma_{s_a}) k_f)^2 t_a, \tag{3.9}$$

where S_a^{KF} is the accelerometer noise PSD modelled within the KF, S_a^{IMU} is the accelerometer noise PSD used to generate the IMU model, k_f is a constant of proportionality to account for the level of variation in the specific force during the motion and, σ_{c_a} and σ_{s_a} are SDs of the accelerometer cross-axis sensitivity and scale factor error used to generate the IMU model, respectively. t_a is the smoothing time described below.

Similarly, the system noise used in the KF to represent the behaviour of the gyroscopes is given by,

$$S_g^{\mathrm{KF}} = 2^2\, S_g^{\mathrm{IMU}} + (\max(\sigma_{c_g}, \sigma_{s_g}) k_\omega)^2 t_a + (\sigma_{G_g} k_f)^2 t_a, \tag{3.10}$$

where S_g^{KF} is the gyroscope noise PSD modelled within the KF, S_g^{IMU} is the gyroscope noise PSD used to generate the IMU model, k_ω is a constant of proportionality to account for the level of variation in the angular rate during the motion and, σ_{c_g}, σ_{s_g} and σ_{G_g} are SDs of the

gyroscope cross-axis sensitivity, scale factor error and g-dependent error used to generate the IMU model, respectively.

The constants of proportionality, k_f and k_ω are calculated based on statistical measurements of the true motion. These two constants have the same units as specific force and angular rate, respectively. They are the average absolute difference in specific force or angular rate between the output smoothed over 10 GNSS epochs (5 seconds) and over 1 epoch for each motion profile. That is,

$$k_f = \text{mean}(\,|\,\text{m}(\boldsymbol{f^b_{ib}}, t_u) - \text{m}(\boldsymbol{f^b_{ib}}, 10t_u)\,|\,) = \begin{cases} 0.1058 \text{ ms}^{-2}, & \text{for car motion} \\ 0.2302 \text{ ms}^{-2}, & \text{for quadcopter motion} \end{cases} \tag{3.11}$$

where $\boldsymbol{f^b_{ib}}$ is the specific force and t_u is the KF measurement update time (0.5 seconds). The function "$\text{m}(a, t)$" takes a moving average of the time series a over t seconds. "$|\quad|$" is the absolute value operator.

$$k_\omega = \text{mean}(\,|\,\text{m}(\boldsymbol{\omega^b_{ib}}, t_u) - \text{m}(\boldsymbol{\omega^b_{ib}}, 10t_u)\,|\,) = \begin{cases} 0.008228 \text{ rad/s}, & \text{for car motion} \\ 0.042038 \text{ rad/s}, & \text{for quadcopter motion} \end{cases} \tag{3.12}$$

where $\boldsymbol{\omega^b_{ib}}$ is the angular rate. The time constant for Equations 3.9 and 3.10 is 10 times the GNSS (measurement) update rate ($t_a = 10t_u$), as this is the period of the moving average.

3.3.5 GNSS parameters

The GNSS simulation settings are chosen to simulate the operation of consumer-grade GNSS user equipment in a relatively benign signal environment. A production system would also need to include integrity checks to ensure that the GNSS positioning was of good quality before beginning INS calibration. IMU calibration when GNSS positioning is working well is an easier case than it would be in general. However, this is reasonable as one would only try to use GNSS to calibrate the INS when good reception is available.

As the focus of this thesis is on the IMU calibration, a relatively crude GNSS model is used. Bias-like GNSS errors were neglected as they only affect position determination and not the INS calibration which is the aim of this research. Additionally, as the simulation is relatively short (3 minutes), the time variation of GNSS systematic errors has been neglected. A constellation of 30 satellites was simulated. The effect of code tracking error is simulated by white noise with a SD of 1m and 0.02 m/s on the pseudo-range and pseudo-range-rate, respectively. Noting that the KF update-interval exceeds the correlation time of typical GNSS tracking errors. Positions and velocities are calculated by un-weighted iterated least squares.

The measurements used for the Kalman filter are GNSS-like generic position and velocity measurements, i.e. loosely-coupled integration. The associated measurement noise SD KF tuning parameters are 2.5m and 0.1 m/s on each axis for the position and velocity, respectively. These values are based on the simulations in [1], this is based on the simulated GNSS noise adjusted to account for time-correlation. Measurement (GNSS) uncertainty that is the

same for all axes is a slight over-bounding of the minimal case, but it is used to keep the integration as general as possible.

3.3.6 Determining the Search Space

Manufacturer	Bosch	ST	Invensense	Xsens
Model	BMA180 [52]	L3G4200D [24]	MPU-9150 [25]	MTi-G [151]
Type	accelerometer	gyroscope	single-chip IMU	factory-calibrated IMU
Accelerometer			±0.784(x&y)	
Bias	$\pm 0.588 \text{ms}^{-2}$	n/a	to 1.47(z) ms^{-2}	0.02ms^{-2}
Noise	$0.00147 \text{ms}^{-2}/\sqrt{\text{Hz}}$	n/a	$0.0039 \text{ms}^{-2}/\sqrt{\text{Hz}}$	0.002 to 0.004 $\text{ms}^{-2}/\sqrt{\text{Hz}}$
Scale factor	±1.5% to 3%	n/a	±3%	±0.03%
Cross-coupling	1.75%	n/a	not specified	aligned to ±0.1°
Non-Linearity	0.15 to 0.75%FS	n/a	0.5%FS	not specified
Gyroscope				
Bias	n/a	± 10 to 75 °/s	±20 °/s	±1 °/s
Noise	n/a	$0.03 °/\text{s}/\sqrt{\text{Hz}}$	$0.005 °/\text{s}/\sqrt{\text{Hz}}$	0.05 to $0.1 °/\text{s}/\sqrt{\text{Hz}}$
Scale factor	n/a	±4% [48]	± 3%	not calibrated
Cross-coupling	n/a	not specified	± 2%	aligned to ±0.1°
Non-Linearity	n/a	0.2% of FS	0.2 % FS	not specified

Table 3.2: A selection of sensor error specifications derived from their datasheets, in the units given. The Bosch, ST and Invensense are consumer-grade MEMS. The Xsens is a factory calibrated MEMS IMU, which costs around $2500. Where a range is given this parameter depends on the full-scale measurement range selected.

There are a large number of possible variables which could be changed to run different Monte Carlo simulations which were listed in Table 3.1. In order to be able to run enough simulations to properly examine the failure boundary using limited computational resources the Monte-Carlo simulations each consist of 100 runs.

Given there are a vast number of possible combinations of the 12 possible input parameters, some assumptions were made to reduce the number of potential combinations.

Firstly, only levels of error that exist for real sensors from tactical to consumer MEMS grade were tested. Thus, no zero-bias or zero-noise tests are presented, nor were any sensors simulated with unrealistically large errors. The ranges for each parameter chosen are given in Table 3.1. Some real sensor specifications are provided in Table 3.2 to demonstrate that the ranges tested encompass currently available sensors.

Secondly, some, but not all, of the possible inputs are varied. For instance, the sensor quantisation and the attitude initialisation error SDs are fixed. Also, it seems very unlikely that certain combinations of errors exist, such as high cross-coupling error with extremely accurate scale factor. For this reason, some of the errors are varied together, with one parameter for the higher-order accelerometer unestimated errors encompassing, accelerometer scale factor error and accelerometer cross-coupling, and a second parameter for the higher-order gyroscope unestimated errors which combines: gyroscope scale factor error, gyroscope

cross coupling and gyroscope g-dependent error. These two error parameter sets are split into five levels for testing, tactical, low, medium, high and very high, which are given in Table 3.3.

Parameter std. dev.	tactical	low	medium	high	very high
Accelerometer					
Scale factor error	0.06%	0.5%	1%	2%	3%
Cross axis sensitivity	0.025%	0.25%	0.5%	0.75%	1%
Gyroscope					
Scale factor error	0.03%	0.3%	1%	2%	3%
Cross-axis sensitivity	0.02%	0.2%	1%	1.5%	2%
g-dependent error (°/hour/g)	1	5	10	50	100

Table 3.3: The five levels of higher-order unestimated IMU errors tested.

This leaves six different distribution parameters to test: accelerometer bias; gyroscope bias; accelerometer noise; gyroscope noise; and the two higher-order error parameters. As the aim is to find the border between success and failure the requirement is to find a five-dimensional subspace of this six dimensional space, analogous with how a surface is a two-dimensional subspace of three-dimensional space.

First, the filter behaviour over the whole the search space were determined by testing points over a coarse grid (7 ($\boldsymbol{b_a}$) ×9 ($\boldsymbol{b_g}$) ×5 ($\boldsymbol{w_a}$) ×5 ($\boldsymbol{w_g}$) ×5 (higher-order accelerometer) ×5 (higher-order gyroscope)). Then having identified the general structure of the space, border regions were re-searched on a much finer grid to find the boundary, using a strategy that tests along one parameter (e.g. $\boldsymbol{b_a}$) until adjacent points are found where one is a pass and the other a fail, then incrementing a second parameter (e.g. $\boldsymbol{b_g}$) and returning to varying the first. This finer grid has a geometrical spacing, that is, the values of the points are a geometrical progression, where each point is 110% of the value of the previous point. This means that the grid appears equally spaced when viewed on a log scale. Therefore each graph in Section 3.4 has more data points along the pass-fail boundary than elsewhere.

3.4 Results

Discussion of the results of the simulations is split into three parts. The first two parts concern the road vehicle simulation: Section 3.4.1 presents results with very small (tactical) higher-order error parameters (scale factor, cross coupling and gyroscope g-dependent error); Section 3.4.2 presents results with higher-order errors large enough to impact performance. Section 3.4.3 examines the results of the quadcopter simulations.

The states which diverge the fastest and thus cause failure are the heading and the z-axis gyroscope bias, which is not particularly surprising when one considers that these are only indirectly corrected by the GNSS positions and velocities, and also that while a error in

pitch or roll would cause a spurious acceleration due to incorrect removal of the specific force due to gravity, a heading error does not have this effect.

As it is impractical to present 6-dimensional diagrams, in this section 3-dimensional diagrams which show the variations in 3 parameters with the other 3 parameters fixed are presented.

3.4.1 Road vehicle results with small higher-order IMU errors

When the higher-order IMU error parameters are very small, the assumptions that were made by using a Kalman filter (KF) with only 15 states are reasonable. That is, in relation to these errors, the filter should be fit for purpose and, if the filter breaks, it must be for a different reason. However, there are error distributions for which the KF fails even with tactical unestimated errors. These are illustrated in Figures 3.7 and 3.8.

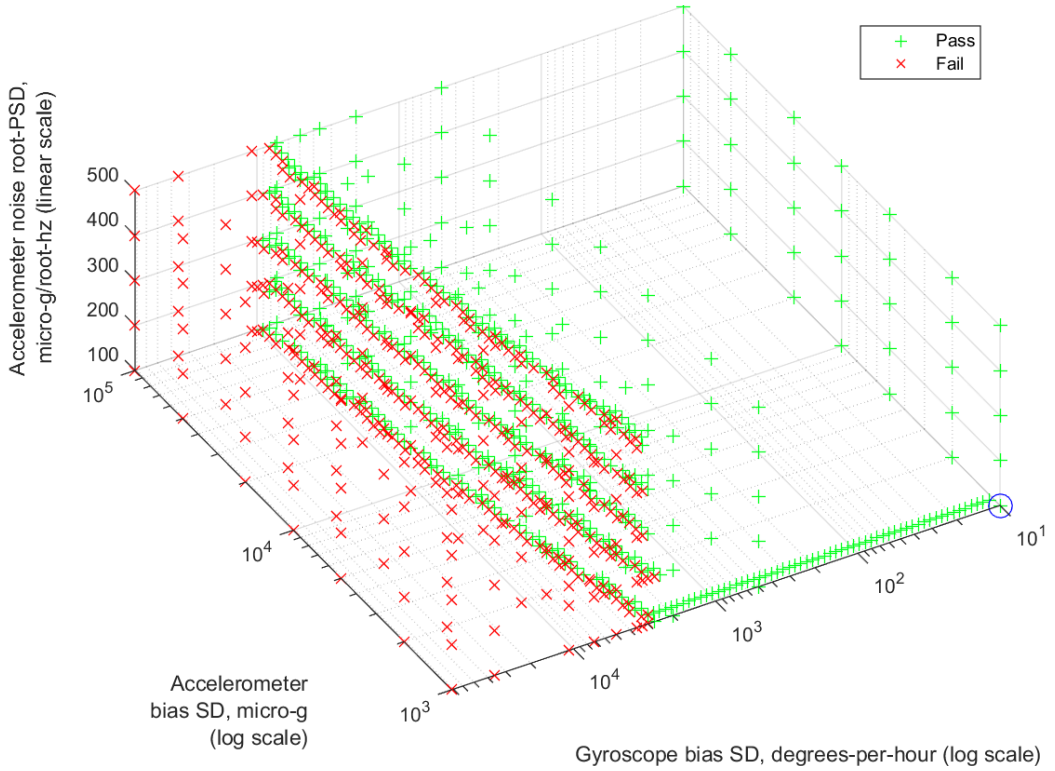


Figure 3.7: 3D subspace of the search space showing where the road vehicle KF fails for different values of $\boldsymbol{b_a}$, $\boldsymbol{b_g}$ and $\boldsymbol{w_a}$, fixing $\boldsymbol{w_g}$ at $0.01°/\sqrt{\text{hr}}$ and both unestimated error parameters at tactical level.

Observing the border between success and failure in these figures, it is clear that the most important error parameter is gyroscope bias. All of the distributions tested with a gyroscope bias standard deviation (SD) below 0.75 degrees per second (°/s or deg/s), equivalent to 2,705 degrees per hour (deg/hr or °/hr), were successful and all those tested above 2.6 °/s (9,400 °/hr) were failures. Note that this failure point is between the specified performance of factory calibrated IMUs and consumer-grade MEMS gyroscopes (see Table 3.2). This makes it clear that the accuracy of the attitude solution is key to INS/GNSS KF stability. This is primarily due to the use of the small angle approximation (see Section 2.4.1).

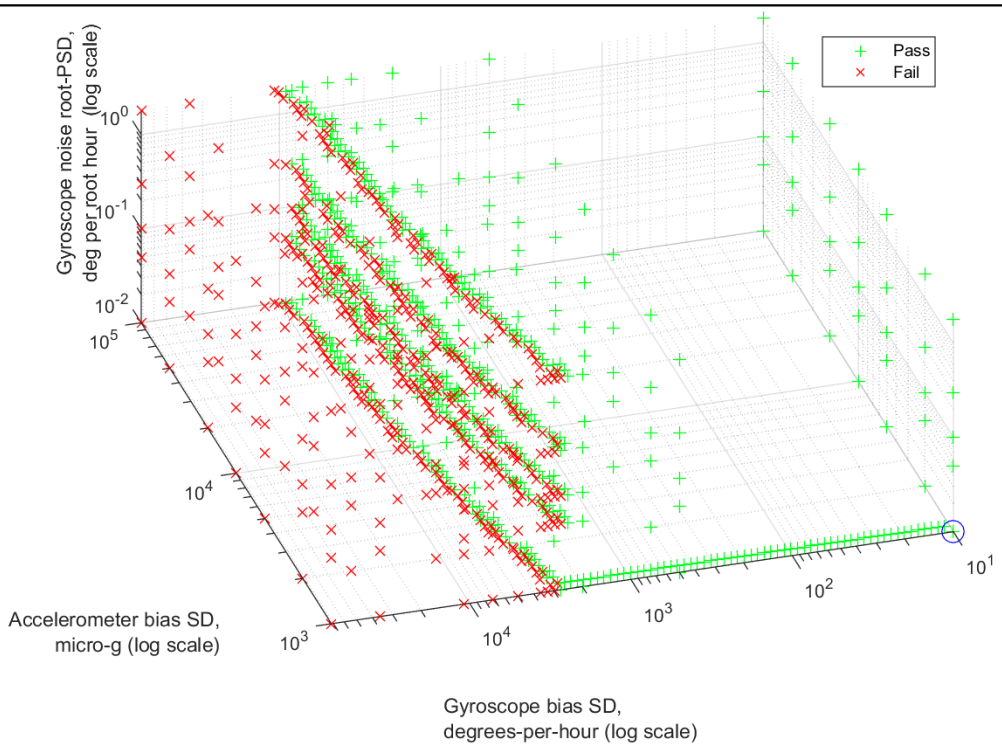


Figure 3.8: 3D subspace of the search space showing where the road vehicle KF fails for different values of $\boldsymbol{b_a}$, $\boldsymbol{b_g}$ and $\boldsymbol{w_g}$, fixing $\boldsymbol{w_a}$ at $100\mu\text{g}/\sqrt{\text{Hz}}$ and both unestimated error parameters at tactical level.

Variation of the border between success and failure according to the other IMU error parameters can also be observed. The accelerometer bias is most significant with the maximum tolerable gyroscope bias SD ranging from 2705 °/hr for a 1 milli-g accelerometer bias SD to 9337 °/hr for a 100 milli-g accelerometer bias SD. It is perhaps surprising that worse accelerometer bias performance allows the filter to cope with more gyroscope bias. A possible explanation is that the larger accelerometer bias uncertainty in the Kalman filter indirectly results in larger attitude uncertainties, enabling it to tolerate greater divergence from the small angle approximation. Another possible explanation is that the is that the correct distribution of state uncertainties is less adequately described by an ellipse as the angular uncertainty becomes larger.

The effect of the IMU's two noise parameters on the position of the boundary is even smaller. The accelerometer noise affects the maximum tolerable gyroscope bias SD by a factor of about 1.25 (Figure 3.7), while the effect of the gyroscope noise is less than a factor of 1.1 over the range simulated (Figure 3.8), despite this range being considerably wider than that of the accelerometer (see Table 3.1).

3.4.2 The effect of the unestimated first-order IMU errors

When the parameters which are not modelled as states in the Kalman filter are varied, it is known that the KF is only suitable if the effects of these parameters are not significant. Thus the question is: 'what size of error is insignificant?'.

As described in Section 3.3.6, 5 levels of higher-order errors of the gyroscopes and accelerometers were tested, as shown in Table 3.3. The gyroscope and accelerometer unestimated errors were varied separately. Both of these parameters were set to tactical (i.e. very low) for the results discussed in Section 3.4.1. Here, the results are presented for when one or both of the gyroscopes and accelerometers higher-order error parameters were set to low, medium, high or very high.

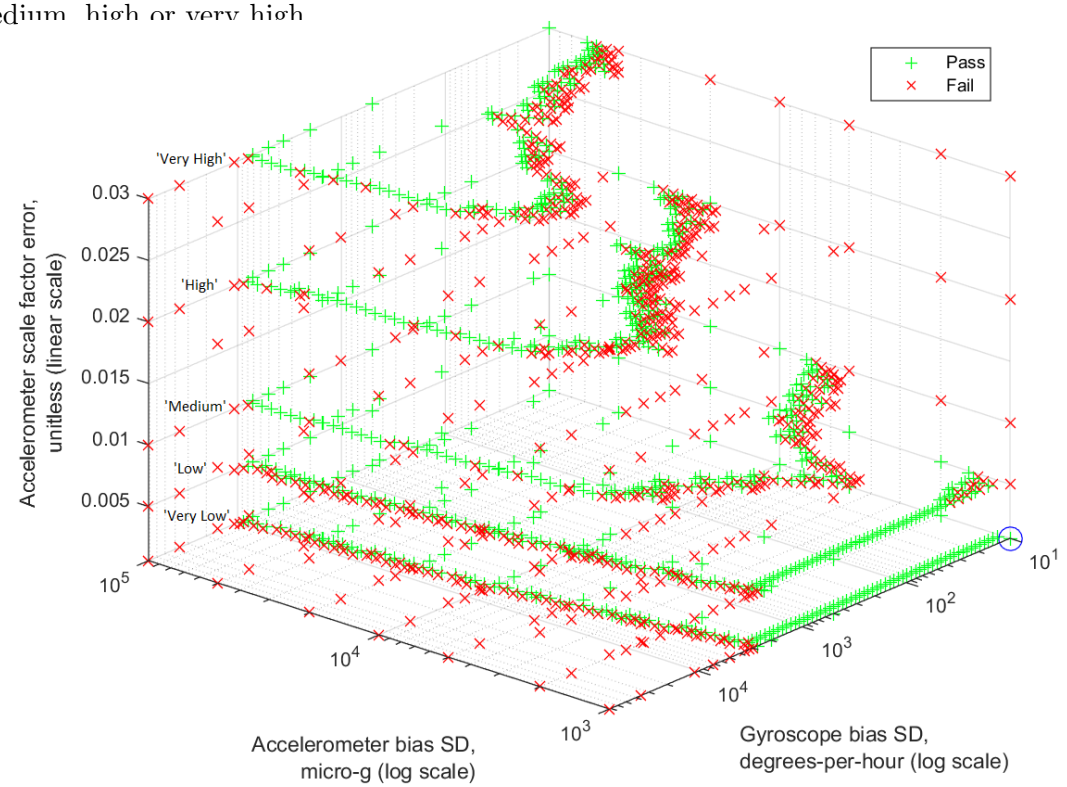


Figure 3.9: 3D subspace of the search space showing where the road vehicle KF fails for different values of $\boldsymbol{b_a}$, $\boldsymbol{b_g}$ and accelerometer unestimated errors, fixing $\boldsymbol{w_g}$ at $0.01°/\sqrt{\text{hr}}$, $\boldsymbol{w_a}$ at $100\mu\text{g}/\sqrt{\text{Hz}}$ and gyroscope unestimated errors at tactical level.

First, the effect of increasing the gyroscope and accelerometer higher-order error parameters individually is examined. Figure 3.9 shows the effect of increasing the higher-order accelerometer error parameters through each of the five levels while holding the two noise parameters and the gyroscope higher-order errors at tactical level. As the level of unestimated error increases (vertical axis) a new unstable area forms and grows in the low-bias corner of the graph, which combines with the pass/fail boundary discussed in Section 3.4.1, to only leave a small stable area when the higher-order errors are very high. Figure 3.10 is the equivalent showing the effect of the gyroscope unestimated error parameter. A similar effect can be observed. However, the range of bias values, for which the KF is stable, is smaller than for the equivalent level of accelerometer unestimated error. This leaves only an extremely small pass area in the high-bias corner when the unestimated error parameter reaches high and no passes at all for very high higher-order errors.

For a given level of unestimated error parameters, if the IMU's noise parameters are higher, the KF is stable for a greater range of bias values. Figures 3.11 and 3.12, are similar to the Figures 3.9 and 3.10, in that they show the variation of the higher-order error parameter of the accelerometers and gyroscopes, respectively. However, they show the part of the subspace where the noise parameter is at the highest end of the range tested for both the

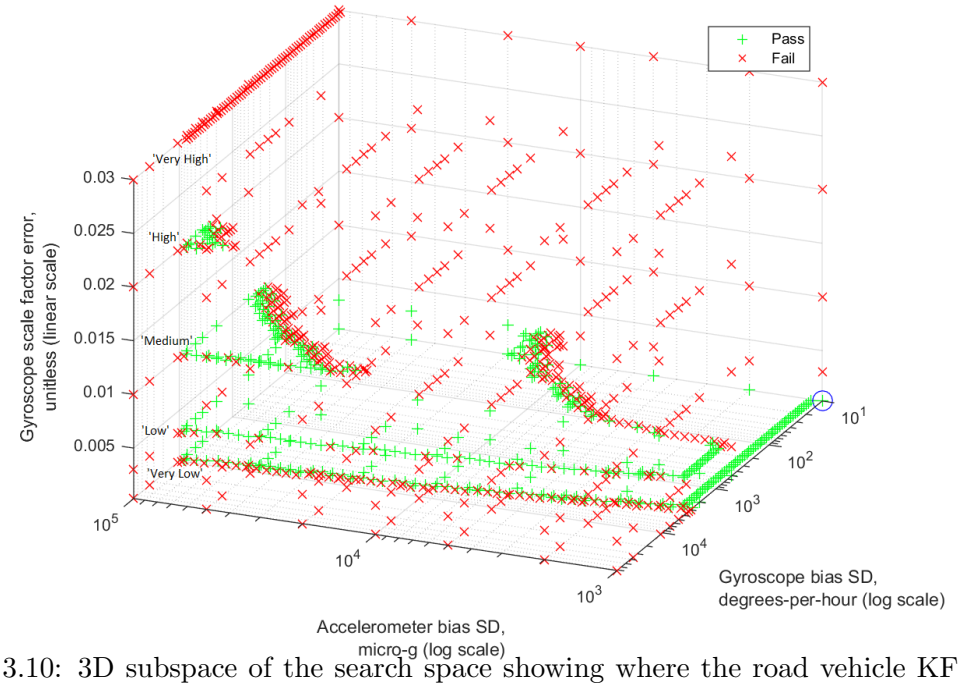


Figure 3.10: 3D subspace of the search space showing where the road vehicle KF fails for different values of $\boldsymbol{b_a}$, $\boldsymbol{b_g}$ and gyroscope unestimated errors, fixing $\boldsymbol{w_g}$ at $0.01°/\sqrt{\text{hr}}$, $\boldsymbol{w_a}$ at $100\mu\text{g}/\sqrt{\text{Hz}}$ and accelerometer unestimated errors at tactical level.

accelerometers and gyroscopes, rather than at the lowest end as was shown in Figures 3.9 and 3.10.

If one compares Figures 3.9 and 3.11, the range of bias values with stable KF performance with low accelerometer higher order errors is improved to the point of being the same as very low/tactical error, rather than only nearly the same, the range of bias values at medium and higher levels is also improved particularly at the lower gyroscope bias end, the stable area appears more rectangular and less v-shaped, range of stable bias values at medium higher order error performance is only sightly worse than the low. Greater actual noise, also means more overmodelling so the additional available noise appears to be covering the noise-like effects of the higher-order accelerometer errors better.

In a similar manner comparing Figures 3.10 and 3.12 also shows an improvement at the low level of gyroscope higher-order errors, which makes the range of stable bias values at low and tactical almost the same. The medium error level performance is also slightly better, the stable area growing slightly towards the low-gyroscope-bias direction. At the high and very high levels, the stable area is still very small and non-existent, respectively, as in Figure 3.10.

The fact that the filter's ability to tolerate unestimated error parameters depends on the level of noise means that, while a tactical level of higher-order error calibration is required for an IMU with tactical-level noise, the filter is less sensitive when the noise is at the level of a low-cost MEMS IMU. In this case only a low level of gyroscope and medium level of accelerometer unestimated error parameters are required. Note that, the overmodelling (exaggeration) of the noise is what is improving the stability of the filter not the noise itself, so the KF-assumed noise could be varied to improve stability.

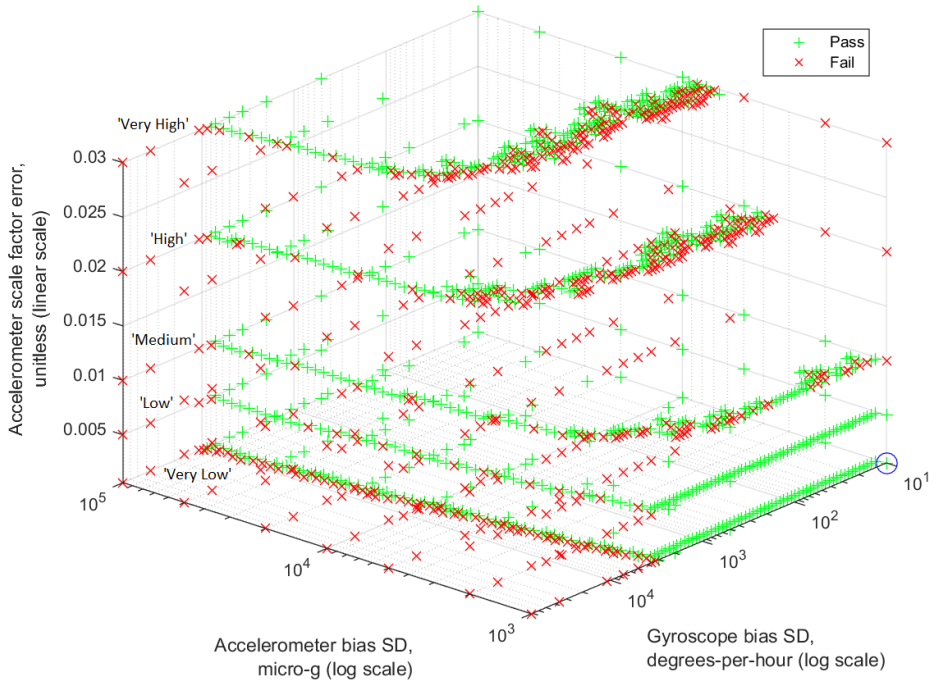


Figure 3.11: 3D subspace of the search space showing where the road vehicle KF fails for different values of $\boldsymbol{b_a}$, $\boldsymbol{b_g}$ and accelerometer unestimated errors, fixing $\boldsymbol{w_g}$ at $1.8°/\sqrt{\text{hr}}$, $\boldsymbol{w_a}$ at $500\mu\text{g}/\sqrt{\text{Hz}}$ and gyroscope unestimated errors at tactical level.

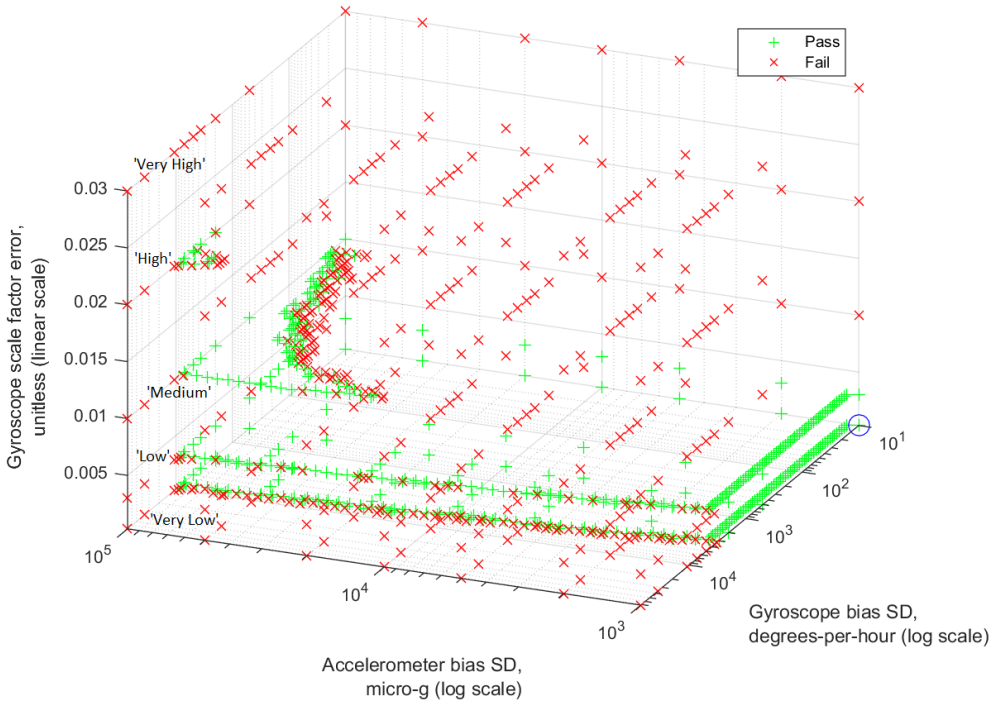


Figure 3.12: 3D subspace of the search space showing where the road vehicle KF fails for different values of $\boldsymbol{b_a}$, $\boldsymbol{b_g}$ and gyroscope unestimated errors, fixing $\boldsymbol{w_g}$ at $1.8°/\sqrt{\text{hr}}$, $\boldsymbol{w_a}$ at $500\mu\text{g}/\sqrt{\text{Hz}}$ and accelerometer unestimated errors at tactical level.

3.4.3 Quadcopter Results

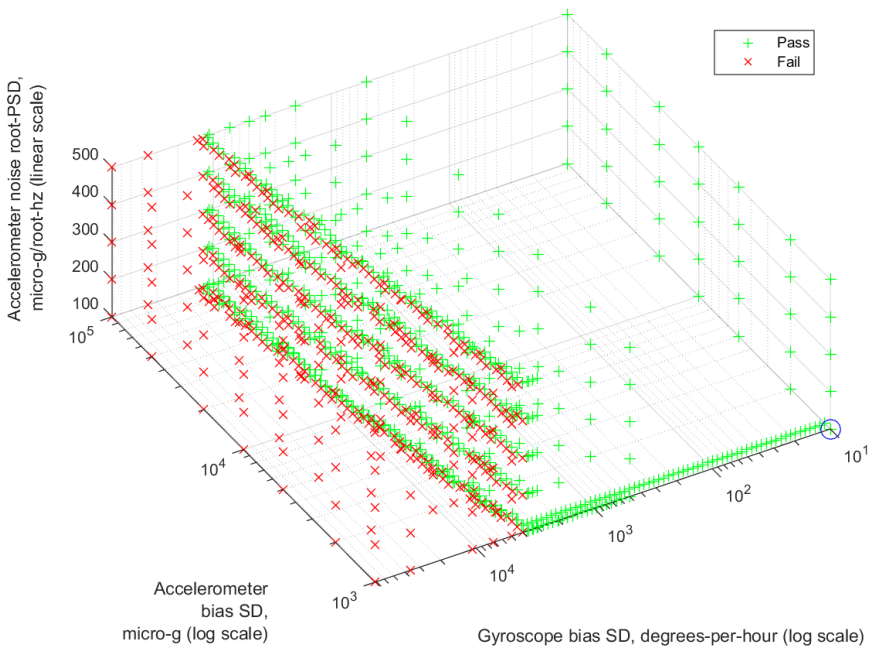


Figure 3.13: 3D subspace of the search space showing where the quadcopter KF fails for different values of $\boldsymbol{b_a}$, $\boldsymbol{b_g}$ and $\boldsymbol{w_a}$, fixing $\boldsymbol{w_g}$ at $0.01°/\sqrt{\text{hr}}$ and both unestimated error parameters at tactical level.

When the higher-order errors are at a low enough level to be insignificant, e.g. at tactical level, the filter stability is similar to that for the car motion. In Figure 3.13, which represents filter behaviour for these low levels of unestimated error, the behaviour is qualitatively almost exactly the same as the equivalent figure for car motion, Figure 3.7. It exhibits a maximum gyroscope bias SD beyond which the filter becomes unstable with some dependence on the accelerometer bias SD, but minimal variation with the accelerometer and gyroscope (not shown) noise levels. Numerically, the maximum possible gyroscope bias SD has increased slightly, ranging from 3684 °/hr for a 1 milli-g accelerometer bias SD to 12430 °/hr for a 100 milli-g accelerometer bias SD. This slight increase relative to the car motion could be due to the biases becoming more observable with the more frequent and varied manoeuvres.

As the level of higher-order errors increase, the differences in filter performance between the car and quadcopter motion profiles become larger. Figure 3.14 illustrates the variation in the stable area with different levels of accelerometer higher-order errors; it is the quadcopter analogue of Figure 3.9. Therefore, the maximum gyroscope bias uncertainty boundary is the same as in Figure 3.13, just as the boundary in Figure 3.7 was repeated in Figure 3.9. However, the range of accelerometer biases that are stable with lower gyroscope biases are considerably smaller with the quadcopter than the equivalent with car motion. Comparing levels of accelerometer higher-order error above very low, between Figures 3.9 and 3.14, the minimum acceptable accelerometer bias has reduced significantly, particularly around 100 degrees-per-hour gyroscope bias.

Figure 3.15 shows the effect of different levels of gyroscope higher-order errors. The analogous behaviour for car motion is shown in Figure 3.10. With very high levels of unestimated

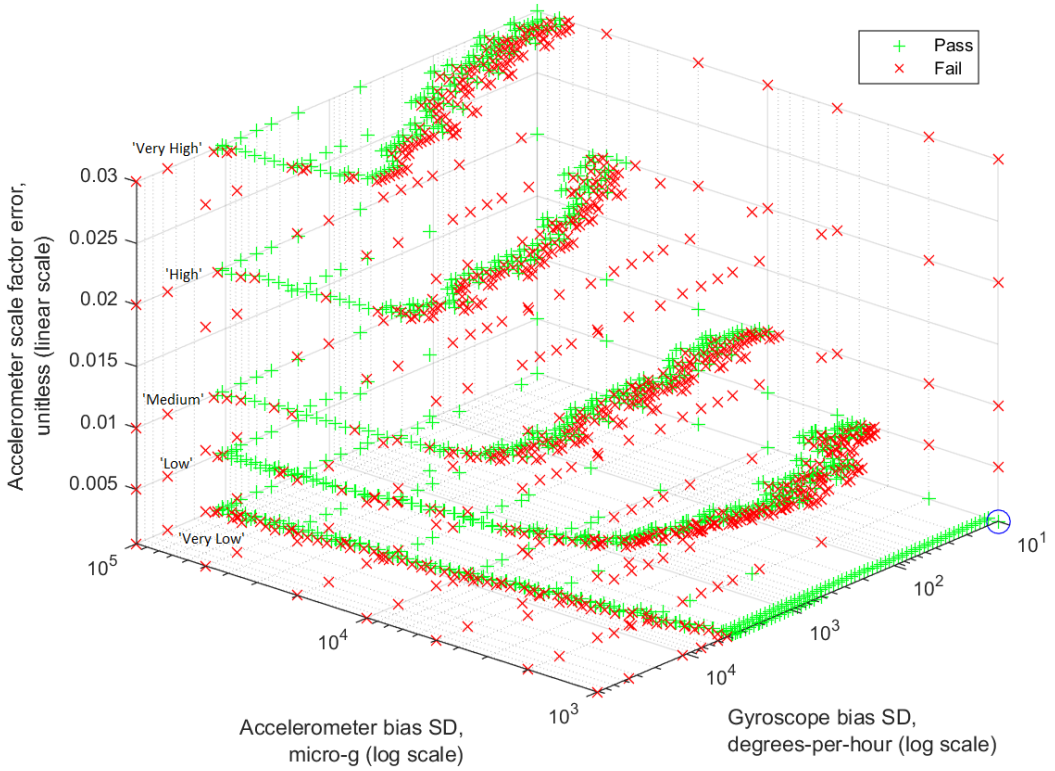


Figure 3.14: 3D subspace of the search space showing where the quadcopter KF fails for different values of $\boldsymbol{b_a}$, $\boldsymbol{b_g}$ and accelerometer unestimated errors, fixing $\boldsymbol{w_g}$ at $0.01°/\sqrt{\text{hr}}$, $\boldsymbol{w_a}$ at $100\mu\text{g}/\sqrt{\text{Hz}}$ and gyroscope unestimated errors at tactical level.

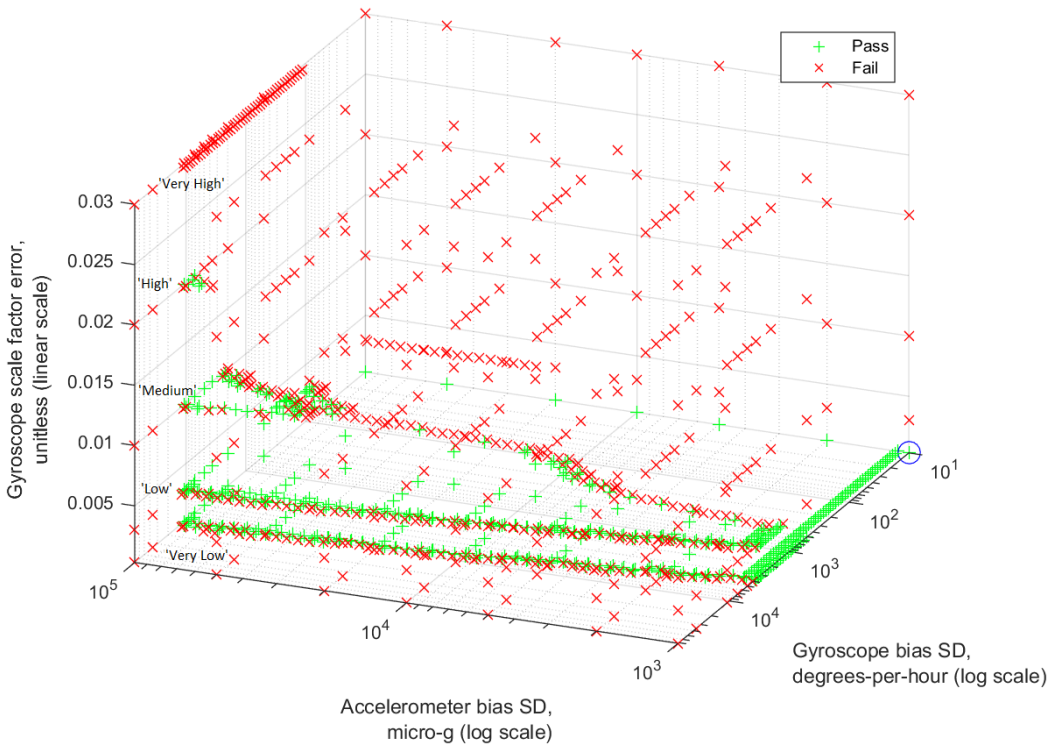


Figure 3.15: 3D subspace of the search space showing where the quadcopter KF fails for different values of $\boldsymbol{b_a}$, $\boldsymbol{b_g}$ and accelerometer unestimated errors, fixing $\boldsymbol{w_g}$ at $1.8°/\sqrt{\text{hr}}$, $\boldsymbol{w_a}$ at $500\mu\text{g}/\sqrt{\text{Hz}}$ and gyroscope unestimated errors at tactical level.

errors neither motion profile enables a stable KF. At high and medium levels of the higher order gyroscope errors the stability regions are very small for the car motion and smaller still for the quadcopter motion. This would make it very difficult to achieve stability by exaggerating the bias SDs assumed within the KF. When the unestimated errors are low the high-gyroscope-bias side of the stability area is very similar to the car motion. However, on the low-gyroscope-bias side, the quadcopter motion is only stable for higher levels of bias. Thus, the assumed gyroscope bias would need to be exaggerated to achieve stability.

These results indicate that that higher dynamics have a marked additional de-stabilising effect on the filter, and this manifests as the stable area shrinking significantly, pushing it further towards the high-bias corner. The impact of these errors on filter stability with quadcopter motion is approximately equivalent to the impact of unestimated errors half a level higher than on car motion.

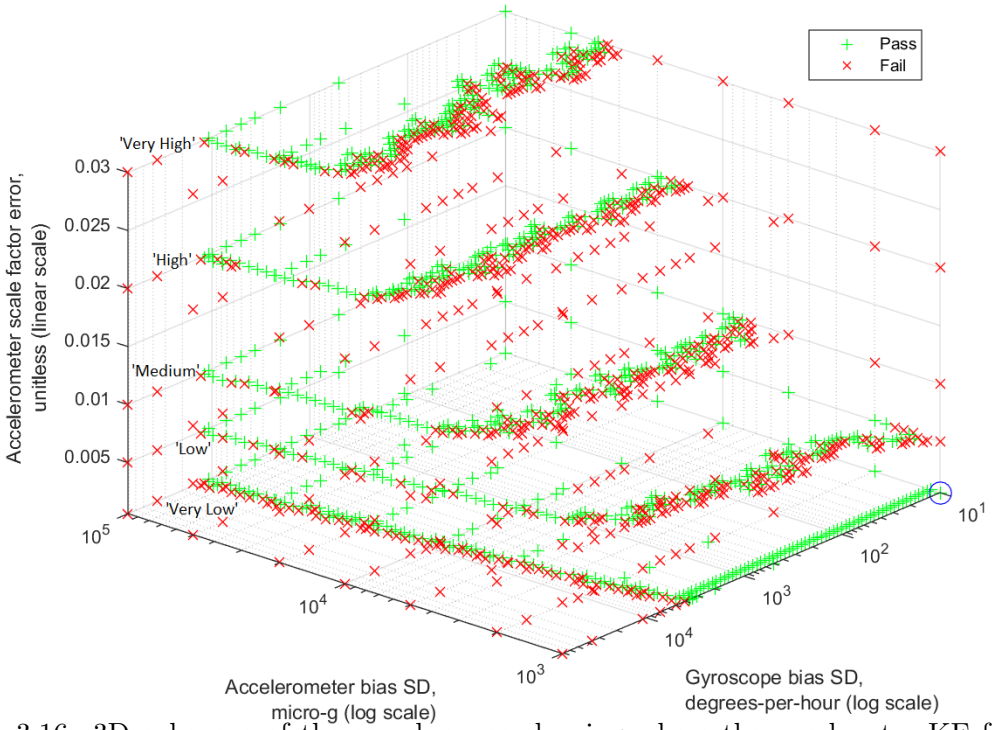


Figure 3.16: 3D subspace of the search space showing where the quadcopter KF fails for different values of $\boldsymbol{b_a}$, $\boldsymbol{b_g}$ and accelerometer unestimated errors, fixing $\boldsymbol{w_g}$ at $1.8°/\sqrt{\text{hr}}$, $\boldsymbol{w_a}$ at $500\mu\text{g}/\sqrt{\text{Hz}}$ and gyroscope unestimated errors at tactical level.

3.5 Discussion

Section 3.4.1 showed that when the levels of the unestimated higher-order errors were very low the noise level for both gyroscope and accelerometer noise made only a small difference to the maximum tolerable bias SD's. However, when the higher-order errors are larger, higher sensor noise aids filter stability, this can be seen by comparing Figures 3.14 and 3.16, noting the significant differences in the low-accelerometer bias SD boundary that comes about when the accelerometer and gyroscope noise parameters are at the high rather than the low end of the range tested.

A possible explanation for this and the other behaviour observed in the previous sections can be made by considering two things. First, consider how the unestimated error sources appear to the filter. As discussed in Section 3.3.4, their average effect appears as bias and the remainder must be treated as noise, which is not white, but correlated to the turns in the trajectory. The tuning of the initial state uncertainties was adjusted to account for the bias-like components. However, this compensation cannot be totally perfect, and the higher the unestimated errors the more significant this is. Second, while the KF tuning adds additional system noise to account for the noise-like behaviour, it adds it constantly throughout the simulation, as the filter is not adaptive. This means that most of the time more system noise is being added than necessary, while during manoeuvres, when the higher-order errors are excited, not enough is being added.

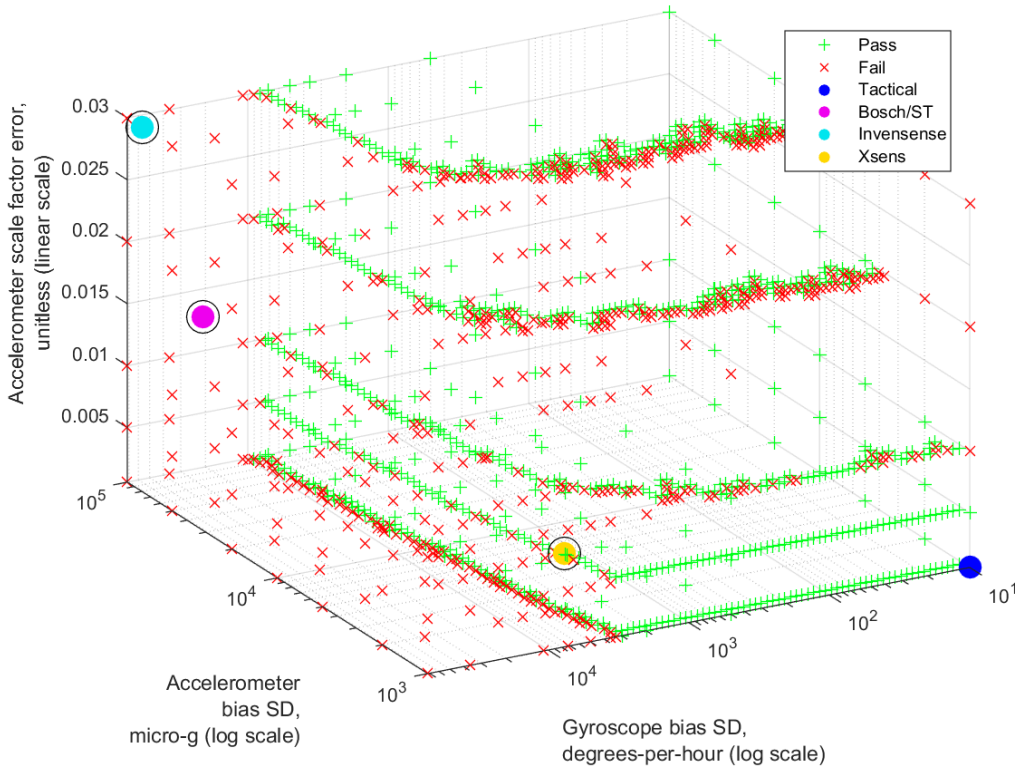
In order to absorb the unestimated IMU error parameters without modelling them as states there needs to be both enough overmodelling of the bias states to cover the extra bias from the higher-order errors and enough overmodelling of the noise to cover the extra noise. The filter tuning used here to account for biases and noise takes a constant multiple of the actual noise and bias to configure the KF for the system noise and bias state uncertainty (respectively). Therefore, when the noise or bias is higher, the overmodelling is also higher in absolute terms, as it is fixed in proportion to the noise and bias. This would explain why the filter can cope with higher unestimated errors when the biases or noise are higher.

This, and the slightly counter-intuitive results discussed in Section 3.4.1, leads to the observation that, in cases where a filter is close to the stability limits, it may be possible to aid KF stability by exaggerating the assumed accelerometer bias uncertainty when the unestimated error parameters are larger. It is possible that this is due to the additional accelerometer bias uncertainty keeping the attitude uncertainty higher which in turn reduces the destabilising effect of departures from the small angle approximation. It would also be possible to exaggerate the noise further during KF tuning, which may, as was discussed at the end of Section 3.4.2, allow the filter to remain stable with greater unestimated errors. However, exaggerating the system noise is a greater problem than exaggerating the bias (state) uncertainty, as rather than just affecting the convergence time by starting 'further away', additional system noise will reduce the accuracy of the estimates *after convergence.* Thus while exaggerating the system noise may avoid the filter failing the resulting filter would produce sub-optimal performance in other ways.

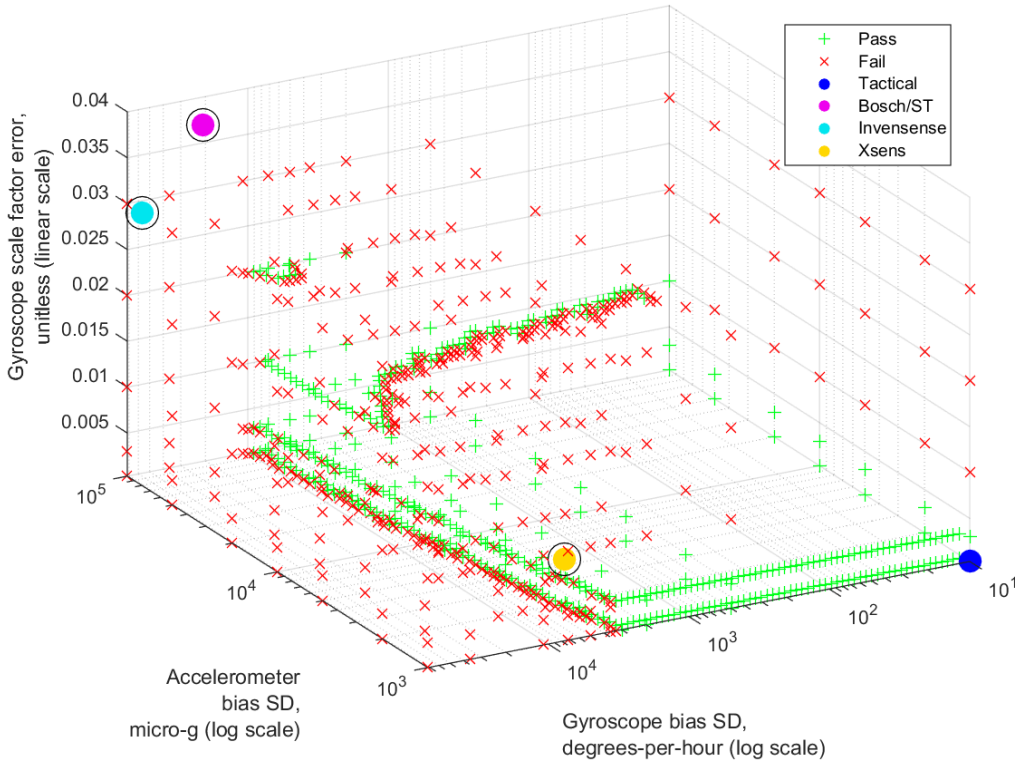
The differences between the motion profiles have already been discussed in Section 3.4.3.

One of the aims of this research is to enable navigation system designers to look up the specifications of the IMU they are planning to use and determine whether a basic Kalman filter INS/GNSS integration will be sufficient to calibrate the sensors in-run. This would save them the time and effort of designing and building the whole system only to find that the sensors' errors are too large for a standard KF to remain stable. In practice, there will be some sensors for which the KF can be stabilised by exaggerating some of the errors specified within the KF's tuning parameters.

As an example of how the results of this simulation could be applied, the real sensor models, whose specifications were given in Table 3.2, are plotted in Figure 3.17 as different coloured



(a) Accelerometer higher-order errors (as Figure 3.11)



(b) Gyroscope higher-order errors (as Figure 3.12)

Figure 3.17: 3D subspace of the search space showing where the road vehicle KF fails for different values of $\boldsymbol{b_a}$, $\boldsymbol{b_g}$ and either accelerometer or gyroscope unestimated errors, fixing $\boldsymbol{w_g}$ at $1.8°/\sqrt{\text{hr}}$, $\boldsymbol{w_a}$ at $500\mu\text{g}/\sqrt{\text{Hz}}$. Also shown are the specifications of selected MEMS sensors: a tactical grade IMU (dark blue), Xsens Mti-G (yellow), Invensense MPU-9150 (light blue), and an IMU comprising a STMicrotronics L3G4200D gyroscope and a Bosch BMA-180 accelerometer (magenta).

circles. It is clear that the two uncalibrated automotive MEMS sensors are well outside the KF stability limit; and the Xsens Mti-G IMU, which is factory calibrated, is right on the boundary. It is also noteworthy that the specification of the Xsens IMU as "aligned to 0.1 degree" [151], is very close to the tactical higher-order errors tested here. As such these results suggest that basic KF integration using it should be stable, provided the tuning is handled carefully.

The result that a standard INS/GNSS KF is unstable when using a uncalibrated low-cost MEMS IMU, is the result that would be expected from examining the literature (see Section 2.4.1). However, this chapter has shown that in addition to the gyroscope bias being too high, the errors which are not being modelled, e.g. scale-factor and cross coupling errors, are large enough that the filter would be unstable even if the biases were lower, for both the accelerometer (Figure 3.17a) and gyroscope (Figure 3.17b).

When considering the widest implications of the research presented in this chapter first one should consider how it is limited. First, while the two motion scenarios tested represent two likely applications of MEMS IMUs with markedly different levels of dynamics, they do not cover every possible use-case, rather they represent realistic extremes in terms of dynamics. While the results between the two cases are qualitatively similar the quantitative differences show that it is important use a situation representative of the intended application.

Second, this chapter used the simplest possible INS/GNSS integration Kalman filter possible. As it is currently configured, it is not appropriate for even medium higher-order gyroscope errors, which are relatively small in MEMS terms, although it is possible that further relaxing the filter tuning may regain stability at the cost of convergence speed and accuracy. We also might assume that if these unmodelled error sources were modelled then the performance reached would, at best, be similar to when they were insignificant. Estimating the higher-order errors as KF states would help mitigate their effects, at the expense of processing load. Even where these errors are difficult to observe, their inclusion as states provide the KF with a much more realistic system error model. However, it is very unlikely that this would enable toleration of gyroscope biases larger than can be tolerated with 'tactical level' higher-order errors

Bearing these limits in mind, a couple of important points can be made. The first is that basic KF integration can be sufficient to calibrate both accelerometer and gyroscope biases if the gyroscope bias SD is below the level of around 1 °/s. This is the aim of the INS/GNSS integration tested in this Chapter as laid out in Section 3.1. This means that any pre-calibration could concentrate on the gyroscope bias, and can leave estimating the accelerometer bias to the INS/GNSS integration, if the unestimated errors are sufficiently small. Fortunately, gyroscope bias is also among the easiest of the IMU errors to pre-calibrate, as it can be observed in a simple static test (see ZARUs in Section 2.3.3 and 2.5.1.3).

A major issue for calibration of all MEMS sensors is that the systematic errors vary with temperature. For example, the L3G4200D gyroscope bias variation with temperature is specified at 0.03 °/s/°C [24]. However, even with a 30°C operating range this is less than 1 °/s. (see also Section 6.6.2). Thus if the sensor were already calibrated before a change in

temperature then simply increasing the state uncertainty in the KF when the temperature changes should allow the INS/GNSS integration to compute the new bias.

3.6 Summary

The results presented in this chapter have found the boundary in terms of sensor performance at which an in-run calibration can be carried out using a basic KF integration. While precise boundary shape and position can vary depending on the motion profile, this is essentially both a maximum permissible level of gyroscope bias, and higher-order errors below the medium level for the accelerometers and the low level for the gyroscopes. As well as providing a reference for navigation system designers, this gives a quality target for both user-conducted calibration and array techniques, which were discussed in Sections 2.5.1.3 and 2.5.2

In the rest of this thesis, these target calibration levels will be used as part of the assessment criteria for both user-conducted calibration (Chapter 4) and array techniques (Chapter 6).

Chapter 4

User-conducted Calibration

This chapter concerns MEMS IMU calibration procedures which can be conducted by the end-user of the IMU without requiring specialist equipment such as rate tables or temperature controlled chambers, described in Section 2.5.1.

The chapter begins with the motivation for this avenue of study (Section 4.1). Then the basic principle of the calibration procedure assessed is laid out (Section 4.2). The various reference frames used for the calibration algorithm and the simulation are introduced (Section 4.3). Then the IMU error model assumed by the calibration algorithms and used in the simulation is presented (Section 4.4). The calibration algorithms which estimate the IMU's systematic errors from the recorded data is presented in Section 4.5, several alternative approaches are used some of which are new. Then simulation method itself is detailed in Section 4.6. The method used to assess the performance the calibration in each Monte Carlo simulation is presented in Section 4.7. The results of the simulation are presented and discussed in Section 4.8. Section 4.9 summarises the this chapter and makes recommendations based its findings.

4.1 Motivation: The purpose of user-conducted calibration procedures

In Chapter 3 the limits of using a Kalman-filter variant to conduct in-run calibration of IMUs were presented. The results confirmed that the levels of error present in a typical consumer-grade MEMS IMU were too high for a basic Kalman filter INS-GNSS integration to remain stable. One possible solution, which retains the MEMS sensors' key advantage of their low cost, while allowing their use with this basic integration architecture is to have the end-user conduct a simple calibration procedure to reduce the size of the sensors' errors to a level where the KF remains stable.

Fortunately, the results presented in Chapter 3 show that the level of sensor accuracy required need not be as low as that of a tactical-grade IMU. However, there are certain levels of performance that need to be achieved, and the idea of this chapter is to examine which factors determine the level of calibration that is achieved in different circumstances and to assess the effect of some innovations to the calibration procedure.

The purpose of a calibration procedure of the type examined in this chapter is to determine the IMU sensors' systematic errors. The user is instructed to carry specific, but simple,

manoeuvres which comprise a calibration procedure. During the procedure the measured output of the sensors is recorded and calculations are made from this output to estimate their actual systematic errors which can be used to calibrate/compensate their readings.

The standard way to demonstrate that a physical calibration procedure works is simply to repeat it several times and demonstrate that it gives the same calibration values consistently, when using the same sensors and the same calibration hardware (e.g. box and table, or rate-table), as in [155]. However, because a set-up like this produces the same answer several times in a row does not mean that that answer is correct, simply it demonstrates that the procedure is not overly affected by sensor noise or human-error, thus representing a necessary, but not sufficient, condition. Neither of these are trivial but if one wishes to test whether a calibration procedure really produces the correct results one has three alternatives.

First, one can use a laboratory calibration of the sensors as a reference. This requires expensive laboratory equipment that many users have no access to, and the reference calibration will only be valid for a particular range of temperatures as many MEMS sensors often have errors that vary with temperature, and can also vary from day to day. Second, one can conduct either a static or GNSS-referenced navigation performance test, by running the inertial navigation equations on both the raw and corrected output of the IMU. This approach can tell you whether you have made the overall performance of the INS better or worse, and give you some idea by how much. However, this does not tell you about each individual type of error, only the overall effect. The final possibility is to conduct a realistic simulation of the calibration procedure. This is the approach taken here; the challenge being to make it realistic.

The calibration procedure detailed in this chapter aims to estimate biases, scale factor and cross-coupling errors for both the accelerometers and gyroscopes in the IMU and the gyroscopes' g-dependent bias (sensitivity to specific force). Also included in the simulation but not estimated by the calibration algorithm are sensor noise, and non-linearity. These are included because the calibration procedure assumes that their effect is negligible, and so the inclusion of them both will lead to a more realistic prediction of the quality of the calibration procedure's error estimates.

We work under the assumption, which is reasonable for the current generation of consumer grade IMUs, that both the accelerometers and gyroscopes are on the same chip, this means that the misalignments between both sets of sensitive axes are covered by their specifications and no additional error is added in the mounting.

4.2 The principle of the calibration procedure

The calibration procedure studied in this Chapter is chosen to not require the use of any equipment other than a flat surface (e.g. a table) and the IMU's housing, which is assumed to be a cuboid box. This allows the end-user to conduct the procedure, which would dramatically reduce the unit cost of the IMU systems relative to using a procedure that would need to be conducted in a laboratory. However, this also imposes a limitation on the errors that can be observed.

Accelerometers measure specific force, that is acceleration from all forces except gravity. This means that the reaction force to the gravitational acceleration is measured when the accelerometers are static, because of this many of the errors of an accelerometer triad can be determined by static tests. Gyroscopes on the other hand measure angular rate, not attitude, so only their biases can be determined by static tests, any attempt at determining scale factor or cross coupling errors requires movement. As such, the procedure tested in this chapter will include both static and dynamic parts.

The idea of calibrating (particularly) accelerometers by enclosing them in a cubic box and taking readings with the box resting on each of its sides is not new. A procedure using this idea has been suggested many times, such as for example, in [177] as a "six-position test", and an arguably more advanced variant using a icosahedron shaped box has been suggested in [155][1]. However, the accuracy of the estimates derived from this kind of procedure and how it depends on the various calibration inputs have not been assessed before this thesis.

In order to determine any calibration of the gyroscopes other than simple bias (zero-rate offset) we need to introduce movement. A common calibration method for low- to mid-cost gyroscopes is to use rate tables and/or a much higher performance gyroscope and then compare its output of that of the high-performance gyroscope [4]. However when one considers calibration methods which might be suitable for user-conducted calibration neither of these techniques is suitable, for reasons of cost. So the technique for calibrating the gyroscopes' alignment matrix, which we use in this chapter, is a version of a zero-attitude (angular position) update (see Section 4.5.2). This meets our requirement for not requiring any expensive equipment. However, it has the disadvantage is that its accuracy will depend on how well the procedure is carried out by the user.

In this chapter, the accuracy of the estimates from both the static and dynamic procedures are assessed through a simulation. There are two versions of both the static and dynamic calibration procedure assessed: a simple version which is easy to implement (A & B), and a more advanced version which is expected to produce better estimates (C & D). Manoeuvre set A represents the frequently used six-position static procedure, as was discussed in Section 2.5.1.3. Set C represents a new extension to this procedure which takes 4 static readings on each face (24 in total). The dynamic manoeuvres (sets B and D) represent an approach that has not been assessed previously in the literature where rotate-and-replace manoeuvres about the principle axes are used to calibrate the gyroscope by a zero attitude update. Set B represents the minimum 3 rotations, and set D extends this to six rotations and uses the box's faces as a guide for more consistent rotation axes.

In Figure 4.1 the naming convention for the sides of this box or calibration cube is presented. Each face is named after the axis which is normal to its surface, and whether the axis is directed into (-) or out of (+) the box. This face-naming convention is used to name the various static orientations for the box. These are named after the uppermost face, so Z-up (ZU) has the +Z face uppermost and Y-down (YD) has the -Y face uppermost; these are detailed further in Section 4.3.3.

[1] In this paper they state that a similar technique could be extended to any *platonic* solid, when in fact it could be extended to *any* convex-sided box, whether that extension would be useful is doubtful.

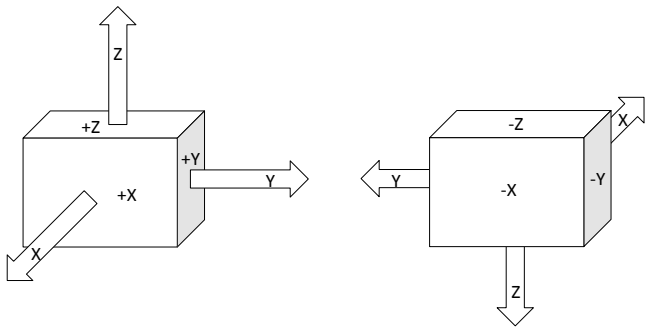


Figure 4.1: The convention used to describe the different faces of the box. The axes shown are those of the box frame (defined in Section 4.3.2)

Using these conventions the basic series of manoeuvres used for the calibration procedure simulated are shown in Figures 4.2, 4.3, 4.4 and 4.5.

The calibration algorithm, which will be described in Section 4.5 assumes that calibration procedure has been carried out precisely as expected, simply because it can have no information about how well or badly it has been performed. This is because departures from the expected measurements due to manoeuvring errors cannot be distinguished from the sensor errors that the algorithm is estimating.

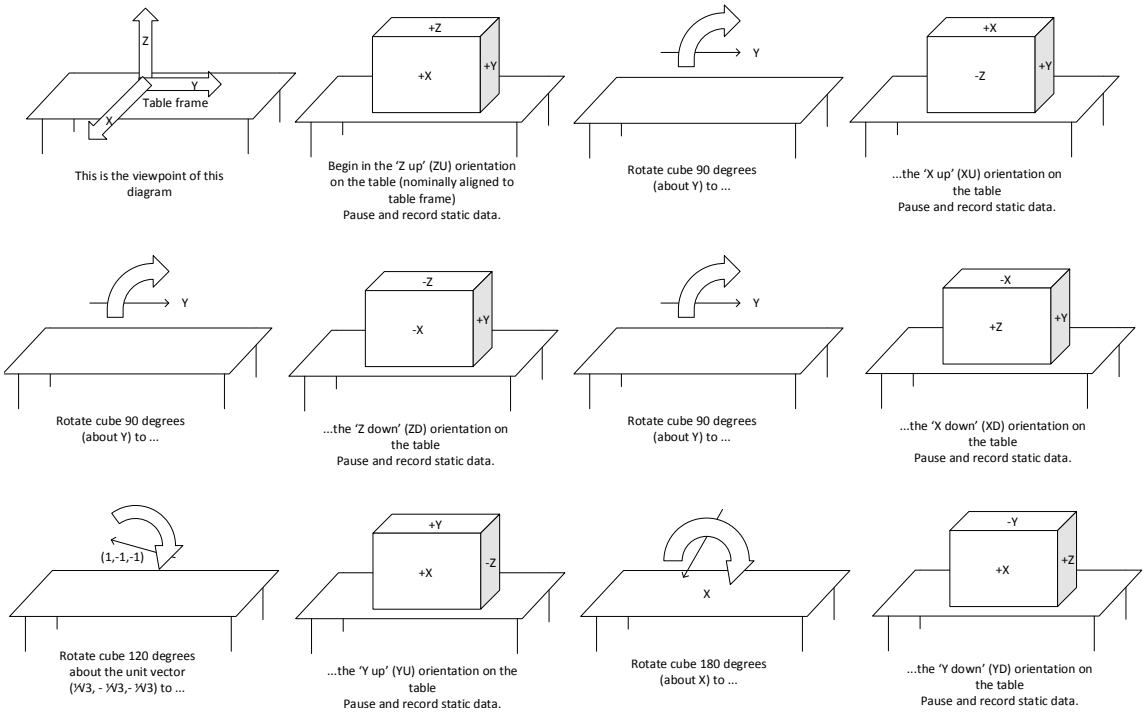


Figure 4.2: Manoeuvre set A of the user-conducted calibration procedure.

In order to understand the next sections a brief summary of the physical movements simulated for the calibration procedure follows. There are 4 sets of manoeuvres: a simple static procedure (set A), a simple dynamic procedure (set B), a more complex static procedure (set C) and a more complex dynamic procedure (set D). One of the static procedures and one of the dynamic procedures are needed to produce an complete set of calibration parameter estimates.

In set A, which is the existing method and shown in Figure 4.2, the box begins in the 'Z-up'

orientation, remains static for a specified time which is one of the inputs (up to 45 seconds), then is rotated onto the next face of the cube, then remains static, this is repeated until all 6 faces have had static data recorded on them.

Then it returns to the 'Z-up' face for the start of set B (Figure 4.3) where the cube is lifted into the air above the table where it is rotated about (as close to) the X-axis for (close to) 360 degrees over a specified time period, then replaced on the table at (as close as possible to) the same position and orientation. This is then repeated with the Y-axis and then the Z-axis.

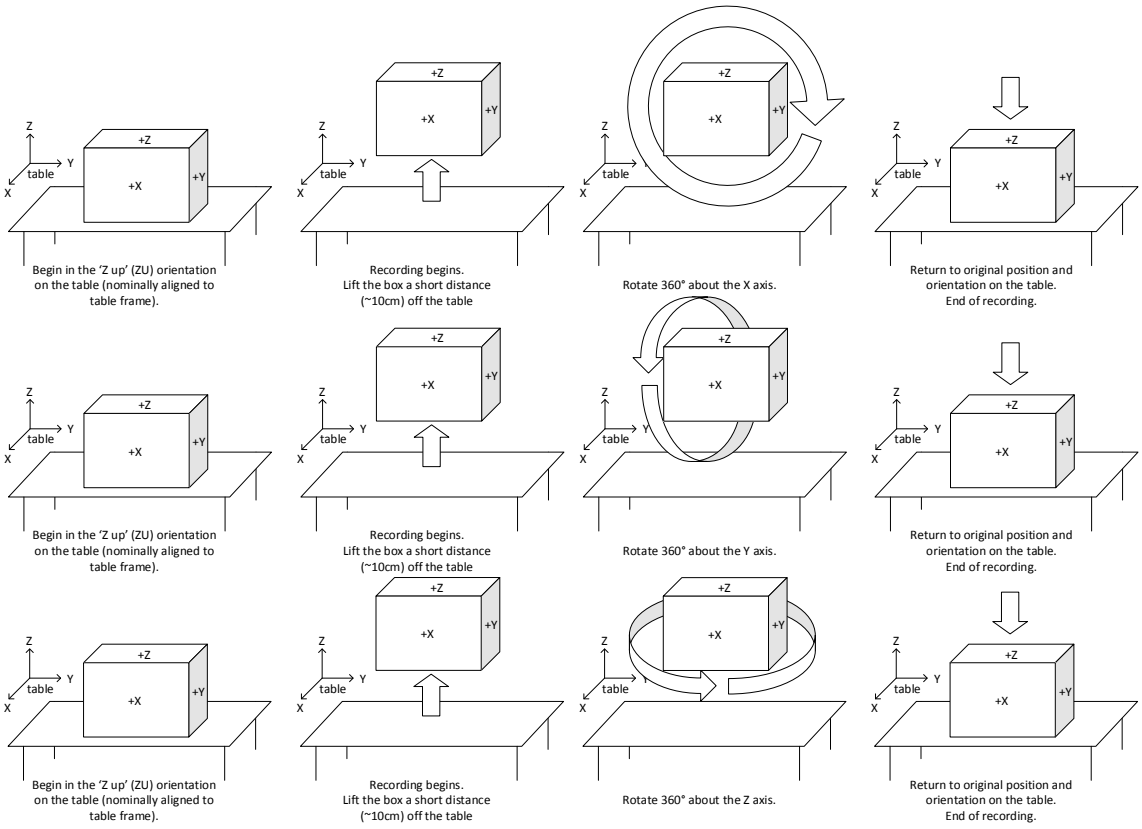


Figure 4.3: Manoeuvre set B of the user-conducted calibration procedure.

Set C is similar to set A but on each face there are four separate static readings taken on each face at 90 degrees heading difference (Figure 4.4).

In set D, the cube is positioned on each of its faces in turn and while on the surface of the table it is rotated through 360 degrees (Figure 4.5).

In set A and C the static data is the only part of the recorded data used, all the data recorded while moving between positions is ignored. The basic principle behind this calibration is that the specific force from the reaction to gravity has a magnitude of g ($\approx 9.81\,\mathrm{ms}^{-2}$) and is normal (and outwards) from the face of the cube which is uppermost. The mean of the measurements is calculated on each face to reduce the effect of noise. Then in the sum of two opposite faces the specific force should cancel out, leaving only (double) the bias, and the difference between two faces will cancel the bias, leaving, e.g. $\begin{bmatrix}0 & 0 & 2g\end{bmatrix}^T$, multiplied by particular components of either the $\mathbf{A_a}$ or $\mathbf{G_g}$ matrices (see Section 2.2.5) for

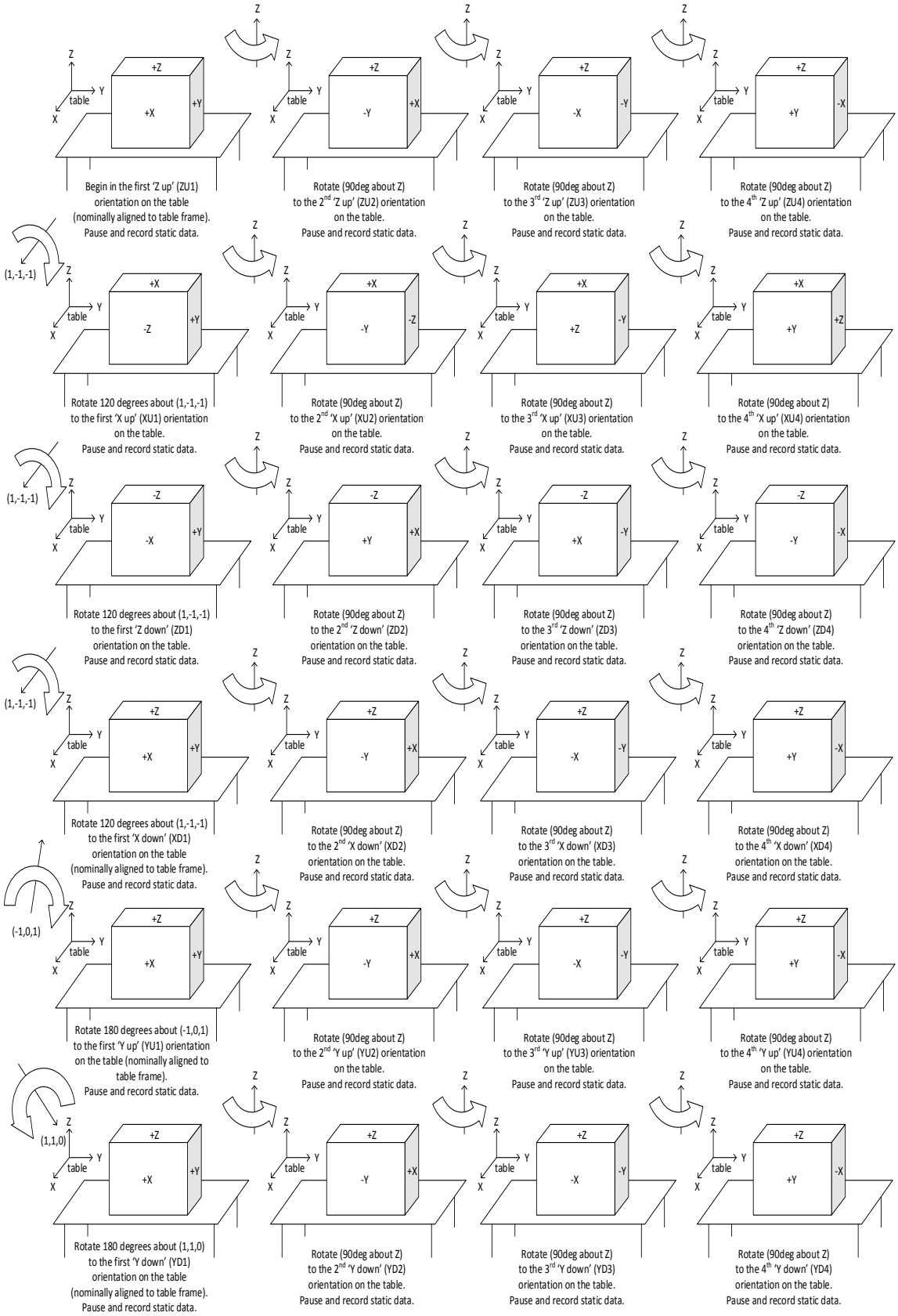


Figure 4.4: Manoeuvre set C of the user-conducted calibration procedure.

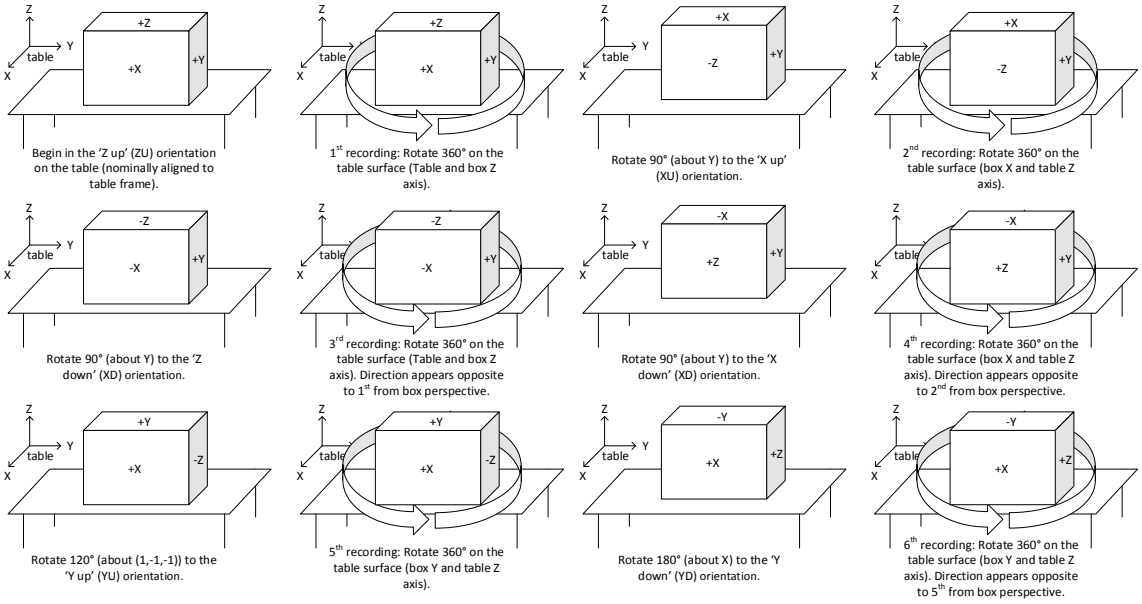


Figure 4.5: Manoeuvre set D of the user-conducted calibration procedure.

the accelerometer and gyroscope respectively, so all 3 pairs enable the whole matrix to be determined. The procedure is sensitive to non-level tables and non-cuboid housing, but the degree of this sensitivity depends on the algorithms and manoeuvres used.

The principle of the dynamic calibration (set B and D) is that each rotation is 360° about a certain axis. Then if the measurements are summed or integrated and the cube has apparently not rotated exactly 360° the amount of apparent angular movement, can be used to determine (one column of) $\mathbf{A_g}$. The improvement that comes from set D is that the rotation axis is typically closer to the intended axis because the cube's faces are used as guides and the repetition of the rotation reduces (by averaging between 2 measured rotations) the effect of integrated noise and procedural errors. The former being more important.

In a final procedure only either set A or C (static) and either set B or D (dynamic) would be needed. However, in this chapter both are simulated for comparison.

It should be noted that with even a very poor initial IMU calibration, if the user were to make a large mistake in the calibration procedure, for example placing the cube on the wrong face, rotating about the wrong axis, or getting the order of manoeuvres wrong, this could easily be flagged up by a simple error checking algorithm. This would lead to either the user being prompted to repeat the calibration or, perhaps in the case of the incorrect order of manoeuvres, the algorithm adapting. As such this kind of human error will not be considered. Additionally there are well established means of stationarity detection [178–180], which detect when the IMU is static, so we assume that identifying the start and finish times of static periods, and the beginning and ending of manoeuvres will not be difficult, and we assume that this is done correctly.

4.3 Reference frames referred to in this chapter

There are a number of coordinate reference frames which are used in this chapter to calculate the position and alignment of the IMU's sensors. Primarily these reference frames deal with generating the test-bed's trajectory during the Monte Carlo simulations of the calibration correctly, although they are also used to describe the effect of various errors on the calibration algorithm. The main concern is attitude errors because they are the most important for all the static readings generated, because they cause a change in the direction of the reaction to gravity. The combination of all the reference frames will be shown in Section 4.3.8. The matrix '$\mathbf{C}_{\mathbf{x}}^{\mathbf{y}}$' represents the rotation from frame x to frame y.

In this thesis, yaw (ψ), pitch (θ) and roll (ϕ) (that order) Euler angles need to be transformed to the coordinate transformation matrices, in which all calculations are conducted, particularly during these frame transformations. There are several different conventions in used for this conversion, this thesis uses (after Equation 2.22 of [1])

$$\begin{aligned}\mathbf{C} &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{pmatrix} \begin{pmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{pmatrix} \begin{pmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} \cos(\theta)\cos(\psi) & \cos(\theta)\sin(\psi) & -\sin(\theta) \\ -\cos(\phi)\sin(\psi)+\sin(\phi)\sin(\theta)\cos(\psi) & \cos(\phi)\cos(\psi)+\sin(\phi)\sin(\theta)\sin(\psi) & \sin(\phi)\cos(\theta) \\ \sin(\phi)\sin(\psi)+\cos(\phi)\sin(\theta)\cos(\psi) & -\sin(\phi)\cos(\psi)+\cos(\phi)\sin(\theta)\sin(\psi) & \cos(\phi)\cos(\theta) \end{pmatrix} \end{aligned} \quad (4.1)$$

4.3.1 IMU or Sensor frame

The sensor frame describes the orientation of the IMU's package outline. This is, in the case of the MEMS single-chip IMU sensors we are considering in this chapter, the orientation that the manufacturer's specifications are defined relative to. For example, manufacturers' define the sensors' alignment (cross-coupling) errors relative to this package outline. This frame is denoted by 's' and shown in Figure 4.6.

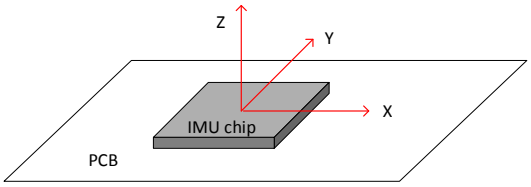


Figure 4.6: The IMU or sensor frame.

4.3.2 Body or box frame

The box frame is the frame describing the nominal position of the box in which the IMU is placed. As the box is not a perfect cube (see Section 4.3.4) the box frame is orientated to the box-average frame. That is, the orthogonal frame where a cube defined by it best fits the calibration cube.

The angular offset between the sensor, s, and box-average, b, frames is described by the coordinate transformation matrix $\mathbf{C_s^b}$, and is generated by the various IMU mounting errors (see Section 4.6.1). This is shown in Figure 4.7. Note that this matrix will contain any alignments that intentionally differ from that of the axes of the box, such as those used for array techniques in Chapter 6, in these situations it could be considered an array-average frame.

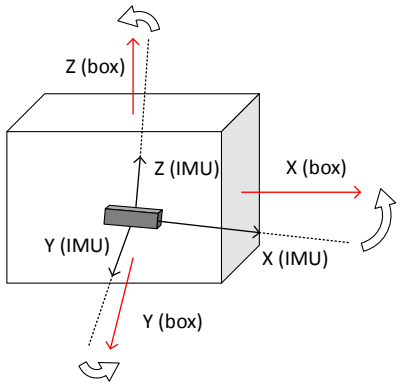


Figure 4.7: The relationship between the IMU or sensor frame and the box frame described by $\mathbf{C_s^b}$

It is important to note that while the specific force and angular rate are measured in the sensor frame, the user actually cares about the motion of the box frame, that is any intentional or unintentional misalignments between the IMU and its box should be removed. Additionally, the calibration must take place relative to an orthogonal frame, there is no way to separate $\mathbf{C_s^b}$ from the sensors' cross-coupling errors as effectively we are looking at misalignments within the IMU package and of the package in the box. Thus the frame we aim to have our calibration relative to is this box-average frame, and this is what the results will be compared to (see Section 4.7).

As a result of this the true trajectory is in the box frame, and the transformation into the IMU frame ($\mathbf{C_s^b}$) is considered part of the IMU error model (as described in Section 4.4.1).

4.3.3 The nominal orientation frame ("which face is on top?")

The nominal orientation frame simplistically describes which face of the cube is resting on the table. There are six nominal orientations,

$$\mathbf{C_b^{ZU}} = \mathbf{I_3}, \quad \mathbf{C_b^{ZD}} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix}, \quad \mathbf{C_b^{YU}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{pmatrix},$$
$$\mathbf{C_b^{YD}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{pmatrix}, \quad \mathbf{C_b^{XU}} = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}, \quad \mathbf{C_b^{XD}} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix} \tag{4.2}$$

According to the part of the calibration manoeuvres that is being generated, the relevant nominal orientation is chosen. To refer to these nominal orientation frames as a general group "NO" is used in place of, for example, "XD" or "ZU".

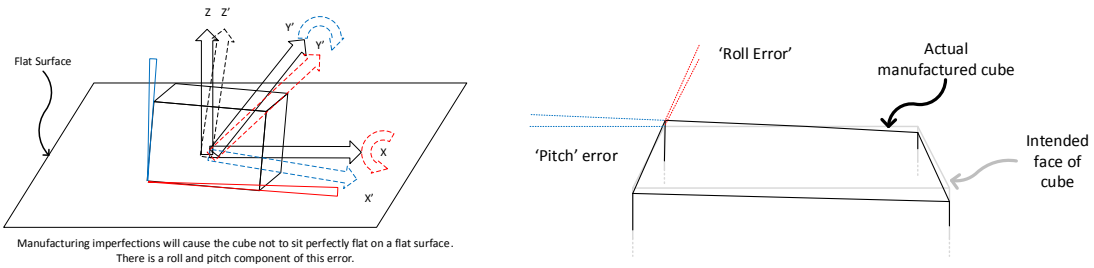


Figure 4.8: The face frame which describes the manufacturing imperfections of the calibration cube

The choice of these six nominal orientations is not arbitrary, they are chosen to represent the easiest way to conducting the set A static procedure. However, they are certainly not an optimal choice of 6 orientations. There are the 24 that could have chosen that are have only -1, 0 and 1 in their rotation matrix. This choice of orientations in fact slightly exacerbates the sensitivity to levelling error (see the results in Section 4.8.1.3), but the user will be more likely to conduct the easy option than the optimal one, so that is what is simulated.

Other static positions on a given face are simulated by using the heading transformation in Section 4.3.5.

4.3.4 Cube face frames

Any calibration cube that could really be constructed would not be a perfectly cubic. This imperfection is described by a rotation for each face of the frame which describes its angular error about the two perpendicular axes which span that face, these become a roll and a pitch error when the cube is on a flat surface. These errors are generated by the 'cube face error' parameter (see Section 4.6.1). This error is illustrated in Figure 4.8.

For each of the six faces there is a separate but fixed (for each Monte Carlo run) pitch and roll (relative to the table, when placed on that face) error namely, $\mathbf{C}_{\mathbf{ZU}}^{\mathbf{ZU}\text{face}}$, $\mathbf{C}_{\mathbf{ZD}}^{\mathbf{ZD}\text{face}}$, $\mathbf{C}_{\mathbf{YU}}^{\mathbf{YU}\text{face}}$, $\mathbf{C}_{\mathbf{YD}}^{\mathbf{YD}\text{face}}$, $\mathbf{C}_{\mathbf{XU}}^{\mathbf{XU}\text{face}}$ and $\mathbf{C}_{\mathbf{XD}}^{\mathbf{XD}\text{face}}$. Each of these matrices are derived from two random normally-distributed parameters, representing pitch and roll errors, the simulation assumes that the six of each of these are independently distributed, although whether this assumption would hold true for a real system would be dependent on the manufacturing methods used for the box.

It should be noted that this error depends on manufacturing imperfections of the face that is *in contact with the table*. For example, the $\mathbf{C}_{\mathbf{XU}}^{\mathbf{XU}\text{face}}$ depends on the manufacturing imperfections in the -X face.

Practically, because every time the cube is placed on a particular face the same face error will apply this transformation is combined with that from the previous section for all practical purposes. So,

$$\mathbf{C}_{\mathbf{b}}^{\mathbf{ZU}\text{face}} = \mathbf{C}_{\mathbf{ZU}}^{\mathbf{ZU}\text{face}}\mathbf{C}_{\mathbf{b}}^{\mathbf{ZU}}, \quad \mathbf{C}_{\mathbf{b}}^{\mathbf{ZD}\text{face}} = \mathbf{C}_{\mathbf{ZD}}^{\mathbf{ZD}\text{face}}\mathbf{C}_{\mathbf{b}}^{\mathbf{ZD}}, \quad \ldots \quad \text{etc.} \tag{4.3}$$

are used because they are fixed for each run.

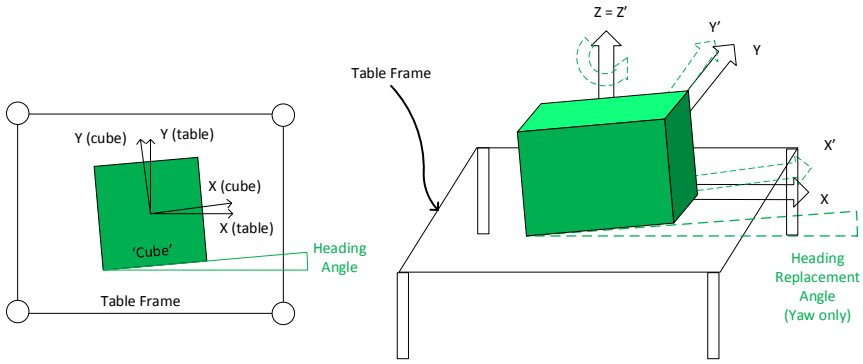


Figure 4.9: The relationship between the table and replacement heading frames ($\mathbf{C}^{\mathbf{t}}_{\mathbf{NO}\text{face}}$).

4.3.5 Heading-on-table frame

This frame describes the heading of the box with respect to the table ($\mathbf{C}^{\mathbf{t}}_{\mathbf{NO}\text{face}}$). This error is only in heading (i.e. about the table z-axis), and the angle between the table frame x-axis and the *NO*face frame x-axis is the 'heading angle'. This is shown in Figure 4.9.

This heading angle describes both deliberate and accidental yaw rotations. Deliberate rotations include the four different positions of manoeuvre set C.

Each time the cube is placed on the table there will be a different small error as the user places it at a slightly different heading angle than intended. This difference between the heading expected by the calibration algorithm and the actual heading is the replacement error. In the simulation this separately generated for each time the cube is placed on the table, as opposed to once per Monte Carlo run for the face errors in Section 4.3.4.

Note that, this frame needs to be 'after' the cube errors so the cube errors will rotate around fixed to the 'nominal orientation', rather than fixed to the table's axes.

4.3.6 Table frame

The table frame (t) describes how the table upon which the calibration procedure takes place differs from level (perpendicular to the gravity vector), it has the same origin as the local tangent frame (Section 4.3.7) but differs from it by two (roll and pitch) table angles, α and β, respectively. These are fixed for a particular run of a Monte-Carlo simulation. This relationship is shown in Figure 4.10

The table frame is almost completely aligned to East (X) and North (Y), differing only by the table angles from which the table deviates from level. As the table is being considered as a plane and the position of its edges is irrelevant to the calibration the simulation of a yaw error is unnecessary. This does not mean that the table need be aligned to East and North, as its edges are not used, rather the frame is defined to be (nearly) co-incident.

Additionally, any real table is unlikely to be perfectly planar. However, if one replaces the box very close (< 2mm) to the same place on the table this effect will be small, and dwarfed by other errors in the procedure, as such this is not considered in this simulation.

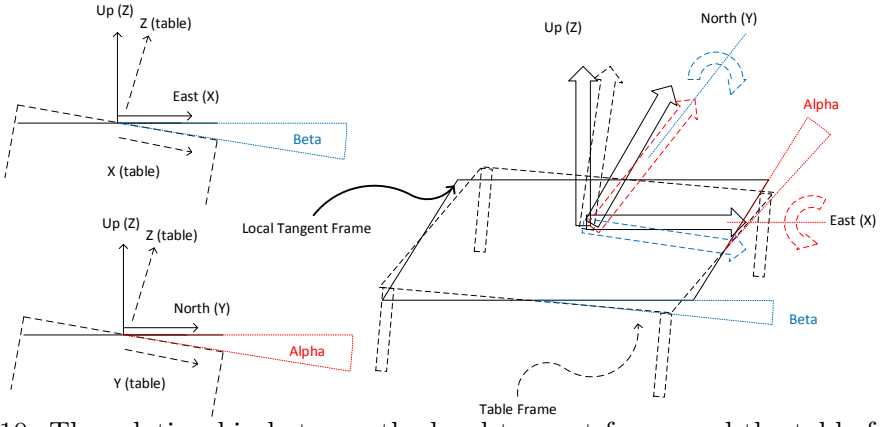


Figure 4.10: The relationship between the local tangent frame and the table frame ($\mathbf{C}_{\mathbf{t}}^{l}$).

4.3.7 Local tangent frame

The local tangent frame is the reference frame in which the test-bed's motion is considered. This frame, denoted by l, has its origin fixed, relative to the surface of the Earth, at a point on the surface of the table. The X-direction is towards the local East, the Y-direction to the local North and the Z-direction in the opposite direction to that of the local gravity vector (i.e. 'up').

Note that, the origin of this local tangent frame is at the latitude and longitude of UCL's main campus in London, although this makes only a very marginal difference because the rotation speed of the Earth is slow in comparison to size the gyroscope errors we are dealing with. Despite this, the rotation of the local tangent frame in inertial space is still simulated correctly.

The calibration procedure does not take into account the rotation of the local calibration frame because it makes only a very small difference and requires that the heading be known.

4.3.8 Relationship between the frames

The following equation describes the chain of rotations needed to describe the attitude of the IMU's sensors in the local tangent frame when the cube is positioned on the table

$$\mathbf{C}_{\mathbf{s}}^{l} = \mathbf{C}_{\mathbf{t}}^{l}\mathbf{C}_{\mathbf{NO}\text{face}}^{\mathbf{t}}\mathbf{C}_{\mathbf{NO}}^{\mathbf{NO}\text{face}}\mathbf{C}_{\mathbf{b}}^{\mathbf{NO}}\mathbf{C}_{\mathbf{s}}^{\mathbf{b}} \tag{4.4}$$

The matrix '$\mathbf{C}_{\mathbf{x}}^{\mathbf{y}}$' represents the rotation from frame x to frame y. 'l' is the local navigation frame. 't' is the table frame. 'NOface' is the frame describing the true misalignment of the specific face the box is resting upon. 'NO' is the nominal (box-average) orientation of the cube (ZU, XD etc.). 'b' is the box frame. 's' is the sensor frame. These frames were explained in Sections 4.3.1 to 4.3.7.

All these frames are used in the simulation of the calibration procedure. However, only some of them are known to the calibration algorithm. The algorithm assumes that the box is perfectly cuboid, thus assumes $\mathbf{C}_{\mathbf{NO}}^{\mathbf{NO}\text{face}} = \mathbf{I_3}$. The procedure using manoeuvre set A (but not set C) assumes that the table is level, i.e. $\mathbf{C}_{\mathbf{t}}^{l} = \mathbf{I_3}$. For other frame transformations such as the 'heading on the table' ($\mathbf{C}_{\mathbf{NO}\text{face}}^{\mathbf{t}}$) and the 'IMUs mounting orientation inside the

box' ($\mathbf{C_s^b}$) the calibration algorithm assumes these have particular intended values. The simulation adds a slightly deviation from these intended values.

The calibration algorithm cannot distinguish the mis-mounting of the entire IMU in the box ($\mathbf{C_s^b}$), from the mis-mounting of individual sensors within the IMU (part of $\mathbf{A_a}$ or $\mathbf{A_g}$). However, this does not matter because the corrected IMU output *relative to the mounting* is what the user requires for navigation so the calibration includes transforming the IMU measurements from the sensor frame to the box frame. This will be discussed further in Section 4.4.

4.4 The IMU error model used for in this chapter

This section describes the IMU error models being used in the simulations in this chapter. The purpose of the IMU error model is to convert the true angular rate and specific force, denoted by column vectors $\boldsymbol{f_{ib}^b}$ and $\boldsymbol{\omega_{ib}^b}$, that is those that would be measured by a perfect IMU, into measured angular rate and measured specific force, $\widetilde{\boldsymbol{f_{is}^s}}$ and $\widetilde{\boldsymbol{\omega_{is}^s}}$, as would be measured by a realistic IMU with measurement and mounting errors. This is detailed in Section 4.4.1. The calibration procedure then aims to correct these measured quantities to create corrected or compensated angular rate and specific force, $\widehat{\boldsymbol{f_{ib}^b}}$ and $\widehat{\boldsymbol{\omega_{ib}^b}}$, which have had the estimates of the systematic errors removed in the manner presented in Section 4.4.2.

It should be noted that this is a more complex model than that in Chapter 3, as simulation of sensor non-linearity is added for this chapter. The reason that a higher order model is used is because, in this chapter, more error sources are being calibrated. It is the author's opinion that it is important to include errors one-order higher than are being estimated in the error model. Thus, as in this chapter sensor scale factor and alignment are estimated (first-order), as such the IMU error model (Equations 4.5 and 4.6), includes quadratic non-linearity terms (second order) which is not included in the equivalent Equations 3.4 and 3.5 in Chapter 3. This allows the higher order errors to affect the estimates, and improves the realism of the simulation.

4.4.1 The IMU error model

A simplified version of the IMU model used in this chapter is

$$\widetilde{\boldsymbol{f_{is}^s}} = \boldsymbol{b_a} + \mathbf{A_a^s}\boldsymbol{f_{is}^s} + \boldsymbol{w_a} + n(\boldsymbol{l_a}, \boldsymbol{f_{is}^s}) \tag{4.5}$$

$$\widetilde{\boldsymbol{\omega_{is}^s}} = \boldsymbol{b_g} + \mathbf{A_g}\boldsymbol{\omega_{is}^s} + \mathbf{G_g}\boldsymbol{f_{is}^s} + \boldsymbol{w_g} + n(\boldsymbol{l_g}, \boldsymbol{\omega_{is}^s}) \tag{4.6}$$

in the special case that the IMU was strictly lined up with the box-frame. The full model used in this chapter, which including the mounting errors of the IMU within the box, is:

$$\widetilde{\boldsymbol{f_{is}^s}} = \boldsymbol{b_a} + \mathbf{A_a^s}(\mathbf{C_b^s}\boldsymbol{f_{ib}^b}) + \boldsymbol{w_a} + n(\boldsymbol{l_a}, (\mathbf{C_b^s}\boldsymbol{f_{ib}^b})) \tag{4.7}$$

$$\widetilde{\boldsymbol{\omega_{is}^s}} = \boldsymbol{b_g} + \mathbf{A_g^s}(\mathbf{C_b^s}\boldsymbol{\omega_{ib}^b}) + \mathbf{G_g^s}(\mathbf{C_b^s}\boldsymbol{f_{ib}^b}) + \boldsymbol{w_g} + n(\boldsymbol{l_g}, (\mathbf{C_b^s}\boldsymbol{\omega_{ib}^b})) \tag{4.8}$$

The function 'n' applies a different coefficient from the 3×2 coefficient matrix ($\boldsymbol{l}$) to the entrywise square of the 3×1 vector $\boldsymbol{v}$ depending on the sign of each entry of $\boldsymbol{v}$, where the

i$^{\text{th}}$ component is denoted by $v(i)$.

$$\boldsymbol{a} = n(\boldsymbol{l}, \boldsymbol{v}) \text{ where for } i = 1 \ldots 3,\ a(i) = \begin{cases} l(i,1) \times v(i)^2, & \text{when } v(i) \geq 0 \\ l(i,2) \times v(i)^2, & \text{otherwise} \end{cases} \tag{4.9}$$

There are a number of ways of interpreting the non-linearity as discussed in Section 2.2.6. Here a quadratic non-linearity, with separate positive and negative coefficients, is assumed on the basis that where both quadratic and cubic coefficients are given (e.g. [47]) the quadratic typically dominates, and the seperate coefficients to capture the fact that the response is not always symmetrical about zero (e.g. [56]). There is no continuity problem because when $x = 0$, $ax^2 = bx^2 \,\forall\, a, b$

The symbols in the above equations are defined as:
$\boldsymbol{f}$ is the specific force on each sensitive axis (3×1 column vector).
$\boldsymbol{b_a}$ is the vector of the accelerometer biases (3×1).
$\mathbf{A_a}$ is the matrix containing the accelerometer scale-factor and cross-coupling errors (3×3).
$\boldsymbol{l_a}$ is a 3×2 matrix containing the coefficients for the non-linearity model for the accelerometers, as can be seen in Equation 4.9, the assumption is that the non-linearity is quadratic in nature, there are two coefficients for each axis, a different (independent) one for positive and one for negative specific force.
$\boldsymbol{w_a}$ is the vector containing the (white) sensor noise on each accelerometer axis (3×1). It is generated independently for each epoch from the specified noise performance. Sensor noise being white is a modelling simplification.

$\boldsymbol{\omega}$ is the angular rate on each sensitive axis (3×1 column vector).
$\boldsymbol{b_g}$ is the vector of the gyroscope biases (3×1);
$\mathbf{A_g}$ is the alignment matrix containing the gyroscope scale-factor and cross-coupling errors (3×3).
$\mathbf{G_g}$ is the matrix containing the gyroscope g-dependent errors. This is the effect where applied specific force is erroneously interpreted as angular rate (3×3).
$\boldsymbol{l_g}$ is a 3×2 matrix containing the coefficients for the non-linearity model for the gyroscope, as can be seen in Equation 4.9, the assumption is that the non-linearity is quadratic in nature, there are two coefficients for each axis, a different (independent) one for positive and one for negative angular rate.
$\boldsymbol{w_g}$ is the vector containing the (white) sensor noise on each gyroscope axis (3×1). It is generated independently for each epoch from the specified noise performance. Sensor noise being white is a modelling simplification.
$\mathbf{C_b^s}$ refers to the coordinate transformation matrix from the *nominal* box frame to the IMU / sensor frame. It thus represents the mis-mounting of the IMU inside the perfect box. It does *not* contain the errors from what the box should be to the actual position of its sides. Note that because this is a rotation matrix

$$\mathbf{C_s^b} = (\mathbf{C_b^s})^T.$$

The superscripts, for $\mathbf{G_g}$, $\mathbf{A_a}$ and $\mathbf{A_g}$, refer to the frame to which the error applies, s for the sensor frame and b for the box frame.
Entries with the diacritic '$\tilde{}$' are the measured values. Entries with the diacritic '$\hat{}$' are

the estimated values (or compensated output for $\widehat{\boldsymbol{f}}$ and $\widehat{\boldsymbol{\omega}}$). Entries with neither are the true values. So for example $\widehat{\boldsymbol{b_a}}$ means 'estimated accelerometer bias' and $\mathbf{G_g^s}$ means 'true gyroscope g-dependent matrix with respect to the sensor frame'.

It should be noted that $\boldsymbol{f_{ib}^b}$ and $\widehat{\boldsymbol{f_{ib}^b}}$ are in the nominal box frame, whereas the measured outputs $\widetilde{\boldsymbol{f_{is}^s}}$ are in the IMU/sensor frame (similarly for $\boldsymbol{\omega_{ib}^b}$, $\widehat{\boldsymbol{\omega_{ib}^b}}$ and $\widetilde{\boldsymbol{\omega_{is}^s}}$). This complicates the estimates which are needed to compensate for the sensor errors, as shall be presented in Section 4.4.2.

4.4.2 The error compensation

The aim of the procedures tested in this chapter is to estimate systematic errors, giving $\widehat{\boldsymbol{b_a}}$, $\widehat{\boldsymbol{b_g}}$, $\widehat{\mathbf{A_a^b}}$, $\widehat{\mathbf{A_g^b}}$ and $\widehat{\mathbf{G_g^b}}$, in order that the effect of those errors can be removed from the measured IMU readings, $\widetilde{\boldsymbol{f_{is}^s}}$ and $\widetilde{\boldsymbol{\omega_{is}^s}}$. Comparing Equations 4.5 and 4.6 with Equations 4.7 and 4.8 one can see that

$$\mathbf{A_a^b} = \mathbf{A_a^s}\mathbf{C_b^s}, \quad \mathbf{A_g^b} = \mathbf{A_g^s}\mathbf{C_b^s} \quad \text{and} \quad \mathbf{G_g^b} = \mathbf{G_g^s}\mathbf{C_b^s}. \tag{4.10}$$

The point of calibrating these errors is so their effect can be removed from the IMU readings to create a compensated The method by which these compensated or calibrated IMU readings are calculated are as follows.

$$\widehat{\boldsymbol{f_{ib}^b}} = \widehat{\mathbf{A_a^b}}^{-1}(\widetilde{\boldsymbol{f_{is}^s}} - \widehat{\boldsymbol{b_a}}) \tag{4.11}$$

$$\widehat{\boldsymbol{\omega_{ib}^b}} = \widehat{\mathbf{A_g^b}}^{-1}(\widetilde{\boldsymbol{\omega_{is}^s}} - \widehat{\boldsymbol{b_g}} - \widehat{\mathbf{G_g^b}}\widehat{\boldsymbol{f_{ib}^b}}) \tag{4.12}$$

These should remove the effects of these systematic errors, leaving only the true measurement, sensor noise and non-linearity. The main assumptions are that the systematic errors have been estimated correctly. The rest of this section will show that this is the case.

First, substituting Equations 4.7 and 4.8 into these

$$\widehat{\boldsymbol{f_{ib}^b}} = \widehat{\mathbf{A_a^b}}^{-1}((\boldsymbol{b_a} + \mathbf{A_a^s}(\mathbf{C_b^s}\boldsymbol{f_{ib}^b}) + \boldsymbol{w_a} + n(\boldsymbol{l_a}, (\mathbf{C_b^s}\boldsymbol{f_{ib}^b}))) - \widehat{\boldsymbol{b_a}}) \tag{4.13}$$

$$\widehat{\boldsymbol{\omega_{ib}^b}} = \widehat{\mathbf{A_g^b}}^{-1}((\boldsymbol{b_g} + \mathbf{A_g^s}(\mathbf{C_b^s}\boldsymbol{\omega_{ib}^b}) + \mathbf{G_g^s}(\mathbf{C_b^s}\boldsymbol{f_{ib}^b}) + \boldsymbol{w_g} + n(\boldsymbol{l_g}, (\mathbf{C_b^s}\boldsymbol{\omega_{ib}^b}))) - \widehat{\boldsymbol{b_g}} - \widehat{\mathbf{G_g^b}}\widehat{\boldsymbol{f_{ib}^b}}) \tag{4.14}$$

Re-arranging these equations makes

$$\widehat{\boldsymbol{f_{ib}^b}} = \widehat{\mathbf{A_a^b}}^{-1}\left[\boldsymbol{b_a} - \widehat{\boldsymbol{b_a}}\right] + \widehat{\mathbf{A_a^b}}^{-1}\left[\mathbf{A_a^s}(\mathbf{C_b^s}\boldsymbol{f_{ib}^b})\right] + \widehat{\mathbf{A_a^b}}^{-1}[\boldsymbol{w_a}] + \widehat{\mathbf{A_a^b}}^{-1}\left[n(\boldsymbol{l_a}, (\mathbf{C_b^s}\boldsymbol{f_{ib}^b}))\right] \tag{4.15}$$

$$\begin{aligned}\widehat{\boldsymbol{\omega_{ib}^b}} = \widehat{\mathbf{A_g^b}}^{-1}\left[\boldsymbol{b_g} - \widehat{\boldsymbol{b_g}}\right] + \widehat{\mathbf{A_g^b}}^{-1}\left[\mathbf{A_g^s}(\mathbf{C_b^s}\boldsymbol{\omega_{ib}^b})\right] + \widehat{\mathbf{A_g^b}}^{-1}\left[\mathbf{G_g^s}(\mathbf{C_b^s}\boldsymbol{f_{ib}^b}) - \widehat{\mathbf{G_g^b}}\widehat{\boldsymbol{f_{ib}^b}}\right] \\ + \widehat{\mathbf{A_g^b}}^{-1}[\boldsymbol{w_g}] + \widehat{\mathbf{A_g^b}}^{-1}\left[n(\boldsymbol{l_g}, (\mathbf{C_b^s}\boldsymbol{\omega_{ib}^b}))\right]\end{aligned} \tag{4.16}$$

Consider the specific force first. If we allow that

$$\widehat{\boldsymbol{b_a}} \approx \boldsymbol{b_a} \tag{4.17}$$

Then Equation 4.15 becomes

$$\widehat{\boldsymbol{f}_{ib}^{b}} = \widehat{\mathbf{A}_{\mathbf{a}}^{\mathbf{b}}}^{-1}\Big[\cancelto{\approx 0}{\boldsymbol{b}_{\boldsymbol{a}} - \widehat{\boldsymbol{b}_{\boldsymbol{a}}}}\Big] + \widehat{\mathbf{A}_{\mathbf{a}}^{\mathbf{b}}}^{-1}\Big[\mathbf{A}_{\mathbf{a}}^{\mathbf{s}}(\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}\boldsymbol{f}_{ib}^{b})\Big] + \widehat{\mathbf{A}_{\mathbf{a}}^{\mathbf{b}}}^{-1}[\boldsymbol{w}_{\boldsymbol{a}}] + \widehat{\mathbf{A}_{\mathbf{a}}^{\mathbf{b}}}^{-1}\Big[n(\boldsymbol{l}_{\boldsymbol{a}}, (\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}\boldsymbol{f}_{ib}^{b}))\Big] \tag{4.18}$$

$$\approx \underbrace{\widehat{\mathbf{A}_{\mathbf{a}}^{\mathbf{b}}}^{-1}\Big[\mathbf{A}_{\mathbf{a}}^{\mathbf{s}}(\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}\boldsymbol{f}_{ib}^{b})\Big]}_{\text{part including } \mathbf{A_a}} + \underbrace{\widehat{\mathbf{A}_{\mathbf{a}}^{\mathbf{b}}}^{-1}[\boldsymbol{w}_{\boldsymbol{a}}]}_{\text{noise}} + \underbrace{\widehat{\mathbf{A}_{\mathbf{a}}^{\mathbf{b}}}^{-1}\Big[n(\boldsymbol{l}_{\boldsymbol{a}}, (\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}\boldsymbol{f}_{ib}^{b}))\Big]}_{\text{non-linearity}} \tag{4.19}$$

In order for the 'part including $\mathbf{A_a}$' to cancel, if it is assumed that

$$\widehat{\mathbf{A}_{\mathbf{a}}^{\mathbf{b}}} \approx \mathbf{A}_{\mathbf{a}}^{\mathbf{s}}\mathbf{C}_{\mathbf{b}}^{\mathbf{s}} \tag{4.20}$$

$$\implies \widehat{\mathbf{A}_{\mathbf{a}}^{\mathbf{b}}}\boldsymbol{f}_{ib}^{b} \approx \mathbf{A}_{\mathbf{a}}^{\mathbf{s}}\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}\boldsymbol{f}_{ib}^{b} \tag{4.21}$$

$$\implies \boldsymbol{f}_{ib}^{b} \approx \widehat{\mathbf{A}_{\mathbf{a}}^{\mathbf{b}}}^{-1}\Big[\mathbf{A}_{\mathbf{a}}^{\mathbf{s}}(\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}\boldsymbol{f}_{ib}^{b})\Big] \tag{4.22}$$

Thus Equation 4.18 becomes

$$\widehat{\boldsymbol{f}_{ib}^{b}} \approx \underbrace{\boldsymbol{f}_{ib}^{b}}_{\text{true specific force}} + \underbrace{\widehat{\mathbf{A}_{\mathbf{a}}^{\mathbf{b}}}^{-1}[\boldsymbol{w}_{\boldsymbol{a}}]}_{\text{noise}} + \underbrace{\widehat{\mathbf{A}_{\mathbf{a}}^{\mathbf{b}}}^{-1}\Big[n(\boldsymbol{l}_{\boldsymbol{a}}, (\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}\boldsymbol{f}_{ib}^{b}))\Big]}_{\text{non-linearity}} \tag{4.23}$$

Now consider the angular rate. If in a similar manner we assume that

$$\widehat{\boldsymbol{b}_{\boldsymbol{g}}} \approx \boldsymbol{b}_{\boldsymbol{g}} \tag{4.24}$$

and

$$\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}} \approx \mathbf{A}_{\mathbf{g}}^{\mathbf{s}}\mathbf{C}_{\mathbf{b}}^{\mathbf{s}} \tag{4.25}$$

$$\implies \widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}\boldsymbol{\omega}_{ib}^{b} \approx \mathbf{A}_{\mathbf{g}}^{\mathbf{s}}\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}\boldsymbol{\omega}_{ib}^{b} \tag{4.26}$$

$$\implies \boldsymbol{\omega}_{ib}^{b} \approx \widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}^{-1}\Big[\mathbf{A}_{\mathbf{g}}^{\mathbf{s}}(\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}\boldsymbol{\omega}_{ib}^{b})\Big] \tag{4.27}$$

In exactly the same way that Equation 4.15 could use the analogous assumptions to make Equation 4.23, these assumptions can be used to simplify Equation 4.16. Thus,

$$\widehat{\boldsymbol{\omega}_{ib}^{b}} = \widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}^{-1}\Big[\cancelto{\approx 0}{\boldsymbol{b}_{\boldsymbol{g}} - \widehat{\boldsymbol{b}_{\boldsymbol{g}}}}\Big] + \Big[\cancelto{\approx \mathbf{I_3}}{\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}^{-1}\mathbf{A}_{\mathbf{g}}^{\mathbf{s}}\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}}\boldsymbol{\omega}_{ib}^{b}\Big] + \widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}^{-1}\Big[\mathbf{G}_{\mathbf{g}}^{\mathbf{s}}(\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}\boldsymbol{f}_{ib}^{b}) - \widehat{\mathbf{G}_{\mathbf{g}}^{\mathbf{b}}}\widehat{\boldsymbol{f}_{ib}^{b}}\Big]$$
$$+ \widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}^{-1}[\boldsymbol{w}_{\boldsymbol{g}}] + \widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}^{-1}\Big[n(\boldsymbol{l}_{\boldsymbol{g}}, (\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}\boldsymbol{\omega}_{ib}^{b}))\Big] \tag{4.28}$$

That is,

$$\widehat{\boldsymbol{\omega}_{ib}^{b}} \approx \boldsymbol{\omega}_{ib}^{b} + \underbrace{\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}^{-1}\Big[\mathbf{G}_{\mathbf{g}}^{\mathbf{s}}(\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}\boldsymbol{f}_{ib}^{b}) - \widehat{\mathbf{G}_{\mathbf{g}}^{\mathbf{b}}}\widehat{\boldsymbol{f}_{ib}^{b}}\Big]}_{\text{part including } \mathbf{G_g}} + \widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}^{-1}[\boldsymbol{w}_{\boldsymbol{g}}] + \widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}^{-1}\Big[n(\boldsymbol{l}_{\boldsymbol{g}}, (\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}\boldsymbol{\omega}_{ib}^{b}))\Big] \tag{4.29}$$

However, in order to cancel the 'part including $\mathbf{G_g}$' we must make two assumptions. First, that

$$\widehat{\mathbf{G}_{\mathbf{g}}^{\mathbf{b}}} \approx \mathbf{G}_{\mathbf{g}}^{\mathbf{s}}\mathbf{C}_{\mathbf{b}}^{\mathbf{s}} \tag{4.30}$$

$$\implies 0 \approx \mathbf{G}_{\mathbf{g}}^{\mathbf{s}}\mathbf{C}_{\mathbf{b}}^{\mathbf{s}} - \widehat{\mathbf{G}_{\mathbf{g}}^{\mathbf{b}}} \tag{4.31}$$

$$\implies 0 \approx \mathbf{G}_{\mathbf{g}}^{\mathbf{s}}\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}\boldsymbol{f}_{ib}^{b} - \widehat{\mathbf{G}_{\mathbf{g}}^{\mathbf{b}}}\boldsymbol{f}_{ib}^{b} \tag{4.32}$$

If it is additionally assumed,

$$\widehat{\boldsymbol{f}_{\boldsymbol{ib}}^{\boldsymbol{b}}} \approx \boldsymbol{f}_{\boldsymbol{ib}}^{\boldsymbol{b}} \tag{4.33}$$

which is reasonable given Equation 4.23, then Equation 4.29 becomes

$$\widehat{\boldsymbol{\omega}_{\boldsymbol{ib}}^{\boldsymbol{b}}} \approx \boldsymbol{\omega}_{\boldsymbol{ib}}^{\boldsymbol{b}} + \widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}^{-1} \left[\cancel{\mathbf{G}_{\mathbf{g}}^{\mathbf{s}}(\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}\boldsymbol{f}_{\boldsymbol{ib}}^{\boldsymbol{b}}) - \widehat{\mathbf{G}_{\mathbf{g}}^{\mathbf{b}}}\widehat{\boldsymbol{f}_{\boldsymbol{ib}}^{\boldsymbol{b}}}} \approx 0 \right] + \widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}^{-1} [\boldsymbol{w}_{\boldsymbol{g}}] + \widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}^{-1} \left[n(\boldsymbol{l}_{\boldsymbol{g}}, (\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}\boldsymbol{\omega}_{\boldsymbol{ib}}^{\boldsymbol{b}})) \right] \tag{4.34}$$

which gives

$$\widehat{\boldsymbol{\omega}_{\boldsymbol{ib}}^{\boldsymbol{b}}} \approx \underbrace{\boldsymbol{\omega}_{\boldsymbol{ib}}^{\boldsymbol{b}}}_{\text{true angular rate}} + \underbrace{\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}^{-1} [\boldsymbol{w}_{\boldsymbol{g}}]}_{\text{noise}} + \underbrace{\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}^{-1} \left[n(\boldsymbol{l}_{\boldsymbol{g}}, (\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}\boldsymbol{\omega}_{\boldsymbol{ib}}^{\boldsymbol{b}})) \right]}_{\text{non-linearity}} . \tag{4.35}$$

4.5 Determination of sensor errors from calibration procedure output

This section presents the method by which the sensors' errors are estimated. As mentioned in Section 4.2, the accelerometers' errors can be calibrated using only static readings. However, the gyroscopes require motion so that their errors can be estimated. As such this section is split into two parts: the 'static' part (Section 4.5.1) and the dynamic part (Section 4.5.2).

The actual manoeuvres that characterise these parts are described in Section 4.2, where set A or C (Figures 4.2 and 4.4) comprise the static and set B or D (Figures 4.3 and 4.5) the dynamic. Note that the dynamic calibration depends on the results of the static part, and inaccuracies from the static part of the procedure will increase the errors in the dynamic part.

4.5.1 The static part of the calibration procedure

The static part of the calibration procedure is based on taking measurements on all six faces of a cuboid box or calibration cube containing the IMU, and averaging the recorded data over a static period of a few seconds to reduce the effect of the sensors' random errors. However, no real physical box would ever be perfectly cuboid, and whatever surface it is placed on might not be exactly perpendicular to the local gravity vector (not level). Here we aim to test how great an effect these and other errors will have on the resulting calibration.

The readings on each face are simply the mean over time of the measurements from each sensor in each of the two triads. In the case of the more advanced procedure (set C) measurements are taken at several headings and then the reading of each face is the average of the reading at each of the different headings.

Procedures which use no equipment other than the box and a flat surface on which to rest it can still vary in their complexity. Two variations are examined: both a simple (set A) and more complex (set C) physical movements. These procedures were presented in more detail in Section 4.2. The simpler version is a single static reading on each side of the box without regarding the heading of the box on the table (set A), the 6 heading orientations are intentionally not optimally chosen, see Sections 4.3.3 and 4.8.1.3. The more complex

version also relies on some kind of straight edge or corner on the table to allow the box to be lined up (for more details on this see Section 4.5.2). In this version, four static readings are taken on each face of the box at headings differing by 90 degrees. This allows the table's deviation from level to be measured and its effect to be (at least partially) compensated for, as follows.

4.5.1.1 Table angle estimation

Calculations using manoeuvre set A assume that the table is level, but if manoeuvre set C is used instead the 'angles by which the table deviates from level', henceforth table angles, can be estimated.

The basic assumption is that for a level table the true reading of the specific force should be g (≈ 9.8 ms^{-2}) in the vertical direction and zero in the other directions, that is,

$$\boldsymbol{f}^{\boldsymbol{l}}_{\boldsymbol{il}} = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{4.36}$$

and that the angular rate should be zero. When the table is not level in the table frame gravity does not apply directly on the table-frame z-axis so,

$$\boldsymbol{f}^{\boldsymbol{t}}_{\boldsymbol{it}} = \begin{bmatrix} \sin(\alpha)\cos(\beta)g \\ \cos(\alpha)\sin(\beta)g \\ \cos(\alpha)\cos(\beta)g \end{bmatrix} \tag{4.37}$$

where α and β are the two table angles, see Figure 4.10.

Consider two sets of static accelerometer readings, $\boldsymbol{f}^{\boldsymbol{h}\,+}_{\boldsymbol{ih}}$, which is aligned to the table frame and $\boldsymbol{f}^{\boldsymbol{h}\,-}_{\boldsymbol{ih}}$ which is rotated through a heading of 180 degrees on the table. Thus their difference is,

$$\begin{aligned} \boldsymbol{f}^{\boldsymbol{h}\,+}_{\boldsymbol{ih}} - \boldsymbol{f}^{\boldsymbol{h}\,-}_{\boldsymbol{ih}} \approx \boldsymbol{f}^{\boldsymbol{t}}_{\boldsymbol{it}} - \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{f}^{\boldsymbol{t}}_{\boldsymbol{it}} &= \begin{bmatrix} \sin(\alpha)\cos(\beta)g \\ \cos(\alpha)\sin(\beta)g \\ \cos(\alpha)\cos(\beta)g \end{bmatrix} - \begin{bmatrix} -\sin(\alpha)\cos(\beta)g \\ -\cos(\alpha)\sin(\beta)g \\ \cos(\alpha)\cos(\beta)g \end{bmatrix} \\ &= \begin{bmatrix} 2\sin(\alpha)\cos(\beta)g \\ 2\cos(\alpha)\sin(\beta)g \\ 0 \end{bmatrix}. \end{aligned} \tag{4.38}$$

Take the first two components (x & y) of this, namely

$$\left(\boldsymbol{f}^{\boldsymbol{h}\,+}_{\boldsymbol{ih}} - \boldsymbol{f}^{\boldsymbol{h}\,-}_{\boldsymbol{ih}}\right)_x = 2\sin(\alpha)\cos(\beta)g \quad \text{and} \quad \left(\boldsymbol{f}^{\boldsymbol{h}\,+}_{\boldsymbol{ih}} - \boldsymbol{f}^{\boldsymbol{h}\,-}_{\boldsymbol{ih}}\right)_y = 2\cos(\alpha)\sin(\beta)g, \tag{4.39}$$

and use these to define

$$P = \frac{1}{2g}\left(\boldsymbol{f}^{\boldsymbol{h}\,+}_{\boldsymbol{ih}} - \boldsymbol{f}^{\boldsymbol{h}\,-}_{\boldsymbol{ih}}\right)_x \quad \text{and} \quad Q = \frac{1}{2g}\left(\boldsymbol{f}^{\boldsymbol{h}\,+}_{\boldsymbol{ih}} - \boldsymbol{f}^{\boldsymbol{h}\,-}_{\boldsymbol{ih}}\right)_y \tag{4.40}$$

Thus,

$$P = \sin(\alpha)\cos(\beta) \quad \text{and} \quad Q = \cos(\alpha)\sin(\beta). \tag{4.41}$$

This can be used to determine α and β, as

$$P^2 = \sin^2(\alpha)\cos^2(\beta) = \sin^2(\alpha) - \sin^2(\alpha)\sin^2(\beta) \implies \sin^2(\alpha) = P^2 + \sin^2(\alpha)\sin^2(\beta) \quad (4.42)$$

$$\implies \sin^2(\alpha) = P^2 + \sin^2(\alpha)\left(\frac{Q}{\cos(\alpha)}\right)^2 = P^2 + Q^2\tan^2(\alpha) \quad (4.43)$$

This can be used to determine α by a simple iterative process where the small angle approximation with Equation 4.41, gives $P = \sin(\alpha)\cos(\beta) \approx \alpha$ so the initial step is

$$\alpha_1 = P \quad (4.44)$$

and the iterative step is

$$\alpha_{n+1} = \sin^{-1}(\text{sign}(\alpha_n)\sqrt{(P^2 + Q^2 \times (\tan\alpha_n)^2)}) \quad (4.45)$$

stopping when $\alpha_{n+1} = \alpha_n$, which takes about three iterations to reach the numerical precision limit in MATLAB, which is far higher than the expected accuracy.

β can be solved using an equivalent iterative process with $\beta_1 = Q$ and the iterative step,

$$\beta_{n+1} = \sin^{-1}(\text{sign}(\beta_n)\sqrt{(Q^2 + P^2 \times (\tan\beta_n)^2)}). \quad (4.46)$$

These estimates of α and β are only used in the rest of the estimation process in the term "$\cos(\alpha)\cos(\beta)g$", as an improved estimate of the specific force in the table z-direction, which resolves a consistent underestimate of the accelerometer scale factor when the table is not level.

It is also important to note that a the true specific forces of $\boldsymbol{f}_{\boldsymbol{ih}}^{\boldsymbol{h}\,+} - \boldsymbol{f}_{\boldsymbol{ih}}^{\boldsymbol{h}\,-}$ is not available to determine P and Q, only the measured values are available. What is used instead is the average of all 12 possible 180° heading pairs from manoeuvre set C of the *measured* specific force. As this is a difference between two measurement the sensors' biases have no effect, see Section 4.5.1.2, however the other sensor errors and the cube's imperfections still affect the result.

It would be possible to extend the six-position test (set A) to a seven-position test where there are two readings on only one of the faces at 180° heading difference. This would still allow an estimate of α and β, albeit a less accurate one. However, this would not have the advantage of cancelling out the effect of non-zero table angles on the two 'horizontal' measurements that is achieved by taking the mean of all four readings from each of the faces (see Equation 4.56), which turns out to be the more significant effect, the results of this are presented in Section 4.8 as the 7-position method.

4.5.1.2 Pairwise sums and differences

The actual method of calculating the estimates of the sensor errors from the static estimates is very simple, given the three main assumptions that the table is level, the box is cubic and the effect of the noise on the time-averaged static measurements is small enough to be neglected by the calibration algorithm. All 3 of these assumptions are tested in the simulation. Thus, the main experimental parameters which might be expected to affect

the outcome of the calibration are essentially how accurate these assumptions are, which is characterised (in the simulation) by the SD of the angle by which the first two are wrong, and how long the sample of data collected on each side is, with a longer sample reducing the effect of sensor noise.

The simple procedures implemented here only use the (time-)average of each of the six sensor readings during the static period on each cube face. These orientations were defined in Section 4.3.3. For example, 'ZU' refers to the 'Z-up' orientation and is considered the standard orientation in this thesis, and 'XD' is the 'X-down' orientation. A general orientation is referred to as 'NO'.

In the simple procedure (set A) these are the averages over a single time-period and in the advanced procedure (set C) these are further averaged over each of the four headings with equal weighting[2].

The sensor readings at each orientation can be denoted by a 'given that' operator, for example, '$\boldsymbol{f_{ib}^{b}}\,|_{ZU}$' for 'true specific force when in the ZU orientation'. However for reasons of brevity the notation is shortened to '$\boldsymbol{f^{ZU}}$'. Similarly the notation for angular rate is $\boldsymbol{\omega_{ib}^{b}}\,|_{ZU} = \boldsymbol{\omega^{ZU}}$. For the respective measured quantities $\widetilde{\boldsymbol{f_{is}^{s}}}\,|_{ZU} = \widetilde{\boldsymbol{f^{ZU}}}$ and $\widetilde{\boldsymbol{\omega_{is}^{s}}}\,|_{ZU} = \widetilde{\boldsymbol{\omega^{ZU}}}$

Also note that this procedure ignores the effect of the non-linearity terms from Section 4.4. Thus Equations 4.7 and 4.8 become,

$$\widetilde{\boldsymbol{f_{is}^{s}}} \approx \boldsymbol{b_a} + \mathbf{A}_{\mathbf{a}}^{\mathbf{s}}(\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}\boldsymbol{f_{ib}^{b}}) + \boldsymbol{w_a} \tag{4.47}$$

$$\widetilde{\boldsymbol{\omega_{is}^{s}}} \approx \boldsymbol{b_g} + \mathbf{A}_{\mathbf{g}}^{\mathbf{s}}(\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}\boldsymbol{\omega_{ib}^{b}}) + \mathbf{G}_{\mathbf{g}}^{\mathbf{s}}(\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}\boldsymbol{f_{ib}^{b}}) + \boldsymbol{w_g} \tag{4.48}$$

If we further assume that time averaging, when conducted over a sufficient period, will make the $\boldsymbol{w_a}$ and $\boldsymbol{w_g}$ insignificant, then

$$\overline{\widetilde{\boldsymbol{f_{is}^{s}}}} \approx \boldsymbol{b_a} + \mathbf{A}_{\mathbf{a}}^{\mathbf{s}}\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}\boldsymbol{f_{ib}^{b}} \tag{4.49}$$

$$\overline{\widetilde{\boldsymbol{\omega_{is}^{s}}}} \approx \boldsymbol{b_g} + \mathbf{A}_{\mathbf{g}}^{\mathbf{s}}\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}\boldsymbol{\omega_{ib}^{b}} + \mathbf{G}_{\mathbf{g}}^{\mathbf{s}}\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}\boldsymbol{f_{ib}^{b}} \tag{4.50}$$

If assumptions of a cubic box and a flat table are valid. This makes the expected values for the true angular rate and specific force in the 'Z-up' orientation under static conditions,

$$\boldsymbol{f^{ZU}} = \begin{bmatrix} f_x^{ZU} \\ f_y^{ZU} \\ f_z^{ZU} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{4.51}$$

$$\boldsymbol{\omega^{ZU}} = \begin{bmatrix} \omega_x^{ZU} \\ \omega_y^{ZU} \\ \omega_z^{ZU} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.52}$$

This gives,

$$\overline{\widetilde{\boldsymbol{f^{ZU}}}} = \boldsymbol{b_a} + \mathbf{A}_{\mathbf{a}}^{\mathbf{s}}\mathbf{C}_{\mathbf{b}}^{\mathbf{s}} \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{4.53}$$

[2] Although an average of 2 headings at 180° heading apart would have provided most of the benefit.

$$\overline{\widetilde{\boldsymbol{\omega}^{\boldsymbol{ZU}}}} = \boldsymbol{b}_{\boldsymbol{g}} + \mathbf{A}^{\mathbf{s}}_{\mathbf{g}}\mathbf{C}^{\mathbf{s}}_{\mathbf{b}}\begin{bmatrix}0\\0\\0\end{bmatrix} + \mathbf{G}^{\mathbf{s}}_{\mathbf{g}}\mathbf{C}^{\mathbf{s}}_{\mathbf{b}}\begin{bmatrix}0\\0\\g\end{bmatrix} \tag{4.54}$$

This illustrates why a dynamic procedure is required to observe $\mathbf{A}^{\mathbf{s}}_{\mathbf{g}}\mathbf{C}^{\mathbf{s}}_{\mathbf{b}}$. When we do not assume that the table is level (set C) the true specific forces (like in Equation 4.37) for each of the four 90°-separated headings of set C (see Figure 4.4), that is a heading angle (see Section 4.3.5) of 0°, 90°, 180° and 270°, become

$$\boldsymbol{f}^{\boldsymbol{ZU1}} = \begin{bmatrix}\sin(\alpha)\cos(\beta)g\\ \cos(\alpha)\sin(\beta)g\\ \cos(\alpha)\cos(\beta)g\end{bmatrix}, \quad \boldsymbol{f}^{\boldsymbol{ZU2}} = \begin{bmatrix}-\cos(\alpha)\sin(\beta)g\\ \sin(\alpha)\cos(\beta)g\\ \cos(\alpha)\cos(\beta)g\end{bmatrix},$$
$$\boldsymbol{f}^{\boldsymbol{ZU3}} = \begin{bmatrix}-\sin(\alpha)\cos(\beta)g\\ -\cos(\alpha)\sin(\beta)g\\ \cos(\alpha)\cos(\beta)g\end{bmatrix}, \quad \boldsymbol{f}^{\boldsymbol{ZU4}} = \begin{bmatrix}\cos(\alpha)\sin(\beta)g\\ -\sin(\alpha)\cos(\beta)g\\ \cos(\alpha)\cos(\beta)g\end{bmatrix} \tag{4.55}$$

Thus the set C equivalent of Equation 4.49 is

$$\begin{aligned}\overline{\widetilde{\boldsymbol{f}^{\boldsymbol{ZU}}}} &= \frac{1}{4}\left(\overline{\widetilde{\boldsymbol{f}^{\boldsymbol{ZU1}}}} + \overline{\widetilde{\boldsymbol{f}^{\boldsymbol{ZU2}}}} + \overline{\widetilde{\boldsymbol{f}^{\boldsymbol{ZU3}}}} + \overline{\widetilde{\boldsymbol{f}^{\boldsymbol{ZU4}}}}\right)\\ &= \frac{1}{4}\left(\boldsymbol{b}_{\boldsymbol{a}} + \mathbf{A}^{\mathbf{s}}_{\mathbf{a}}\mathbf{C}^{\mathbf{s}}_{\mathbf{b}}\boldsymbol{f}^{\boldsymbol{ZU1}} + \boldsymbol{b}_{\boldsymbol{a}} + \mathbf{A}^{\mathbf{s}}_{\mathbf{a}}\mathbf{C}^{\mathbf{s}}_{\mathbf{b}}\boldsymbol{f}^{\boldsymbol{ZU2}} + \boldsymbol{b}_{\boldsymbol{a}} + \mathbf{A}^{\mathbf{s}}_{\mathbf{a}}\mathbf{C}^{\mathbf{s}}_{\mathbf{b}}\boldsymbol{f}^{\boldsymbol{ZU3}} + \boldsymbol{b}_{\boldsymbol{a}} + \mathbf{A}^{\mathbf{s}}_{\mathbf{a}}\mathbf{C}^{\mathbf{s}}_{\mathbf{b}}\boldsymbol{f}^{\boldsymbol{ZU4}}\right)\\ &= \boldsymbol{b}_{\boldsymbol{a}} + \mathbf{A}^{\mathbf{s}}_{\mathbf{a}}\mathbf{C}^{\mathbf{s}}_{\mathbf{b}}\left(\frac{1}{4}\left(\begin{bmatrix}\sin(\alpha)\cos(\beta)g\\ \cos(\alpha)\sin(\beta)g\\ \cos(\alpha)\cos(\beta)g\end{bmatrix} + \begin{bmatrix}-\cos(\alpha)\sin(\beta)g\\ \sin(\alpha)\cos(\beta)g\\ \cos(\alpha)\cos(\beta)g\end{bmatrix}\right.\right.\\ &\quad \left.\left. + \begin{bmatrix}-\sin(\alpha)\cos(\beta)g\\ -\cos(\alpha)\sin(\beta)g\\ \cos(\alpha)\cos(\beta)g\end{bmatrix} + \begin{bmatrix}\cos(\alpha)\sin(\beta)g\\ -\sin(\alpha)\cos(\beta)g\\ \cos(\alpha)\cos(\beta)g\end{bmatrix}\right)\right)\\ &= \boldsymbol{b}_{\boldsymbol{a}} + \mathbf{A}^{\mathbf{s}}_{\mathbf{a}}\mathbf{C}^{\mathbf{s}}_{\mathbf{b}}\begin{bmatrix}0\\0\\\cos(\alpha)\cos(\beta)g\end{bmatrix}\end{aligned} \tag{4.56}$$

By a similar process of substitution the equivalent of Equation 4.50 is

$$\overline{\widetilde{\boldsymbol{\omega}^{\boldsymbol{ZU}}}} = \frac{1}{4}\left(\overline{\widetilde{\boldsymbol{\omega}^{\boldsymbol{ZU1}}}} + \overline{\widetilde{\boldsymbol{\omega}^{\boldsymbol{ZU2}}}} + \overline{\widetilde{\boldsymbol{\omega}^{\boldsymbol{ZU3}}}} + \overline{\widetilde{\boldsymbol{\omega}^{\boldsymbol{ZU4}}}}\right) = \boldsymbol{b}_{\boldsymbol{g}} + \mathbf{G}^{\mathbf{s}}_{\mathbf{g}}\mathbf{C}^{\mathbf{s}}_{\mathbf{b}}\begin{bmatrix}0\\0\\\cos(\alpha)\cos(\beta)g\end{bmatrix} \tag{4.57}$$

The fact that the components of $\overline{\widetilde{\boldsymbol{f}^{\boldsymbol{ZU1}}}}$ to $\overline{\widetilde{\boldsymbol{f}^{\boldsymbol{ZU4}}}}$ only appear in the sum $\left(\overline{\widetilde{\boldsymbol{f}^{\boldsymbol{ZU1}}}} + \overline{\widetilde{\boldsymbol{f}^{\boldsymbol{ZU2}}}} + \overline{\widetilde{\boldsymbol{f}^{\boldsymbol{ZU3}}}} + \overline{\widetilde{\boldsymbol{f}^{\boldsymbol{ZU4}}}}\right)$ means that Equation 4.56 is not sensitive to the order or identification of the set C rotations (and Equation 4.57 for the same reason). This has implications which will be discussed in Section 4.5.3.

If we now consider pairs of gyroscope or accelerometer readings taken on opposite faces, for example 'ZU' and 'ZD', now for both the true angular rate would be zero and the true specific force would be of equal magnitude but apply in the opposite direction. So if we take

the sum of two, the component due to specific force should cancel (assuming a level table)

$$\overline{\widetilde{\boldsymbol{f}^{\boldsymbol{ZU}}}}+\overline{\widetilde{\boldsymbol{f}^{\boldsymbol{ZD}}}} = \boldsymbol{b_a}+\mathbf{A}_\mathbf{a}^\mathbf{s}\mathbf{C}_\mathbf{b}^\mathbf{s}\begin{bmatrix}0\\0\\g\end{bmatrix}+\boldsymbol{b_a}+\mathbf{A}_\mathbf{a}^\mathbf{s}\mathbf{C}_\mathbf{b}^\mathbf{s}\begin{bmatrix}0\\0\\-g\end{bmatrix} = 2\boldsymbol{b_a}+\mathbf{A}_\mathbf{a}^\mathbf{s}\mathbf{C}_\mathbf{b}^\mathbf{s}\left(\begin{bmatrix}0\\0\\g\end{bmatrix}+\begin{bmatrix}0\\0\\-g\end{bmatrix}\right) \tag{4.58}$$

and for the difference the bias cancels

$$\overline{\widetilde{\boldsymbol{f}^{\boldsymbol{ZU}}}}-\overline{\widetilde{\boldsymbol{f}^{\boldsymbol{ZD}}}} = \boldsymbol{b_a}+\mathbf{A}_\mathbf{a}^\mathbf{s}\mathbf{C}_\mathbf{b}^\mathbf{s}\begin{bmatrix}0\\0\\g\end{bmatrix}-\boldsymbol{b_a}-\mathbf{A}_\mathbf{a}^\mathbf{s}\mathbf{C}_\mathbf{b}^\mathbf{s}\begin{bmatrix}0\\0\\-g\end{bmatrix} = \mathbf{A}_\mathbf{a}^\mathbf{s}\mathbf{C}_\mathbf{b}^\mathbf{s}\left(\begin{bmatrix}0\\0\\g\end{bmatrix}-\begin{bmatrix}0\\0\\-g\end{bmatrix}\right) = \mathbf{A}_\mathbf{a}^\mathbf{s}\mathbf{C}_\mathbf{b}^\mathbf{s}\begin{bmatrix}0\\0\\2g\end{bmatrix}. \tag{4.59}$$

Note that, if using the set C measurements *without assuming a level table*, these equations become

$$\overline{\widetilde{\boldsymbol{f}^{\boldsymbol{XU}}}}-\overline{\widetilde{\boldsymbol{f}^{\boldsymbol{XD}}}} = \mathbf{A}_\mathbf{a}^\mathbf{s}\mathbf{C}_\mathbf{b}^\mathbf{s}\begin{bmatrix}2\cos(\alpha)\cos(\beta)g\\0\\0\end{bmatrix}, \tag{4.60}$$

$$\overline{\widetilde{\boldsymbol{f}^{\boldsymbol{YU}}}}-\overline{\widetilde{\boldsymbol{f}^{\boldsymbol{YD}}}} = \mathbf{A}_\mathbf{a}^\mathbf{s}\mathbf{C}_\mathbf{b}^\mathbf{s}\begin{bmatrix}0\\2\cos(\alpha)\cos(\beta)g\\0\end{bmatrix}, \tag{4.61}$$

$$\overline{\widetilde{\boldsymbol{f}^{\boldsymbol{ZU}}}}-\overline{\widetilde{\boldsymbol{f}^{\boldsymbol{ZD}}}} = \mathbf{A}_\mathbf{a}^\mathbf{s}\mathbf{C}_\mathbf{b}^\mathbf{s}\begin{bmatrix}0\\0\\2\cos(\alpha)\cos(\beta)g\end{bmatrix}. \tag{4.62}$$

These equations motivate the error estimation methods presented below.

4.5.1.3 Bias estimation

Two alternative algorithms are implemented to estimate the sensor biases from the static measurements, both can use either set A or C manoeuvres.

In the first technique, henceforth Technique 1, the accelerometer bias is estimated by summing all six faces, this reduces errors by increasing the effective averaging time, reducing the effect of noise. Although two faces are sufficient to get a more coarse estimate, see Equation 4.67.

$$\widehat{\boldsymbol{b_a}} = \frac{1}{6}\left(\overline{\widetilde{\boldsymbol{f}^{\boldsymbol{XU}}}}+\overline{\widetilde{\boldsymbol{f}^{\boldsymbol{XD}}}}+\overline{\widetilde{\boldsymbol{f}^{\boldsymbol{YU}}}}+\overline{\widetilde{\boldsymbol{f}^{\boldsymbol{YD}}}}+\overline{\widetilde{\boldsymbol{f}^{\boldsymbol{ZU}}}}+\overline{\widetilde{\boldsymbol{f}^{\boldsymbol{ZD}}}}\right). \tag{4.63}$$

If the box was perfectly cuboid (and placed precisely as expected), then we can use Equation 4.58

$$\widehat{\boldsymbol{b_a}} = \frac{1}{6}\left(2\boldsymbol{b_a}+2\boldsymbol{b_a}+2\boldsymbol{b_a}\right) = \boldsymbol{b_a}, \tag{4.64}$$

which is the target value in Equation 4.17.

The gyroscope bias estimate is conducted in an analogous way, namely,

$$\widehat{\boldsymbol{b_g}} = \frac{1}{6}\left(\overline{\widetilde{\boldsymbol{\omega}^{\boldsymbol{XU}}}}+\overline{\widetilde{\boldsymbol{\omega}^{\boldsymbol{XD}}}}+\overline{\widetilde{\boldsymbol{\omega}^{\boldsymbol{YU}}}}+\overline{\widetilde{\boldsymbol{\omega}^{\boldsymbol{YD}}}}+\overline{\widetilde{\boldsymbol{\omega}^{\boldsymbol{ZU}}}}+\overline{\widetilde{\boldsymbol{\omega}^{\boldsymbol{ZD}}}}\right) \tag{4.65}$$

It can be seen that by, for example the similarity between Equations 4.53 and 4.54, under the perfect cube assumption, these will cancel in the same pairwise fashion as in Equation 4.63.

$$\widehat{\boldsymbol{b_g}} = \frac{1}{6}\left(2\boldsymbol{b_g} + 2\boldsymbol{b_g} + 2\boldsymbol{b_g}\right) = \boldsymbol{b_g} \tag{4.66}$$

which is the target value in Equation 4.24.

There is reason to believe (see Section 4.8 specifically Equation 4.83) that an alternative algorithm for estimating the accelerometer and gyroscope biases may be less sensitive to cube and/or table alignment errors, although more sensitive to some other error sources (e.g. sensor noise). The following alternative algorithm is implemented in the simulation, described as Technique 2,

$$\widehat{\boldsymbol{b_a}} = \begin{bmatrix} \widehat{b_{ax}} \\ \widehat{b_{ay}} \\ \widehat{b_{az}} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} \overline{\widetilde{f^{XU}}}_x + \overline{\widetilde{f^{XD}}}_x \\ \overline{\widetilde{f^{YU}}}_y + \overline{\widetilde{f^{YD}}}_y \\ \overline{\widetilde{f^{ZU}}}_z + \overline{\widetilde{f^{ZD}}}_z \end{bmatrix}. \tag{4.67}$$

Where the final x, y or z subscripts and non-bold script signify that this refers to a single component of the vector (similarly in Equation 4.69). This is justified by the same pairwise elimination of other errors as Equation 4.63. As follows, taking only the third row of Equation 4.58 gives

$$\overline{\widetilde{\boldsymbol{f^{ZU}}}} + \overline{\widetilde{\boldsymbol{f^{ZD}}}} = 2\boldsymbol{b_a} \implies \overline{\widetilde{f^{ZU}}}_z + \overline{\widetilde{f^{ZD}}}_z = 2\widehat{b_{az}} \tag{4.68}$$

Which is the third row of Equation 4.67, the other 2 rows follow by similarity. Note that this can be calculated with manoeuvre set A or C.

If the IMU is not nominally lined up with the box-frame, then the face-pairs would need to be re-arranged (rather than the lower-case 'x', 'y' and 'z'). The equivalent Technique 2 method for $\widehat{\boldsymbol{b_g}}$ is,

$$\widehat{\boldsymbol{b_g}} = \begin{bmatrix} \widehat{b_{gx}} \\ \widehat{b_{gy}} \\ \widehat{b_{gz}} \end{bmatrix} = \frac{1}{2}\begin{bmatrix} \overline{\widetilde{\omega^{XU}}}_x + \overline{\widetilde{\omega^{XD}}}_x \\ \overline{\widetilde{\omega^{YU}}}_y + \overline{\widetilde{\omega^{YD}}}_y \\ \overline{\widetilde{\omega^{ZU}}}_z + \overline{\widetilde{\omega^{ZD}}}_z \end{bmatrix}. \tag{4.69}$$

This Technique 2 calibration algorithm is also calculated and presented in the results of this chapter (Section 4.8), so that the two possibilities can be compared.

4.5.1.4 $\mathbf{A_a}$ and $\mathbf{G_g}$ estimation

While the average of single pair of opposite face readings would have been sufficient for the bias estimation, in order to observe all the entries in $\widehat{\mathbf{A_a}}$ and $\widehat{\mathbf{G_g}}$ all six readings are required. $\widehat{\mathbf{A_a}}$ is estimated as follows.

$$\widehat{\mathbf{A}}_{\mathbf{a}}^{\mathbf{b}} = \frac{1}{2\cos(\alpha)\cos(\beta)g}\left[\left(\overline{\widetilde{\boldsymbol{f^{XU}}}} - \overline{\widetilde{\boldsymbol{f^{XD}}}}\right), \left(\overline{\widetilde{\boldsymbol{f^{YU}}}} - \overline{\widetilde{\boldsymbol{f^{YD}}}}\right), \left(\overline{\widetilde{\boldsymbol{f^{ZU}}}} - \overline{\widetilde{\boldsymbol{f^{ZD}}}}\right)\right] \tag{4.70}$$

Using Equations 4.60, 4.61 and 4.62 this becomes

$$\widehat{\mathbf{A}_{\mathbf{a}}^{\mathbf{b}}} \approx \frac{1}{2\cos(\alpha)\cos(\beta)g}\left[\left(\mathbf{A}_{\mathbf{a}}^{\mathbf{s}}\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}\begin{bmatrix}2\cos(\alpha)\cos(\beta)g\\0\\0\end{bmatrix}\right),\right.$$
$$\left.\left(\mathbf{A}_{\mathbf{a}}^{\mathbf{s}}\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}\begin{bmatrix}0\\2\cos(\alpha)\cos(\beta)g\\0\end{bmatrix}\right),\left(\mathbf{A}_{\mathbf{a}}^{\mathbf{s}}\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}\begin{bmatrix}0\\0\\2\cos(\alpha)\cos(\beta)g\end{bmatrix}\right)\right]$$
$$= \left[\left(\mathbf{A}_{\mathbf{a}}^{\mathbf{s}}\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}\begin{bmatrix}1\\0\\0\end{bmatrix}\right),\left(\mathbf{A}_{\mathbf{a}}^{\mathbf{s}}\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}\begin{bmatrix}0\\1\\0\end{bmatrix}\right),\left(\mathbf{A}_{\mathbf{a}}^{\mathbf{s}}\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}\begin{bmatrix}0\\0\\1\end{bmatrix}\right)\right]$$
$$= \mathbf{A}_{\mathbf{a}}^{\mathbf{s}}\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}\left[\begin{bmatrix}1\\0\\0\end{bmatrix},\begin{bmatrix}0\\1\\0\end{bmatrix},\begin{bmatrix}0\\0\\1\end{bmatrix}\right] = \mathbf{A}_{\mathbf{a}}^{\mathbf{s}}\mathbf{C}_{\mathbf{b}}^{\mathbf{s}} \quad (4.71)$$

which is the target value in Equation 4.20.

If using set A measurements, then replace "$2\cos(\alpha)\cos(\beta)g$" with "$2g$", and use equations of the form of Equation 4.59 to prove this in the same manner as Equation 4.71, assuming a level table.

$\widehat{\mathbf{G}_{\mathbf{g}}^{\mathbf{b}}}$ is estimated as follows.

$$\widehat{\mathbf{G}_{\mathbf{g}}^{\mathbf{b}}} = \frac{1}{2\cos(\alpha)\cos(\beta)g}\left[\left(\overline{\widetilde{\boldsymbol{\omega}^{\boldsymbol{XU}}}} - \overline{\widetilde{\boldsymbol{\omega}^{\boldsymbol{XD}}}}\right),\left(\overline{\widetilde{\boldsymbol{\omega}^{\boldsymbol{YU}}}} - \overline{\widetilde{\boldsymbol{\omega}^{\boldsymbol{YD}}}}\right),\left(\overline{\widetilde{\boldsymbol{\omega}^{\boldsymbol{ZU}}}} - \overline{\widetilde{\boldsymbol{\omega}^{\boldsymbol{ZD}}}}\right)\right] \quad (4.72)$$

Which gives $\widehat{\mathbf{G}_{\mathbf{g}}^{\mathbf{b}}} \approx \mathbf{G}_{\mathbf{g}}^{\mathbf{s}}\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}$, as required for Equation 4.30. This is justified by exactly the same method as Equation 4.70.

4.5.2 The dynamic part of the calibration procedure

This section presents the procedure for estimating the gyroscopes' alignment matrix, $\mathbf{A}_{\mathbf{g}}$, which, as presented in Section 2.2.5, contains the scale-factor and cross-coupling gyroscope errors. As gyroscopes measure angular rate, an estimate of $\mathbf{A}_{\mathbf{g}}$ cannot be made from static data, there needs to be movement[3]. As was discussed in Section 2.5.1, the normal calibration method is to mount the gyroscope on a rate table and/or rigidly to a much higher performance gyroscope and then compare the output of the two [4]. This allows an accurate angular rate signal to be compared with the gyroscopes actual angular rate output, which makes estimation of both bias and $\mathbf{A}_{\mathbf{g}}$ relatively straightforward, and, if different magnitudes of angular rate are included, allows non-linearity terms to be estimated also. However, when one considers calibration methods which might be suitable for user-conducted calibration, neither of these techniques is suitable, for reasons of cost.

Some authors present "low-cost" calibration methods using a single-axis turntable as e.g. [153, 162], or some other specially constructed calibration apparatus (e.g. [156]). There are other

[3]Some authors have used the rotation of the Earth to calibrate their sensors, e.g. [160], while 'static' but consumer-grade MEMS are not accurate enough to use that as a signal.

authors who use hand rotations, but these either involve a very complex procedure, where the required operating precision is very high [159] or high computational costs and/or reference to additional sensors (e.g. machine learning to integrate magnetometers in [161]).

In this chapter, the technique for calibrating the gyroscopes' alignment matrix uses the 360° rotations of either manoeuvre set B or D, and it requires only a flat surface[4] and a 'corner' to calibrate the IMU within its housing. Its chief advantage is that it requires no expensive external equipment and its main disadvantage is that the accuracy of the results will depend on how well the procedure is carried out by the user. There are two further disadvantages. First, that the accuracy depends on the other systematic errors of the gyroscope, and to a lesser extent the accelerometers via the g-dependent error[5], being estimated accurately and calibrated out. Second, that gyroscope noise reduces the accuracy, as it conducts dead-reckoning over the time interval in question (t) . This is because the measurements are integrated, which makes the effect of noise $\propto \sqrt{t}$, rather than averaged (as was the case in Section 4.5.1), which makes the effect of noise $\propto \frac{1}{\sqrt{t}}$ [1]. This means that conducting the rotation quickly will (in general) make the calibration more accurate, unless it is so fast that the sensors exceed their measurement range (clipping the output).

This technique relies on the IMU returning to its original orientation, integrating the output while moving; as such, it is a zero attitude update.

A realistic method for conducting a fairly accurate zero-attitude update is to begin by lining up the box in a corner of a flat surface (e.g. the edge and back of a shelf on a book-shelf). Two examples of such a corner are shown in Figure 4.11. This lined-up position is used to force the orientation (attitude) to be the same before and after a manoeuvre. If the IMU is lined up before a manoeuvre, if afterwards it is returned to *the same corner* and positioned on *the same face* of the box and at *the same orientation*. If conducted in this way the total change in orientation should be very close to zero. This is because the relative orientation change solution is unaffected by angular errors of the cube's construction (as it returns to the same face), and the tables deviation from level, and additionally because the difference in table-frame heading between the 'before' and 'after' orientations will be very small, due to the 'corner'.

In this dynamic part of the calibration, the only error term that is estimated is $\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}$. This is a 3x3 matrix containing the gyroscopes' scale factor and cross-coupling errors as well as describing its mounting angle within the cube (see Equation 4.25). It is needed (Equation 4.12) to transform the measured angular rate $\widetilde{\boldsymbol{\omega}_{\boldsymbol{ib}}^{\boldsymbol{b}}}$ into an estimated angular rate $\widehat{\boldsymbol{\omega}_{\boldsymbol{ib}}^{\boldsymbol{b}}}$.

In a perfectly aligned and error-free IMU this estimated $\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}$ would be a 3x3 identity matrix $\mathbf{I_3}$. However, as was noted in Equation 4.25, this matrix is should be a combination of a rotation matrix, representing the IMU's alignment with the box-frame, and another matrix ($\mathbf{A}_{\mathbf{g}}^{\mathbf{s}}$) representing the scale-factor and cross-coupling errors, $\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}} \approx \mathbf{A}_{\mathbf{g}}^{\mathbf{s}}\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}$. This $\mathbf{A}_{\mathbf{g}}^{\mathbf{s}}$ being

[4] It does not need it to be level.

[5] This is because the Equation 4.12 that calculates $\widehat{\boldsymbol{\omega}_{\boldsymbol{ib}}^{\boldsymbol{b}}}$ uses the value of $\widehat{\boldsymbol{f}_{\boldsymbol{ib}}^{\boldsymbol{b}}}$ from Equation 4.11 under the assumption that $\widehat{\boldsymbol{f}_{\boldsymbol{ib}}^{\boldsymbol{b}}} = \boldsymbol{f}_{\boldsymbol{ib}}^{\boldsymbol{b}}$, that is that the errors have been corrected. Note that $\boldsymbol{w_a} \neq 0$, as there is no time averaging, so even if the accelerometer systematic errors are estimated perfectly then $\widehat{\boldsymbol{f}_{\boldsymbol{ib}}^{\boldsymbol{b}}} \approx \boldsymbol{f}_{\boldsymbol{ib}}^{\boldsymbol{b}} + \boldsymbol{w_a}$ and so accelerometer noise has a direct effect on $\widehat{\boldsymbol{\omega}_{\boldsymbol{ib}}^{\boldsymbol{b}}}$ and this the technique relies on this being small due to the small numerical values of $\mathbf{G_g}$.

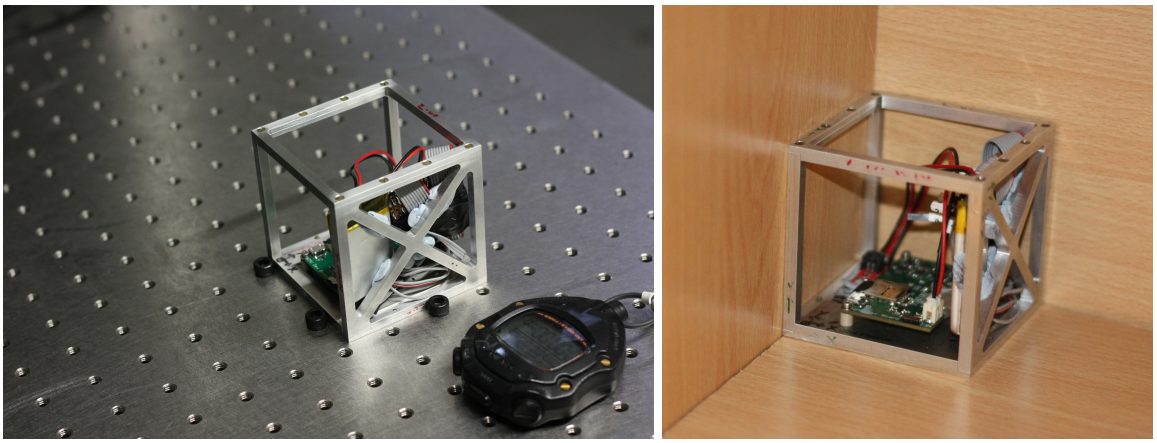


Figure 4.11: The calibration box in a corner. The 'corner' used for the experiments in Chapter 5 (left), and another possible example (right)

effectively an identity matrix with a small (up to ±3%) perturbation on all 9 entries. This combination means that it does not have the simplifying properties of a rotation matrix which only has 3 degrees of freedom. So $\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}$ has the full nine degrees of freedom of a generic 3×3 matrix.

In order to observe $\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}$, we must assume that the (post-compensation) effect of the other systematic errors is small. That is,

$$\widehat{\boldsymbol{f}_{\boldsymbol{is}}^{\boldsymbol{s}}} \approx \boldsymbol{f}_{\boldsymbol{ib}}^{\boldsymbol{b}} + \boldsymbol{w}_{\boldsymbol{a}} \tag{4.73}$$

$$\widehat{\boldsymbol{\omega}_{\boldsymbol{is}}^{\boldsymbol{s}}}^{-} \approx \mathbf{A}_{\mathbf{g}}^{\mathbf{s}}\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}\boldsymbol{\omega}_{\boldsymbol{ib}}^{\boldsymbol{b}} + \boldsymbol{w}_{\boldsymbol{g}} = \widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}^{-}\boldsymbol{\omega}_{\boldsymbol{ib}}^{\boldsymbol{b}} + \boldsymbol{w}_{\boldsymbol{g}} \tag{4.74}$$

where, $\widehat{\boldsymbol{\omega}_{\boldsymbol{is}}^{\boldsymbol{s}}}^{-}$ denotes the corrected angular rate without the correct estimate of $\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}$ included[6].

Even with these simplifications, at least 3 separate rotation manoeuvres about non-coplanar axes are required to estimate $\mathbf{A}_{\mathbf{g}}^{\mathbf{s}}\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}$. It is assumed that, after the 360° rotation, the net angular-position change is zero (i.e. the rotation matrix representing the rotation from start to end positions is $\mathbf{I_3}$). Under this assumption, any overall measured attitude change (integrated from $\widehat{\boldsymbol{\omega}_{\boldsymbol{is}}^{\boldsymbol{b}}}^{-}$) over the interval is an attitude-error. Note that this can be represented by a 3×3 matrix, but this is a rotation matrix so only contains 3 independent pieces of information. This is not enough to estimate the 9 unknowns. Consider this example; there is a an attitude-error about the x-axis. Is this error due to the x-sensor scale factor, or either y- or z-axis motion being incorrectly interpreted as x-axis motion (cross-coupling), or even some combination of these? That is, it could come from any of the elements in the first row of $\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}$.

The minimum number of separate rotation-and-replacement sets that make it possible for determination of all the elements of $\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}$ is three, provided that the rotation axes are non-coplanar. To increase the chance that this the procedure makes $\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}$ observable the three manoeuvres should be as different as possible. The two examples tested in this chapter,as

[6]The algorithms require some pre-estimate of $\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}$, denoted by $\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}^{-}$. One example might be $\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}^{-} = \mathbf{I_3}$, assuming that the b and s frames are nominally aligned, alternative starting estimates will be covered in Section 4.5.2.1.

mentioned in Section 4.2, are: to pick up and rotate the IMU about each of the cardinal axes one by one (replacing it in between) (set B); and to keep the box in contact with the table while rotating 360° for each face, one-by-one (set D).

These rotations about the cardinal axes are primarily selected because they are the easiest for the user to perform without specialist equipment. However, they also can be used to form a relatively simple analytical solution. If the rotation is perfectly about the intended axis the true angular rate will be only about that axis, so any error resulting from the rotation about the x-axis will only be due to the first column of the $\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}$ matrix, the y-axis rotation will only be affected by the second column, and the z-axis rotation the third.

This simplification is not valid when the actual axis of rotation is a few degrees off from the intended cardinal axis. However, this chapter investigates how accurate the rotations need to be to provide an improvement on no calibration.

It is also important to note that the accuracy of the estimated $\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}$ obtained by any technique using measured IMU outputs depends on how well corrected the other error sources are and how significant the un-calibrated errors (e.g. noise and non-linearity) are, as we are assuming that the other sources of error have been removed. As can be seen from Equation 4.16 the corrected angular rate also uses the corrected specific force (for g-dependant bias correction), so errors in the corrected specific force still have a (very) small effect on the corrected angular rate.

There are a number of approximations made in the calculation of $\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}$, which rely on corrections being small, such as small angle approximations. For this reason and to feed back the more accurate estimates into Equation 4.74, the calibration algorithms tested here are iterative processes.

4.5.2.1 An initial estimate

Any iterative process to estimate $\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}$ needs to be initialised, which must begin by using the best current estimate of $\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}$ and the statically-determined estimates to correct the measured angular rate, using Equation 4.74.

One possibility is to use the nominal orientation of the IMU, if the sensor and box-average frames are nominally aligned[7] this is $\mathbf{I_3}$.

Another possible initial value for $\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}$ is to use the (static test derived) estimate of $\widehat{\mathbf{A}_{\mathbf{a}}^{\mathbf{b}}}$. This is because in a typical case the alignment error between the IMU and the box will be greater than the cross-coupling error within the IMU. A typical cross-coupling error of ±2% corresponds to approximately 1 degree of alignment error which makes $\widehat{\mathbf{A}_{\mathbf{a}}^{\mathbf{b}}}$ a good first estimate of $\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}$ whenever the mounting error is significantly greater than 2 degrees (1 degree of cross-coupling from the accelerometer plus 1 degree from the gyroscope). That is,

$$\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}} \approx \widehat{\mathbf{A}_{\mathbf{a}}^{\mathbf{b}}} \implies \mathbf{C}_{\mathbf{b}}^{\mathbf{s}}\mathbf{A}_{\mathbf{g}}^{\mathbf{s}} \approx \mathbf{C}_{\mathbf{b}}^{\mathbf{s}}\mathbf{A}_{\mathbf{a}}^{\mathbf{s}} \tag{4.75}$$

[7] This is considered to be the case for the simulations in this chapter, but is not for the physical experiments in Chapter 5, for which the nominal orientations are given in Figure A.4.

However, it should be noted that this method makes cannot correct for any gyroscope scale factor, in fact it presumes that the accelerometer scale factor and gyroscope scale factor are equal, thus it is potentially better to use 1 for the diagonal entries.

4.5.2.2 Method 1 - sum of angular increments

This method is called the sum method because it uses the sum of angular increments as a proxy for integration. This method relies on an iterative process to estimate $\widehat{\mathbf{A}_{\mathbf{g}}}$. This has a number of steps that must be repeated until convergence. The estimation process could have been initialised with $\widehat{\mathbf{A}_{\mathbf{g}_{(1)}}} = \mathbf{I_3}$ but, as it will in general be better, the results presented here use $\widehat{\mathbf{A}_{\mathbf{g}_{(1)}}} = \widehat{\mathbf{A}_{\mathbf{a}}}$. This method relies on the three rotations being carried out around the sensor cardinal axes, and deviation from the sensors' cardinal axes will decrease its reliability. It uses the assumption that the motion has taken place around a box-frame cardinal axis, and the resulting error in $\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}$ is heavily dependent on this assumption being true.

First the measured sensor output must be corrected with the current best estimates of the sensor errors, including the current estimate of $\widehat{\mathbf{A}_{\mathbf{g}_{(i)}}}$.

The sum is taken of all of the angular increments during each of the rotation periods. This is taken as a (computationally fast) proxy for actual integration of the full angular rate solution, it is not a correct solution because it neglects the non-commutativity of rotations, although when the axis-error angles are small enough it may be sufficient. It is also more stable (as seen in Section 4.8).

$$\widehat{\mathbf{A}_{\mathbf{g}_{(i+1)}}} = \left(\begin{bmatrix} \sum \widehat{\boldsymbol{\omega_{xX}}} & \sum \widehat{\boldsymbol{\omega_{xY}}} & \sum \widehat{\boldsymbol{\omega_{xZ}}} \\ \sum \widehat{\boldsymbol{\omega_{yX}}} & \sum \widehat{\boldsymbol{\omega_{yY}}} & \sum \widehat{\boldsymbol{\omega_{yZ}}} \\ \sum \widehat{\boldsymbol{\omega_{zX}}} & \sum \widehat{\boldsymbol{\omega_{zY}}} & \sum \widehat{\boldsymbol{\omega_{zZ}}} \end{bmatrix} \div (2\pi R) \right) \quad \widehat{\mathbf{A}_{\mathbf{g}_{(i)}}} \tag{4.76}$$

Where R is the output data rate (in Hertz), needed to turn the angular rate into angular increments, and $\sum \widehat{\boldsymbol{\omega_{wK}}}$ refers to the sum over the period of the rotation about the 'K' axis of the 'w' axis gyroscope measurements (rad/s) corrected using Equation 4.12. The '2π' is the total rotation (radians) expected by the algorithm.

If the rotation takes place perfectly about a single axis and the sensors are already perfectly calibrated the bracketed expression in Equation 4.76 would equal the identity matrix this correction may get smaller between successive runs, although the approximations made will mean that it requires several iterations.

4.5.2.3 Method 2 - integrating method

This method is similar to the sum method, in that the solution is initialised an estimate such as $\widehat{\mathbf{A}_{\mathbf{g}_{(0)}}} = \widehat{\mathbf{A}_{\mathbf{a}}}$ and successive $\widehat{\mathbf{A}_{\mathbf{g}_{(i)}}}$ estimates are solved for iteratively.

First the full navigation equations (see Section 2.2.3) are run over the three periods of motion in question leading to calculated attitude changes of $\mathbf{C}_{\mathbf{Xend}}^{\mathbf{Xstart}}$, $\mathbf{C}_{\mathbf{Yend}}^{\mathbf{Ystart}}$ and $\mathbf{C}_{\mathbf{Zend}}^{\mathbf{Zstart}}$ for the motion about the X-, Y- and Z-axes, respectively.

This attitude change is then converted to column-vectors of Euler angles $\boldsymbol{e_X}$, $\boldsymbol{e_Y}$ and $\boldsymbol{e_Z}$, e.g.

$$\mathbf{C}^{\mathbf{Xstart}}_{\mathbf{Xend}} = \boldsymbol{e_X} = \begin{bmatrix} \phi_X \\ \theta_X \\ \psi_X \end{bmatrix} \tag{4.77}$$

where the attitude change of the X-axis rotation is denoted in terms of yaw (ψ_X), pitch (θ_X) and roll (ϕ_X) These Euler angles are then divided by the total rotation (2π), gives a normalised error for each axis which can be combined (by the assumption that each rotation is perfectly about an axis) into a correction matrix. Then the iterative process is,

$$\widehat{\mathbf{A}_{\mathbf{g}(i+1)}} = \widehat{\mathbf{A}_{\mathbf{g}(i)}} + \left(\begin{bmatrix} \boldsymbol{e_X} & \boldsymbol{e_Y} & \boldsymbol{e_Z} \end{bmatrix} \div (2\pi) \right) \tag{4.78}$$

Note that, the assumption that the Euler angle formulation of the total attitude error ($\boldsymbol{e_X}$, $\boldsymbol{e_Y}$ and $\boldsymbol{e_Z}$) is equivalent to the errors induced on a single axis is an approximation that relies on the corrections being small. As will be shown in Section 4.8, this makes this method sensitive more sensitive to the initialisation errors. However, this method is less sensitive to the errors in the rotation axis than the sum method.

An alternative formulation for the iterative step is,

$$\widehat{\mathbf{A}_{\mathbf{g}(i+1)}} = \widehat{\mathbf{A}_{\mathbf{g}(i)}} \left(1 + \left(\begin{bmatrix} \boldsymbol{e_X} & \boldsymbol{e_Y} & \boldsymbol{e_Z} \end{bmatrix} \div (2\pi) \right) \right). \tag{4.79}$$

This proved to be less stable than Equation 4.78 in preliminary tests so was not implemented in the final simulation.

4.5.2.4 Method 3 - the hybrid method

This method is a combination of the initial estimate and the sum method. Here the sum method is carried out, to give $\widehat{\mathbf{A}_{\mathbf{g}\Sigma}}$ but only the diagonal entries are used. The off-diagonal entries in the estimate are derived from $\widehat{\mathbf{A}_{\mathbf{a}}}$ as in the initial estimate. That is,

$$\widehat{\mathbf{A}_{\mathbf{g}_{\mathrm{hyb}}}} = \begin{bmatrix} \widehat{A_{g\Sigma\ 1,1}} & \widehat{A_{a1,2}} & \widehat{A_{a1,3}} \\ \widehat{A_{a2,1}} & \widehat{A_{g\Sigma\ 2,2}} & \widehat{A_{a2,3}} \\ \widehat{A_{a3,1}} & \widehat{A_{a3,2}} & \widehat{A_{g\Sigma\ 3,3}} \end{bmatrix} \tag{4.80}$$

It might be reasonably expected that this method will perform well when the mounting error is large in comparison to the accelerometer and gyroscope cross-coupling error, that is, when Equation 4.75 applies.

4.5.2.5 The difference between using manoeuvre set B and D

Two separate sets of manoeuvres are tested in this Chapter; the minimum 3 rotations shown in Figure 4.3 (set B) and a set of 6 rotations, one on each face of the cube, as shown in Figure 4.5 (set D).

The set D measurements differ from the set B in two important ways, they take place with the box remaining in contact with the table, secondly, they take place on all six faces so there are 6 measurements available rather than 3.

Both the sum and hybrid methods were used are calculated using the set D measurements, and the correction at each step of the iterative process averaged between the 'up' and 'down' faces of the box.

Using the face as a guide makes rotating the box about a particular cardinal axis depend on the accuracy of the box's construction rather than the skill of the user. The user-related variation that remains after the rotation axis is fixed, that is, the degree of variation in rotation speed is not relevant to the calibration procedure provided it is not so fast that the gyroscope saturates. This will generally make the rotation much closer to the intended axis as is reflected by the default settings (see Section 4.6.1) of 1° for the box SD and 5° for the part B rotation axis error. Second, the fact that there are two measurements taking place for each cardinal axis means that the effect of the noise is averaged between the two (reducing its effect by a factor of $\sqrt{2}$). Additionally because these two measurements are on opposite faces, specific force applies in the opposite direction to each thus (nearly) removing the effect of residual g-dependent error[8]. Finally because the rotations are in the same direction in the table frame (and from the point of view of the user), they are in opposite directions in the box frame, this removes part of the effect of residual gyroscope bias. Finally, because in our model the faces' deviations from cubic are independent, the axis error of each rotation is independent, so averaging the two effectively makes this error's effect on the results $\sqrt{2}$ smaller.

4.5.2.6 Alternative numerical solutions

There are alternatives to coming up with an analytic solution to calculate $\widehat{\mathbf{A_g}}$. There are various numerical techniques that can determine minima of output functions using methods without any specific knowledge of the underlying structure, these can be referred to generally as global optimisation techniques. Some authors have already applied machine learning algorithms to IMU calibration, such as in for example [161] see Section 2.5.1.3.

Note that, one reason why analytic solutions are preferable is that many applications for low-cost inertial navigation would use low-cost embedded hardware and thus processing power will be limited by battery life and the cost of high-performance processors. For this type of platform, running machine learning algorithms every time the calibration procedure is run becomes highly impractical, particularly if this means there is a significant delay needed to calculate the estimates from the procedure.

A preliminary experiment was run using the simulated annealing method [181], which demonstrated the limitations of this approach. The objective function which compensates the IMU readings with the new estimates and re-integrates the dynamic part of the procedure took around 3 seconds, and, as this is a nine-dimensional optimisation, this resulted in it taking very long time to produce a final estimate. This makes it impractical for most low-cost applications, and thus it is not presented here.

[8] This is only 'nearly' removed because the cube is not perfect.

4.5.3 Sensitivity to user error

As the user calibration methods presented in this chapter rely on the user to carry out a specific set of manoeuvres, it is reasonable to expect that these manoeuvres be simple enough that they take no particular degree of skill to accomplish. It should also be expected that the kind of errors that an average user is likely to make can either be compensated for or detected so that the procedure can be repeated.

As mentioned in Section 4.2, some types of gross-user errors are not considered in the simulation. However in the experiments presented in Chapter 5, both automatic detection of the static periods, and manoeuvre error detection were implemented.

The nature of the rotations depicted in Figures 4.2 to 4.5, in combination with a 'corner' as in Figure 4.11 make certain types of procedural mistakes very unlikely, e.g. rotating the box 330° in one of the manoeuvres in set D. The following possible mistakes are much more likely, based on the author's experience performing at least 50 of each of manoeuvre sets A, B, C and D for experiments including those presented in Chapter 5.

The user collects the data from the faces in the 'wrong order' or forgets one.
Given that the one of the main current uses for low-cost MEMS accelerometers is tilt sensing and the box is known to be cuboid, detecting 'which face is down' is trivially easy, as even uncalibrated consumer-grade accelerometers can be used to estimate pitch and roll to within 10°. Production software could use this to fix data collected in the wrong order, although the prototype software only flags these for manual fixing. If one of the face's static data is forgotten or invalid, it is still possible to compute (less accurate) estimates of $\boldsymbol{b_a}$, $\boldsymbol{b_g}$, $\mathbf{A_a}$ and $\mathbf{G_g}$. Equation 4.58 shows that measurements from a single pair of opposite faces is sufficient to estimate $\boldsymbol{b_a}$ (or $\boldsymbol{b_g}$), so if for example the orientation 'XD' were forgotten then Equation 4.63 could be adapted to be $\widehat{\boldsymbol{b_a}} = \frac{1}{4}\left(\overline{\widetilde{\boldsymbol{f^{YU}}}} + \overline{\widetilde{\boldsymbol{f^{YD}}}} + \overline{\widetilde{\boldsymbol{f^{ZU}}}} + \overline{\widetilde{\boldsymbol{f^{ZD}}}}\right)$. Using this $\widehat{\boldsymbol{b_a}}$ then the term, $\left(\overline{\widetilde{\boldsymbol{f^{XU}}}} - \overline{\widetilde{\boldsymbol{f^{XD}}}}\right)$ in Equation 4.70, can be replaced by $\left(2\left(\overline{\widetilde{\boldsymbol{f^{XU}}}} - \widehat{\boldsymbol{b_a}}\right)\right)$. This can be shown by subtracting $\widehat{\boldsymbol{b_a}}$ from both sides of Equation 4.53, and comparing it to Equation 4.62. This could actually be extended to two missing measurements making 4 the minimum set of measurements needed, as long as there was at least one from each pair of opposite faces. An equivalent method can be used to estimate $\boldsymbol{b_g}$ and $\mathbf{G_g}$ from less than 6 measurements. Forgetting one of the rotations for set B cannot be fixed, but the user can easily be prompted to repeat that axis' rotation. Set D also requires at least one rotation around each axis, but as there are two of each in the manoeuvre set, a reduced accuracy estimate can be made with some of measurements missing (or invalid).

The user gets the 'nominal orientation' wrong.
The specific 'nominal orientations' given in Section 4.3.3 are used in the simulation, but are not needed by the algorithm. So getting it 'wrong' is actually impossible, all that matters is having the correct face upwards. The algorithm using the set A measurements does not use the heading of the box at all. Set C likewise requires that the 4 headings are 90° apart, but does not need these all to be precisely lined up with the table frame. The dynamic manoeuvre sets B and D only require the same orientation at the start and the end of the

manoeuvres what that orientation is is irrelevant.

During set C, the user accidentally collects 3 or 5 different headings rather than 4, or rotates in the wrong direction collecting them in the 'wrong order'
This is surprisingly easy to do. Marking one corner of each face makes it less likely. Integrating the gyroscope output (or simply summing angular increments) between the end of one static period and the start of the next makes it simple to distinguish 90° heading changes from knocks on the table ($< 5°$ computed attitude change) or 360° (e.g. set D) manoeuvres this can in turn be used to detect extra measurements, which can be ignored, or whether there are missing ones. The average of four 90° separated measurements in Equation 4.57 can be replaced by the average of two 180° ones if one of the four is missing, e.g. average of $\widetilde{\overline{\boldsymbol{f}^{\boldsymbol{ZU1}}}}$ and $\widetilde{\overline{\boldsymbol{f}^{\boldsymbol{ZU3}}}}$ can be used if $\widetilde{\overline{\boldsymbol{f}^{\boldsymbol{ZU4}}}}$ is missing without significantly affecting the results. Also, in Equation 4.56 and 4.57 it does not matter if the rotations are in the 'wrong order' because the heading rotations were conducted in the other direction.

During set B or D, the user rotates the cube in the 'wrong direction' or by the wrong amount e.g. 270°.
This is very easy to fix if the sum of angular increments over the manoeuvre is approximately -360° when 360° was expected then, rather than dividing by '2π' in Equations 4.76 and/or 4.78, divide by '-2π'. Similarly the user is unlikely to accidentally replace the cube in the corner between 90° orientations, so if the sum of angular increments about the rotation axis can be rounded to the nearest $\frac{\pi}{2}$ radians and that used instead of the 2π in Equations 4.76 and/or 4.78. The initial IMU calibration should be good enough that the nearest 90° should be the correct amount.

During set B or D, the user rotates the test-bed too fast.
If the box is rotated too quickly the sensor output will be clipped. This is easy to detect but cannot be fixed. The manoeuvre could be marked as invalid, and the user prompted to repeat it.

Accidentally triggering the static detection while holding the IMU in ones hands. This is virtually impossible with proper thresholds in the stationarity detection algorithms.

The main remaining user-related error is failing to return the box to exactly the same heading, i.e. getting it a few degrees wrong by failing to press the box back into the 'corner' properly. This is one of the errors included in the simulation, and as the results in Sections 4.8.1.3 and 4.8.1.5 show, this has only a small effect on the calibration accuracy.

4.6 Simulation overview

This chapter uses Monte Carlo simulations to examine the difference between the true calibration conditions and those assumed by the calibration algorithm on the accuracy of the resulting calibration. This will answer questions like "How does the angular accuracy of the box in which the sensors are mounted affect the estimate of the accelerometer scale factor and cross-coupling matrix?" or "How does the amount of time for which the sensors remain static on each face of the cube effect the accuracy of the gyroscope bias estimate?". Some of

these questions, like for instance the second of those posed above can be answered analytically. However, for some errors, particularly those that can only be determined dynamically, there is no simple analytic formulation, so results of the Monte Carlo simulations will be particularly instructive.

This section outlines the way in which the Monte Carlo simulation tests a particular set of input parameters. Many sets of input parameters were tested in order to determine the effect of each parameter and how the different calibration algorithms and manoeuvre sets compare to each other.

4.6.1 Input parameters

The calibration simulation's input parameters are divided into those which affect how well the manoeuvres are carried out (how precisely, how accurately, over how much time), in 'TESTspec', and the raw performance of the IMU before calibration 'IMUspec'. Some of these specify a fixed value but most are used to generate the various randomly determined errors that are described in the later sections and these input parameters are the standard deviations (SDs) used to determine the various errors used in the simulation. Note, that the default values referred to in the following list are those chosen as a starting point to which the other errors are fixed as the inputs are varied one-by-one.

The file 'TESTspec' comprises the following information:

1. 'ODR' - the output data rate at which the inertial sensors are being sampled for the procedure (default: 100Hz)
2. 'WaitTime' - The amount of time for which the IMU is left static on each face of the cube during sets A and C (default: 10 sec)
3. 'cube face error' - the SD used to generate the angle by which each face of the cube deviates (in both pitch and roll) from a perfect cube. See Figure 4.8 and Section 4.3.4.
4. 'cube half length' and 'cube half length error' the distance each face of the cube is from the centre of the cube and by how much this varies (default: 0.05m ± 0.001m - i.e. a 10cm cube with length errors of SD 1mm each end)
5. 'table angle' - The SD of the (two) angles by which the table deviates from perfectly level (around East and North) (default: 1°), see Section 4.3.6.
6. 'replacementHeadErrorSetA', 'replacementHeadErrorSetB' , 'replacementHeadErrorSetC' and'replacementHeadErrorSetD'- the SD used to generate the angle by which the testbed is rotated from the heading it was intended to be replaced at each time it replaced on the table. (defaults: 10° for set A, 0.5° for B, C and D, all converted to radians). See Figure 4.9 and Section 4.3.5.
7. 'setA position', 'setB position', 'setC position' , 'setD position'- the SD of the position (in table frame) (x and y, the z is from 'cubehalflengtherror' and is fixed) error each time the testbed is replaced on the table (defaults 20mm for set A 1mm for sets B, C and D)
8. 'TransTime' - the time over which the sensors move to a new orientation during sets A and C (default: 2 sec); this has no effect on the results.

9. 'setB waittime' - the time for which the testbed is held static between rotations in set B (default: 2 sec); this has no effect on the results.
10. 'setB rotation time' - the time taken for each rotation in set B and D (default: 10 sec). Note that too short a rotation time will cause the sensors to 'clip' as they exceed their measurement range.
11. 'setB midair position Error' - the SD of the position error (in all 3 dimensions) from the target point 10cm above the origin point at the start of each rotation (default: 1cm).
12. 'setB midair start Att Error' - the SD of the attitude error (in all 3 dimensions) from the target orientation at the start of each rotation (default: 10°).
13. 'setB midair axisError' - the SD of the angular error (in elevation and azimuth) from the target axis for each rotation (default: 5°).
14. 'setB midair spinError' - the SD of the angle by which the rotation over- or undershoots 360° (default: 10°).

The file 'IMUspec' comprises the following information:

1. 'gyro range' - The maximum output which the gyroscope can sense. Any measured gyroscope output in excess of this is clipped, i.e. set to the maximum positive or negative value (default: 250 °/s ≈ 4.4 rad/s).
2. 'gyro adcword', 'gyro sensitivity', 'gyro precision' - These can all equivalently determine the quantization level, or the smallest amount of angular rate that can be sensed. They are all defined as different sensors may specify different equivalent formulations (default: word 16-bit, precision 1.3×10^{-4} rad/s).
3. 'gyro biasSD' - the SD used to generate the gyroscopes' bias for each axis (default 5 °/s ≈ 0.087 rad/s).
4. 'gyro scalefactorSD' - the SD used to generate the gyroscopes' scale factor error on each axis (default: 0.03 unitless).
5. 'gyro crossaxisSD' - the SD used to generate the gyroscopes' cross coupling error on each axis (default: 0.02 unitless).
6. 'gyro nonlinearity' - the SD used to generate the gyroscopes' six non-linearity error constants, two for each axis to separately apply to positive and negative angular rates (default: 0.001 of full scale output). The error is assumed to be quadratic, so these are the quadratic coefficients.
7. 'gyro gdepSD' - the SD used to generate the gyroscopes' g-dependent bias error on each axis (0.1 °/s/g ≈ 1.78e-04rad m s^{-1}).
8. 'gyro noisepsd' - the gyroscope sensor noise root-PSD. (default: 0.01 °/s/$\sqrt{\text{Hz}}$ ≈ 1.75×10^{-4} rad/s /$\sqrt{\text{Hz}}$)
9. 'accel range' - The maximum output which the accelerometer can sense. Any measured output in excess of this is clipped, i.e. set to the maximum positive or negative value. (Default: $2g \approx 19.6 \text{ms}^{-2}$)
10. 'accel adcword', 'accel sensitivity', 'accel precision' - These can all equivalently determine the quantization level, or the smallest amount of specific force that can be sensed. They are all defined as different sensors may specify different equivalent formulations (default: word 16-bit, precision 5.99×10^{-4} ms^{-2}).

11. 'accel scalefactorSD' - the SD used to generate the accelerometers' scale factor error on each axis (default: 0.03 unitless).
12. 'accel nonlinearity' - the SD used to generate the accelerometers' six non-linearity error constants, two for each axis to separately apply to positive and negative specific forces (default: 0.005 of full scale output). The error is assumed to be quadratic, so these are the quadratic coefficients.
13. 'accel crossaxisSD' - the SD used to generate the accelerometers' cross axis error on each axis (default: 0.02 unitless).
14. 'accel biasSDxy' 'accel biasSDz' - the SD used to generate the accelerometers' bias for each axis (default: 60 milli-g ≈ 0.588 ms^{-2}). The Z-axis can be specified separately as sometimes the IMU specifications differ between the axes.
15. 'accel noisepsd' - the accelerometer sensor noise root-PSD. (default: 300 micro-g/$\sqrt{\text{Hz}} \approx 0.0029\text{ms}^{-2}/\sqrt{\text{Hz}}$)
16. 'orientationerrorSD pr' 'orientationerrorSD y' 'PCBorientationerrorSD pr' 'PCBorientationerrorSD y' these mounting errors determine the angle between the nominal box frame and the actual IMU frame ($\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}$, see Sections 4.3.2 and 4.4.1). There are two separate errors to allow arrays of IMUs to be modelled. See Figure 4.7 and Section 4.3.2 (defaults: 5°, 2°, 1° and 1°, respectively).

Note that, the calibration procedure assumes that all the errors generated from each SDs are independent identically distributed random variables (IIDRVs) and are Gaussian.

4.6.2 Simulation process

The simulation process is presented in Figure 4.12. The accuracy with which the procedure is carried out and specifications of the IMU are the simulation's inputs (TESTspec and IMUspec from Section 4.6.1). First, 500 full sets of errors are generated from the input SDs for the simulation trajectory (TESTerrors) and the IMU (IMUerrors). The IMUerrors parameter contains all the fixed, but unknown to the procedure, values which are used in the IMU model both those that are estimated directly, e.g. $\boldsymbol{b_g}$, or indirectly, e.g. $\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}$, and those that are not estimated, e.g. non-linearity parameters. The random errors, i.e. $\boldsymbol{w_a}$ and $\boldsymbol{w_g}$, are not stored in IMUerrors but generated randomly when the measured IMU readings are generated. The TESTerrors parameter contains all the true values for the trajectory generation, e.g. the table angles and each heading replacement error, this stores all the information required to generate the true trajectory, and thus also all the information needed to generate the true inertial readings, $\boldsymbol{f_{ib}^b}$ and $\boldsymbol{\omega_{ib}^b}$. Then the Monte Carlo part of the procedure starts, in the case of the results presented here there are 500 repetitions carried out.

This repeating part begins by using the i$^{\text{th}}$ set of TESTerrors to generate the trajectory of an imperfectly carried out calibration procedure. This trajectory includes an imperfect version of the set A, B, C and D manoeuvres. This trajectory is used to determine what the output of a perfect IMU would be if it underwent this trajectory. Then the i$^{\text{th}}$ set of IMUerrors are used to generate the output that the particular IMU being calibrated would use, using the error models in Section 4.4.1. This static part of this measured IMU output

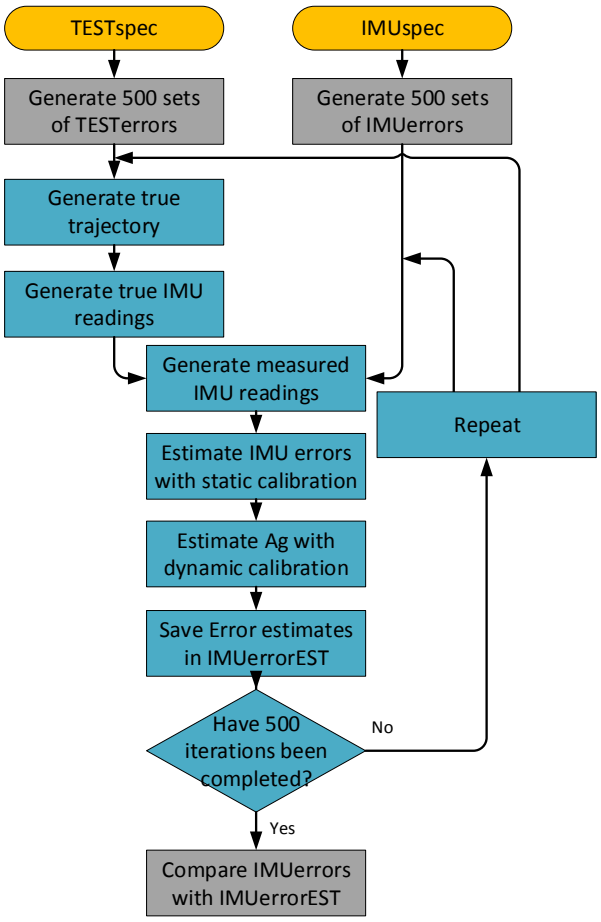


Figure 4.12: Flow Diagram of the Monte Carlo simulation algorithm. Inputs in yellow and Monte Carlo part in blue.

is then used by the algorithms described in Section 4.5.1, to calculate a $\widehat{\boldsymbol{b_a}}$, $\widehat{\boldsymbol{b_g}}$, $\widehat{\mathbf{A_a}}$ and $\widehat{\mathbf{G_g}}$ estimate. Then the output of the advanced (set C) static procedure is used to correct the IMU output, using the method described in Section 4.4.2. This corrected output is then used in the dynamic part of output using the algorithms described in Section 4.5.2 to calculate various estimates of $\widehat{\mathbf{A_g}}$. This is then saved as the i^{th} entry of IMUerrorEST.

Once this has been repeated 500 times, the estimates stored in IMUerrorEST are compared with the true values of the estimated IMU errors which are stored in IMUerrors in the manner which will be described in Section 4.7 to calculate the accuracy of each of the various techniques for calibration. The results of this comparison are the output of the simulation.

4.7 Calculation of calibration accuracy

In order to determine the accuracy of the calibration consistently with different input values a comparison method is needed. The metrics that matter for post-calibration navigation performance are the errors remaining after calibration, also known as residual errors. In a physical test these values cannot be known, but the purpose of running simulations is so that these true values are known.

Most navigation applications of low-cost IMUs would involve integrating them in a Kalman filter architecture and because a KF describes the errors only in terms of second moments (variances), then higher order statistics concerning the residual error from the calibration process are not relevant. This is how the KF based experiments presented in Chapter 3 considered the sensors, so this is the level on which a comparison with those results need to be made, and thus information on third or fourth statistical moments in will not be presented the results in this chapter, noting that the simulation generates errors from Gaussian distributions.

For each run in the Monte Carlo simulation the ideal, or target, outputs of the calibration are described by Equations 4.17, 4.20, 4.24, 4.30 and 4.25, including, where applicable, compensating for the mounting error of the sensors within the box ($\mathbf{C_b^s}$). Then the simplest comparison we can make is the difference between the calibration value (for each particular error and repeat) and the target value. Then the root-mean-squared (RMS) is taken over each axis or matrix element and over all the Monte Carlo repeats as a measure of how effective the calibration is. Mean values were also computed for debugging purposes, but are not presented here. RMS is used because it takes into account any systematic over- or under-estimation of the calibration procedure which would not be reflected in the SD.

It is possible that there might be a difference in the accuracy of the calibration for, e.g. the biases on the various axes, this was checked for during debugging but is not presented as it is not present. In a real system, differences in the noise density for the sensor axes would result in slightly different levels of accuracy, but in the simulation presented here there is no difference, and as the procedure is entirely symmetric there is no reason for any asymmetric results. So, rather than examining X- Y- and Z- axis results as 3 separate lines on the figures each the RMS over 500 data points, we examine the RMS over all 3 axes together so the RMS is calculated over 1500 data points, as the procedure and inputs are symmetric there is no reason to expect that any additional insight will be gained by examining them separately, so a single less-noisy line is shown for each bias-estimation technique. The axes are homogeneous as all the procedure uses the same number of measurements with each axis vertical, thus the estimates of each axis are equally affected by gravity.

The quality of the estimates for the $\widehat{\mathbf{A}_\mathbf{a}^\mathbf{b}}$, $\widehat{\mathbf{A}_\mathbf{g}^\mathbf{b}}$ and $\widehat{\mathbf{G}_\mathbf{g}^\mathbf{b}}$ matrices were measured as follows. In an analogous way to that of the bias, the residual errors (i.e. the difference between target and estimated values) for every Monte Carlo run and all nine entries (weighted equally) were used to calculate one RMS over the 4500 points (9 entries $\times$ 500 runs). This is the simplest viable scheme, and is used to keep consistency across all the error types.

This is a sensible method of assessing $\widehat{\mathbf{G}_\mathbf{g}}$ where all the entries are IIDRVs, and the term which cancels the effect of g-dependent error in Equation 4.29 is $\widehat{\mathbf{G}_\mathbf{g}^\mathbf{b}} - \mathbf{G}_\mathbf{g}^\mathbf{s}\mathbf{C}_\mathbf{b}^\mathbf{s} \approx 0$.

For $\widehat{\mathbf{A}_\mathbf{a}^\mathbf{b}}$ and $\widehat{\mathbf{A}_\mathbf{g}^\mathbf{b}}$ the results of the diagonal and off-diagonal elements could have been examined separately, particularly as they are specified separately, but in this Chapter the overall measurement is used instead. Also note that the alignment error cancels in Equation 4.29 as $\widehat{\mathbf{A}_\mathbf{g}^\mathbf{b}}^{-1}\mathbf{A}_\mathbf{g}^\mathbf{s}\mathbf{C}_\mathbf{b}^\mathbf{s} \approx \mathbf{I_3}$ so the difference $(\widehat{\mathbf{A}_\mathbf{g}^\mathbf{b}} - \mathbf{A}_\mathbf{g}^\mathbf{s}\mathbf{C}_\mathbf{b}^\mathbf{s})$ is an approximation of the residual error not the exact value.

If no calibration were carried out, the assumption would be that $\widehat{\boldsymbol{b_a}}$ and $\widehat{\boldsymbol{b_g}}$ were $\begin{bmatrix}0 & 0 & 0\end{bmatrix}^T$,

that $\widehat{\mathbf{A_a}}$ and $\widehat{\mathbf{A_g}}$ were $\mathbf{I_3}$ and the $\widehat{\mathbf{G_g}}$ matrix was $\mathbf{0_3}$, as these are the average (expected) values. Thus, the starting calibration for each simulation, shown by the green line in the figures below, was calculated by comparing these expected values with the actual errors generated. This shows the pre-calibration RMS error in a consistent manner with the error estimates above. The green line is not totally flat in the graphs due to it only being a finite sample, generated from the input SD (which is illustrated by a dotted green line). This is needed to compare the results of the various calibration procedures with no calibration in a consistent manner, particularly $\widehat{\mathbf{A}_\mathbf{g}^\mathbf{b}}$ and $\widehat{\mathbf{A}_\mathbf{a}^\mathbf{b}}$ as mentioned above.

4.8 Simulation results

This section presents the results of the Monte Carlo simulations described in the preceding section.

The approach that is taken is to test a set of default input parameters and vary the other input parameters one-by-one. The default value for each input parameter is listed in Section 4.6.1, the intention being that the default IMUspec represents a typical MEMS IMU (represented by the Invensense MPU-9250 [23]) and the 'testSPEC' defaults a realistic level of precision in the calibration procedure.

It is important to note that this approach can only give a snapshot of the effect of each error, in the sense that varying a single particular input parameter will only show its relative significance relative to when the others are at default level. However, if reducing the error tested below default level results in an improvement in accuracy that input is a significant source of error. Conversely, if the error has to be raised by an order of magnitude or more to have a visible effect on the calibration accuracy then that error is not a significant source of error at default level.

When considering the results of this chapter, it is important to note that the simulation and *the theoretical idea* of pre-calibrating the sensors are based on some modelling assumptions. The calibration procedure estimates constant and first-order errors, and some second order errors are also included in the simulation's model, but real sensors exhibit even higher order errors even though these are relatively small in relation to the 1^{st}- and 2^{nd}-order sources (see Section 2.2.5). While these are small effects for uncalibrated low-cost MEMS, if the constant (bias) and linear (scale factor and cross coupling) errors are very well calibrated then the higher-order errors will be more relatively more important. However, a more important assumption is that the systematic error sources coefficients are constant in time. This implies that the coefficient variation during the calibration and between the time of calibration and use is small enough that its effect can be ignored. Despite it being known that temperature affects both biases and scale factor errors, and that they also vary in-run. The validity of these assumptions will depend on the size of the target residual errors. That is, when the residual errors in the constant and linear terms become very small, limits of the domain of validity of both these modelling assumptions are reached. This is most likely to effect $\widehat{\boldsymbol{b_g}}$, which as will be shown in e.g. Figure 4.13b is frequently estimated to better a tactical level of accuracy, at this level the fixed coefficients assumption is unlikely to be

valid, at least for more than a few minutes. Thus the validity of the modelling assumptions for $\boldsymbol{b_g}$ becomes the limiting factor for the accuracy of $\widehat{\boldsymbol{b_g}}$ rather than the procedure or pre-calibration IMU errors.

The graphs presented in this section, Figures 4.13 to 4.31, represent the results in a similar manner. Each sub-graph represents one of the sensor errors estimated in the calibration procedure. The X-axis shows the input parameter that is being varied and the Y-axis is the measure of the quality of the estimate, i.e. RMS of the residual error (as described in Section 4.7). The green lines represent the errors before the calibration procedure is applied, i.e. the un-calibrated errors. The red lines represent the simpler techniques, i.e. those using the set A or B data. The blue lines represent the more complex techniques, i.e. the set C or D data. For $\widehat{\boldsymbol{b_a}}$ and $\widehat{\boldsymbol{b_g}}$ the Technique 1 results are represented by a solid line, and the Technique 2 results by a dotted line. In the $\widehat{\mathbf{A}^{\mathbf{b}}_{\mathbf{a}}}$ and $\widehat{\mathbf{G}^{\mathbf{b}}_{\mathbf{g}}}$ results the 7-position results are represented with a dotted line. In the $\widehat{\mathbf{A}^{\mathbf{b}}_{\mathbf{g}}}$ results the different algorithms sum-method, integrated method and hybrid method are represented by different lines, with different data-point markers allowing them to be compared.

Note that, the default inputs, which were defined in Section 4.6.1, are shown in each of this section's graphs by a black vertical dotted line. The solid horizontal black lines represent tactical IMU performance, defined as the minimum values tested in Chapter 3; see Table 3.1. The horizontal magenta lines represent the target values that are the results of Chapter 3, 1 °/s of gyroscope bias, medium accelerometer first-order errors and low gyroscope first-order errors, see Table 3.3.

Note that, if the experiments were run with markedly different default settings, such as for example starting with a tactical level IMU, the relative effects of the input parameters would be different.

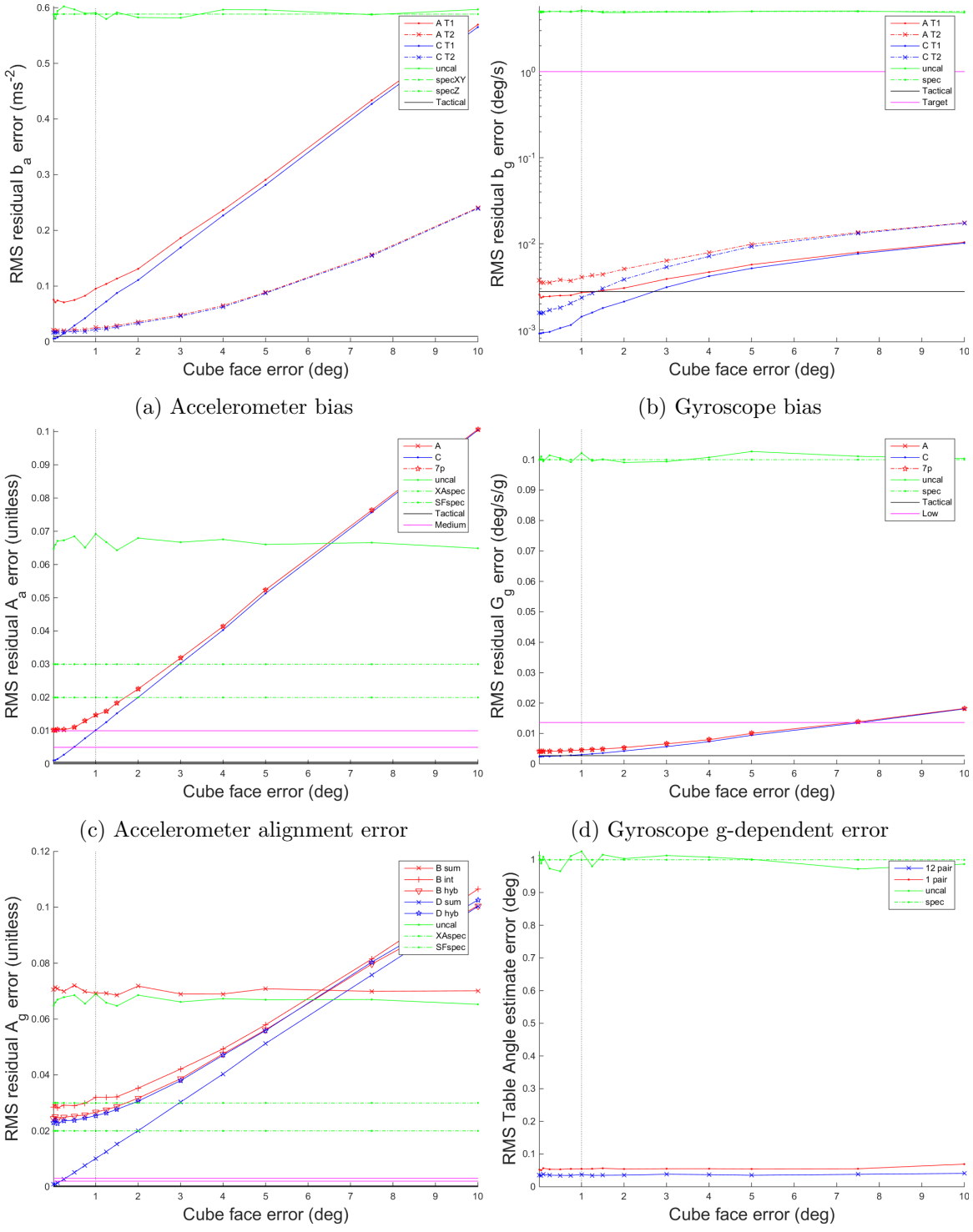
4.8.1 The effect of the calibration procedure's accuracy

There are a large number of ways in which the calibration procedure could deviate from the being perfectly carried out. This section presents the effect of varying the input parameters, which describe how well the calibration procedure has been carried out, one-by-one.

4.8.1.1 Cube accuracy

The first result presented here, in Figure 4.13, is for the effect of the angular precision of manufacture of the calibration cube in which the IMU is mounted. This is one of the most significant error sources, and affects all the estimated errors, so all six are shown. This error is simulated by the parameter cube face error (see Section 4.6.1) which is the SD used to generate the two face errors for each face in each Monte Carlo run as described in Section 4.3.4 and Figure 4.8.

In order to examine the effect of this error a wide range of angular precisions were tested ranging from 0.01° to 10°. This range of inputs is clearly unrealistic, although they are simulated to examine the effect of taking the accuracy to extremes. A box where the sides



(a) Accelerometer bias

(b) Gyroscope bias

(c) Accelerometer alignment error

(d) Gyroscope g-dependent error

(e) Gyroscope alignment error

(f) Table angle estimate error

Figure 4.13: The effect of the cube's angular precision on the residual errors post-calibration.

deviate from cubic by a SD of 10 degrees would not even look very much like a cube, more something a 6 year old might make from cardboard and sellotape. Equally if calibration cubes with an angular accuracy of 0.01° needed to be made for each IMU, it would probably be cheaper just to use standard factory calibration techniques. The default level chosen of 1° SD is not a particularly challenging level of angular accuracy, and does not represent high-cost precision manufacturing, rather the level of accuracy the IMU's housing would have without making any additional effort to make it accurate. Also note that, while the assumption of the simulation is that the cube deviations-from-parallel on each face are independent identically distributed Gaussian random variables, the likelihood is that most manufacturing techniques would create correlated cube-errors across a batch of cubes and/or between faces on one cube.

The level of cube precision strongly affects the accelerometer calibration both the bias (Figure 4.13a) and the alignment matrix (Figure 4.13c). For the simple static calibration method (6-position set A) the proportional improvement with decreasing bias remains until the table's deviation from level (1° SD) becomes more significant, for the more advanced method (24-position set C) the proportional improvement continues to hold although the improvement starts to reduce and is slightly under 10× between 0.1 and 0.01°. An explanation for this phenomena is as follows, the calibration technique assumes that the cube is cubic, and so assumes that the specific force *in the nominal orientation frame* (see Section 4.3.3) is $\widetilde{\boldsymbol{f}^{\boldsymbol{ZU}}} = \begin{bmatrix} 0 & 0 & g \end{bmatrix}^T$. This assumed value is used to estimate the accelerometer biases, $\boldsymbol{b_a}$, and alignment matrix $\mathbf{A_a}$, using Equations 4.70 and 4.63 (for Technique 1). However, if we include *only* the error in this assumption due to the relevant face error frame (see Section 4.3.4)

$$\boldsymbol{f}^{\boldsymbol{ZU}} = \begin{bmatrix} \sin(\gamma_{ZU})\cos(\delta_{ZU})g \\ \cos(\gamma_{ZU})\sin(\delta_{ZU})g \\ \cos(\gamma_{ZU})\cos(\delta_{ZU})g \end{bmatrix} \tag{4.81}$$

where, γ_{ZU} and δ_{ZU} are the two angles (pitch and roll) by which the cube face 'ZU' deviates from cubic. Of course, both γ and δ for each of the six faces are independent identically distributed random variables (IIDRV), so for example Equation 4.63 becomes

$$\begin{aligned} \widehat{\boldsymbol{b_a}} = \frac{1}{6}\Bigg(& \begin{bmatrix} \cos(\gamma_{XU})\cos(\delta_{XU})g \\ \cos(\gamma_{XU})\sin(\delta_{XU})g \\ \sin(\gamma_{XU})\cos(\delta_{XU})g \end{bmatrix} + \boldsymbol{b_a} + \begin{bmatrix} -\cos(\gamma_{XD})\cos(\delta_{XD})g \\ -\cos(\gamma_{XD})\sin(\delta_{XD})g \\ -\sin(\gamma_{XD})\cos(\delta_{XD})g \end{bmatrix} + \boldsymbol{b_a} \\ & + \begin{bmatrix} \sin(\gamma_{YU})\cos(\delta_{YU})g \\ \cos(\gamma_{YU})\cos(\delta_{YU})g \\ \cos(\gamma_{YU})\sin(\delta_{YU})g \end{bmatrix} + \boldsymbol{b_a} + \begin{bmatrix} -\sin(\gamma_{YD})\cos(\delta_{YD})g \\ -\cos(\gamma_{YD})\cos(\delta_{YD})g \\ -\cos(\gamma_{YD})\sin(\delta_{YD})g \end{bmatrix} + \boldsymbol{b_a} \\ & + \begin{bmatrix} \sin(\gamma_{ZU})\cos(\delta_{ZU})g \\ \cos(\gamma_{ZU})\sin(\delta_{ZU})g \\ \cos(\gamma_{ZU})\cos(\delta_{ZU})g \end{bmatrix} + \boldsymbol{b_a} + \begin{bmatrix} -\sin(\gamma_{ZD})\cos(\delta_{ZD})g \\ -\cos(\gamma_{ZD})\sin(\delta_{ZD})g \\ -\cos(\gamma_{ZD})\cos(\delta_{ZD})g \end{bmatrix} + \boldsymbol{b_a} \Bigg). \end{aligned} \tag{4.82}$$

Thus the error in the estimate of the bias, *due to cube accuracy alone,* becomes

$$\widehat{\boldsymbol{b_a}} - \boldsymbol{b_a} = \frac{g}{6}\begin{bmatrix} \cos(\gamma_{XU})\cos(\delta_{XU}) - \cos(\gamma_{XD})\cos(\delta_{XD}) + \sin(\gamma_{YU})\cos(\delta_{YU}) - \sin(\gamma_{YD})\cos(\delta_{YD}) + \sin(\gamma_{ZU})\cos(\delta_{ZU}) - \sin(\gamma_{ZD})\cos(\delta_{ZD}) \\ \cos(\gamma_{XU})\sin(\delta_{XU}) - \cos(\gamma_{XD})\sin(\delta_{XD}) + \cos(\gamma_{YU})\cos(\delta_{YU}) - \cos(\gamma_{YD})\cos(\delta_{YD}) + \cos(\gamma_{ZU})\sin(\delta_{ZU}) - \cos(\gamma_{ZD})\sin(\delta_{ZD}) \\ \sin(\gamma_{XU})\cos(\delta_{XU}) - \sin(\gamma_{XD})\cos(\delta_{XD}) + \cos(\gamma_{YU})\sin(\delta_{YU}) - \cos(\gamma_{YD})\sin(\delta_{YD}) + \cos(\gamma_{ZU})\cos(\delta_{ZU}) - \cos(\gamma_{ZD})\cos(\delta_{ZD}) \end{bmatrix}$$

so each component of the error is a re-arrangement of the following, when each angle is re-

placed with a numbered angle θ_i and the standard small-angle approximations are applied,

$$\widehat{\boldsymbol{b_a}} - \boldsymbol{b_a} =$$
$$\frac{g}{6}\left(\sin(\theta_1)\cos(\theta_2) - \sin(\theta_3)\cos(\theta_4) + \cos(\theta_5)\sin(\theta_6) - \cos(\theta_7)\sin(\theta_8) + \cos(\theta_9)\cos(\theta_{10}) - \cos(\theta_{11})\cos(\theta_{12})\right)$$
$$\approx \frac{g}{6}\left(\theta_1(1-\frac{\theta_2^2}{2}) - \theta_3(1-\frac{\theta_4^2}{2}) + (1-\frac{\theta_5^2}{2})\theta_6 - (1-\frac{\theta_7^2}{2})\theta_8 + (1-\frac{\theta_9^2}{2})(1-\frac{\theta_{10}^2}{2}) - (1-\frac{\theta_{11}^2}{2})(1-\frac{\theta_{12}^2}{2})\right)$$
$$= \frac{g}{6}\left(\theta_1 - \frac{\theta_1\theta_2^2}{2} - \theta_3 + \frac{\theta_3\theta_4^2}{2} + \theta_6 - \frac{\theta_5^2\theta_6}{2} - \theta_8 + \frac{\theta_8\theta_7^2}{2} + 1 - \frac{\theta_9^2}{2} - \frac{\theta_{10}^2}{2} + \frac{\theta_9^2\theta_{10}^2}{4} - 1 + \frac{\theta_{11}^2}{2} + \frac{\theta_{12}^2}{2} - \frac{\theta_{11}^2\theta_{12}^2}{4}\right)$$
$$\approx \frac{g}{6}\left(\theta_1 - \theta_3 + \theta_6 - \theta_8\right)$$
$$\implies \mathbf{E}(\widehat{b_{a\,i}} - b_{a\,i}) \approx 0$$
$$\text{and } \mathbf{Var}(\widehat{b_{a\,i}} - b_{a\,i}) \approx 4\left(\frac{g}{6}\right)^2 \mathbf{Var}(\theta_i) \text{ as independent}$$
$$\implies (\widehat{b_{a\,i}} - b_{a\,i}) \sim \mathbf{N}\left(0, \left(\frac{g}{3}\sigma\right)^2\right), \text{ assuming } \theta_i \sim \mathbf{N}(0,\sigma^2) \quad (4.83)$$

For the Technique 2 bias estimation method, using only one pair of readings to estimate bias for each axis, this becomes,

$$\widehat{\boldsymbol{b_a}} - \boldsymbol{b_a} = \frac{g}{2}\left(\cos(\theta_9)\cos(\theta_{10}) - \cos(\theta_{11})\cos(\theta_{12})\right)$$
$$\approx \frac{g}{2}\left((1-\frac{\theta_9^2}{2})(1-\frac{\theta_{10}^2}{2}) - (1-\frac{\theta_{11}^2}{2})(1-\frac{\theta_{12}^2}{2})\right)$$
$$= \frac{g}{2}\left(1 - \frac{\theta_9^2}{2} - \frac{\theta_{10}^2}{2} + \frac{\theta_9^2\theta_{10}^2}{4} - 1 + \frac{\theta_{11}^2}{2} + \frac{\theta_{12}^2}{2} - \frac{\theta_{11}^2\theta_{12}^2}{4}\right)$$
$$\approx \frac{g}{4}\left(-\theta_9^2 - \theta_{10}^2 + \theta_{11}^2 + \theta_{12}^2\right)$$
$$\implies \mathbf{E}(\widehat{b_{a\,i}} - b_{a\,i}) \approx 0$$
$$\text{and } \mathbf{Var}(\widehat{b_{a\,i}} - b_{a\,i}) \approx \left(\frac{g}{4}\right)^2 \mathbf{Var}(\theta_i^2) = \left(\frac{g}{4}\right)^2 (3\sigma^4 - (\sigma^2)^2) = \left(\frac{g}{4}\right)^2 2\sigma^4$$
$$\text{as independent and Gaussian}$$
$$\implies (\widehat{b_{a\,i}} - b_{a\,i}) \sim \mathbf{N}\left(0, \left(\frac{g}{\sqrt{2}}\sigma^2\right)^2\right), \text{ assuming } \theta_i \sim \mathbf{N}(0,\sigma^2) \quad (4.84)$$

A comparison between Equations 4.83 and 4.84 shows the Technique 2 method of calculating $\widehat{\boldsymbol{b_a}}$ is much less sensitive to the misalignment of the cube's faces. This can be seen in Figure 4.13a where the Technique 2 method considerably outperforms the normal (Technique 1) method for most of the values. However, when the cube face error SD is 0.25° or lower the Technique 2 method performs worse. This is because it is more sensitive to most other error sources, this will be discussed in Section 4.8.1.4.

The statically determined gyroscope errors $\widehat{\boldsymbol{b_g}}$ and $\widehat{\mathbf{G_g^b}}$ are shown in Figures 4.13b and 4.13d, respectively. The estimation of gyroscope bias is much better than the minimum (1 °/s) standard required (magenta line) for stable Kalman filter integration (see Section 3.4.1), and for smaller cube face errors exceeds tactical performance (solid black line). $\widehat{\boldsymbol{b_g}}$ shows a direct improvement with the precision of the cube in a similar manner to that of accelerometer bias. However, this improvement is very small in comparison with the size of the starting error, so much that a log-scale is required to see it. This is simply because, determining the bias requires removing the component of the angular rate output due to specific force (i.e. $\mathbf{G_g}$), which unsurprisingly is small in comparison with the size of the bias

in a gyroscope, as the sensor is designed to minimise this effect, in contrast an accelerometer which is designed to be sensitive to specific force, so its bias is small in comparison to its sensitivity to specific force.

In the same way as discussed previously for the accelerometer bias the improvement with cube precision for the simpler (set A) method plateaus once the table's error from level (see Section 4.8.1.2) becomes more significant than that from the cube, and for the more complex (set C) method the improvement continues but plateaus at a greater angle than the accelerometer bias. An explanation for this is that the sensor noise for the gyroscope is becoming the dominant source of error, this will be discussed below, for example in Equation 4.100. For a similar reason the Technique 2 bias estimation technique is less accurate than Technique 1, due to its increased sensitivity to other sensor errors such as noise, which is a more significant error source for the gyroscope calibration, see Section 4.8.1.4.

In the same way there is also a direct improvement with cube precision in the gyroscope g-dependent error calibration. However, as can be seen in Figure 4.13d, the improvement levels off at a greater angular error than for the other statically determined errors. As the magnitude of the error being determined is small, the sensor noise becomes more significant than the error from the calibration cube imprecision at a greater cube angle, than $\widehat{\boldsymbol{b_g}}$. This is because the elements of $\widehat{\mathbf{G}_\mathbf{g}^\mathbf{b}}$ come from only two opposite faces, where the $\widehat{\boldsymbol{b_g}}$ uses all six faces, so there is $\approx \sqrt{3}$ more sensor noise on $\widehat{\mathbf{G}_\mathbf{g}^\mathbf{b}}$.

It should be noted that the difference between the algorithms using the six-position (set A) and 24-position (set C) manoeuvres is mostly due to better compensation for the table angle, see Equation 4.56 and 4.57, but there is also difference in their compensation for noise. So most of the improvement seen in Figure 4.13 is due to the compensation for the 1° SD table angle. However, the set A manoeuvres have only a single static measurement on each face, the set C manoeuvres have four so the static averaging time is 4× longer so the remaining effect of the noise on set C is half as great ($\frac{1}{\sqrt{4}}$) as for set A.

For the dynamic part of the calibration procedure, the cube's accuracy is also important. Its effect is shown in Figure 4.13e. For the less advanced procedure, set B, where the cube is rotated mid-air there is no *direct* effect on the manoeuvres' accuracy from the cube's accuracy. The indirect effects of the cube's accuracy, i.e. better estimation of the gyroscope biases and g-dependent error, do not result in substantial improvement in $\widehat{\mathbf{A}_\mathbf{g}^\mathbf{b}}$. Note that the set C advanced static method results are used in the computation of $\widehat{\mathbf{A}_\mathbf{g}^\mathbf{b}}$ so the indirect improvement continues throughout the range of values tested, but its effect is insignificant in comparison with the other errors.

In contrast, in the set D advanced dynamic calibration method the cube's accuracy has a direct effect *in addition to* the indirect improvement mentioned above because the accuracy of the axis of rotation in each rotation (of the six) is determined by how accurate the cube is. This means that the improvement continues (almost) linearly with improving cube accuracy throughout the range tested. It can be seen that the precision required from for a stable KF integration (see Chapter 3) of low level $\widehat{\mathbf{A}_\mathbf{g}}$ errors (see Table 3.3) is reached at around a cube angle of 0.25° This shows, in combination with Section 4.8.1.7, that the error in the rotation axis is the most significant source of error at default levels.

As can be seen in Figure 4.13e, the hybrid methods performs similarly with set A or C manoeuvres, as it reaches the limit of its accuracy which is determined by the IMU input parameters (see Section 4.8.2.3)

4.8.1.2 Table levelling error

In Figure 4.14 the effect of varying the SD of table's angle-from-level (hereafter table angle) is shown. As has been presented earlier there are three techniques used in the static part of the user calibration procedure here. These are the simple 6-position (set A), the advanced 24-position (set C) and a halfway-house 7-position, which applies only to the $\mathbf{A_a}$ and $\mathbf{G_g}$ matrices. As can be seen by comparing Equations 4.55 and 4.81, there is considerable similarity between the way that this table angle and the previously referred to cube angle affect the main assumption of the static calibration procedure, namely that the specific force due to gravity acts only vertically. Note that the default settings for these two error sources are both 1° SD, so at default level they should be approximately equally significant. Note also, the extreme values tested do not reflect a realistic range, a table which deviates from level by 10° SD is very clearly not level, and even a cheap spirit level should allow a surface level to within a degree to be found.

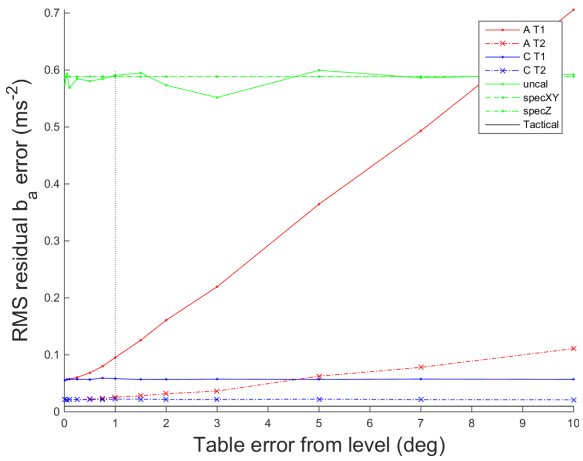
The effect of table angle on the statically determined errors is shown in Figures 4.14a, 4.14b, 4.14c and 4.14d. The simpler 6-position procedure is strongly affected by the table angle, showing a direct proportional improvement with smaller table angles until it plateaus when the cube angle becomes more significant. The advanced 24-position procedure on other hand appears completely unaffected by the table angle throughout the range tested, while the correction is not perfect it does reduce the error to much smaller than, for example, that from the cube error.

The fact that the 7-position method is not significantly more accurate than the 6-position method, despite the additional information, can be explained by how it differs from the 24-position method.

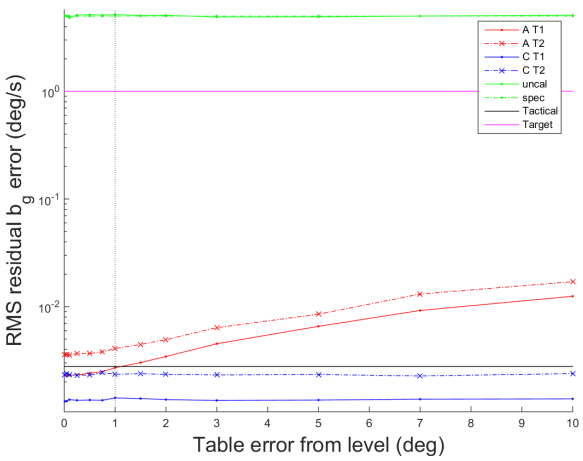
In the 24-position method, expected (nominal orientation frame) specific force after averaging over the 4 headings on each face is $\begin{bmatrix} 0 \\ 0 \\ \cos(\alpha)\cos(\beta)g \end{bmatrix}$, as seen in Equation 4.56. The algorithm then uses the estimated table angles to produce,

$$\frac{1}{\cos(\widehat{\alpha})\cos(\widehat{\beta})} \begin{bmatrix} 0 \\ 0 \\ \cos(\alpha)\cos(\beta)g \end{bmatrix} \approx \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix}. \tag{4.85}$$

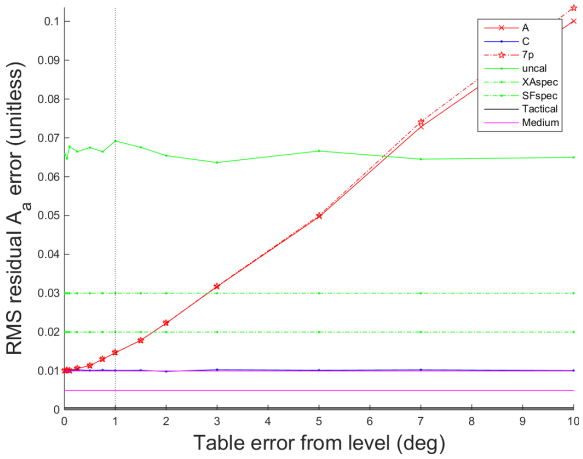
On the other hand the 7-position method, takes one reading on each face and one pair of measurements at 180° heading difference to estimate the two table angles. This table angle estimate is less accurate than the 24-position method, which uses 12 pairs rather than one, but at default settings this means only that it is 95% accurate at estimating the table angle, rather than 98%, as can be seen in Figure 4.14f as '1 pair' and '12 pair'. Even neglecting this estimation error and assuming $\widehat{\alpha} = \alpha$, $\widehat{\beta} = \beta$, there is only one heading recorded for



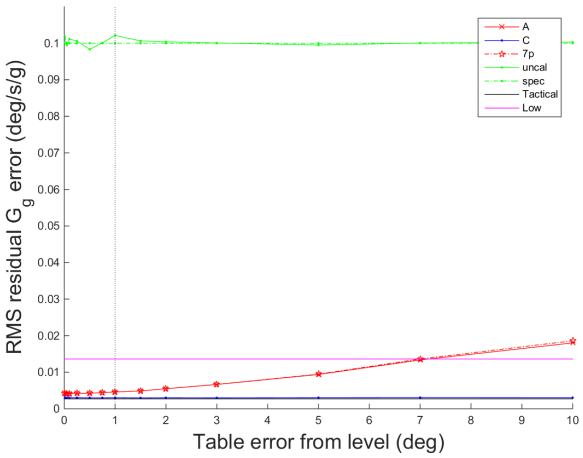
(a) Accelerometer bias



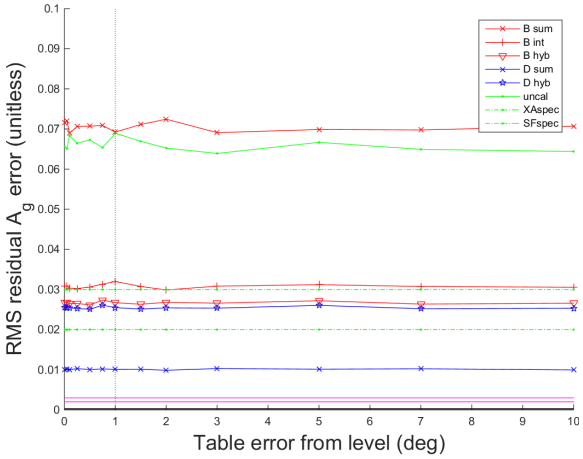
(b) Gyroscope bias



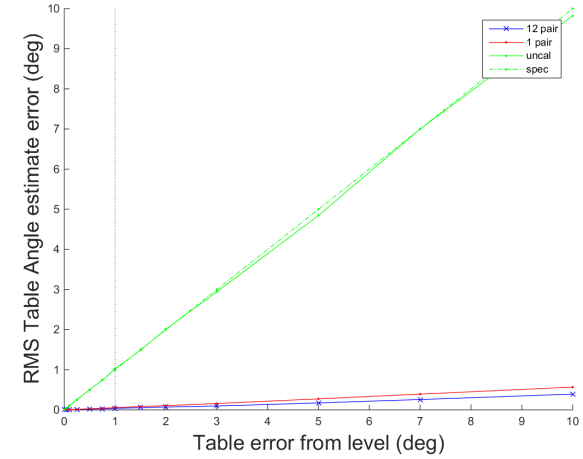
(c) Accelerometer alignment error



(d) Gyroscope g-dependent error



(e) Gyroscope alignment error



(f) Table angle estimate error

Figure 4.14: The effect of the table's deviation from level on the residual errors post-calibration.

each face, so the equivalent of Equation 4.85 is, without loss of generality (w.l.o.g.)

$$\frac{1}{\cos(\widehat{\alpha})\cos(\widehat{\beta})}\begin{bmatrix}\sin(\alpha)\cos(\beta)g\\ \cos(\alpha)\sin(\beta)g\\ \cos(\alpha)\cos(\beta)g\end{bmatrix} \approx \begin{bmatrix}\tan(\alpha)g\\ \tan(\beta)g\\ g\end{bmatrix} \approx \begin{bmatrix}\alpha-\frac{\alpha^3}{3}\\ \beta-\frac{\beta^3}{3}\\ 1\end{bmatrix} g \tag{4.86}$$

Note also that this is almost the same as the six-position equivalent where no table angle estimate is made

$$\begin{bmatrix}\sin(\alpha)\cos(\beta)g\\ \cos(\alpha)\sin(\beta)g\\ \cos(\alpha)\cos(\beta)g\end{bmatrix} \approx \begin{bmatrix}\alpha(1-\frac{\beta^2}{2})g\\ (1-\frac{\alpha^2}{2})\beta g\\ (1-\frac{\alpha^2}{2})(1-\frac{\beta^2}{2})g\end{bmatrix} = \begin{bmatrix}\alpha-\frac{\alpha\beta^2}{2})\\ \beta-\frac{\alpha^2\beta}{2}\\ 1-\frac{\alpha^2}{2}-\frac{\beta^2}{2}+\frac{\alpha^2\beta^2}{4}\end{bmatrix} g \tag{4.87}$$

As Equations 4.86 and 4.87 are almost equally bad approximations of $\begin{bmatrix}0 & 0 & g\end{bmatrix}^T$, is why the 7 position method has essentially identical performance to the 6-position method.

The table angle has no significant effect on the accuracy of the dynamic calibration can be seen in Figure 4.14e. The dynamic procedure does not assume that the table is level (only that it is flat) so the table becoming less level has no direct effect on the dynamic calibration procedure. Additionally because both the set B and set D procedures use the set C static measurements, which were shown above not to be affected by the table angle so there is no indirect effect on $\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}$, via the compensation for other systematic errors. This explains the result in Figure 4.14e.

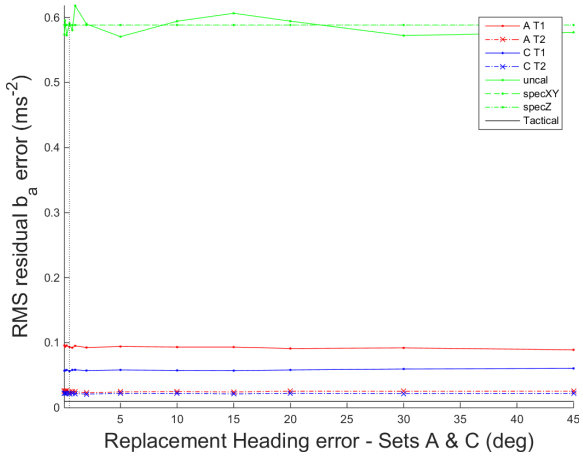
4.8.1.3 Replacement error in heading for the static part of the calibration procedure (sets A and C)

This section presents the results for varying the accuracy with which the calibration cube is placed on the table for each measurement taken during the static part of the calibration procedure. This is represented by a heading error, describing an accidental extra rotation about the table's z-axis. This is described in Section 4.3.5. This error changes the way that the table angles of the previous section affect the IMU.

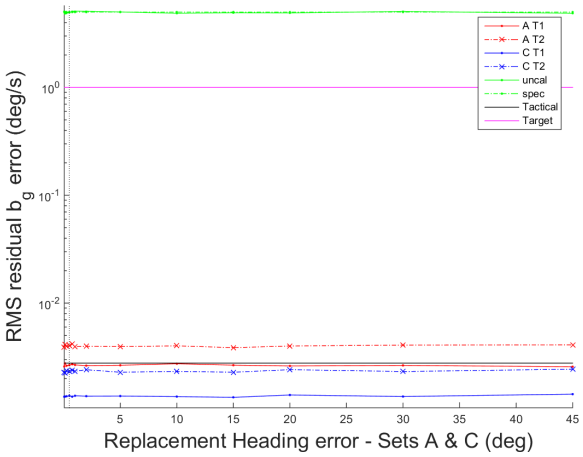
As the set A simple calibration method completely ignores the table angles it would be surprising if this had any effect. However, the set C advanced static calibration method relies on the 90° heading differences to get rid of the table angle effects, one might expect that this method would rely on the heading replacement errors being relatively small.

Figure 4.15 shows the large range of replacement error SDs tested from 0.05 to 45°, this represents range from 'more accurate than realistically possible' to 'barely paying any attention'. The set A technique is unsurprisingly unaffected, randomising the heading even appears to slightly help $\widehat{\boldsymbol{b_a}}$, by randomising the effect of the table errors. The fact that this is slightly helpful indicates that the choice of nominal orientations (see Section 4.3.3) are not optimal as putting a large random heading error reduces the size of the error. The implication is that an optimally selected set of six (of the 24 used for set C) orientations would be better than the arbitrarily chosen six in set A.

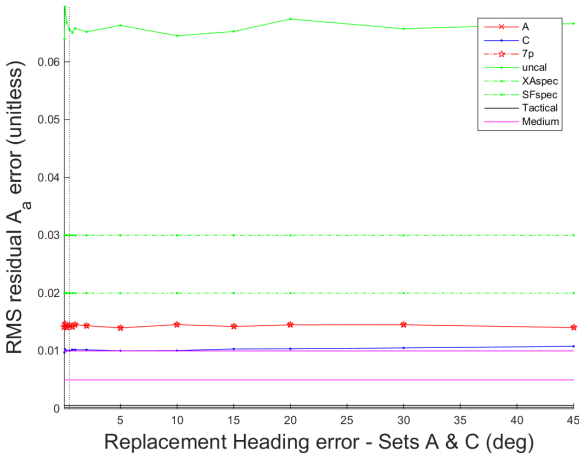
For the more advanced techniques, as might be expected the accuracy of the table angle estimates suffers with even a few degrees of heading replacement error (see Figure 4.15f and



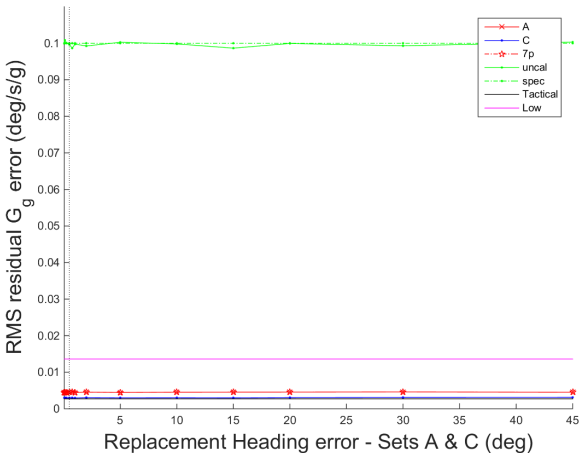
(a) Accelerometer bias



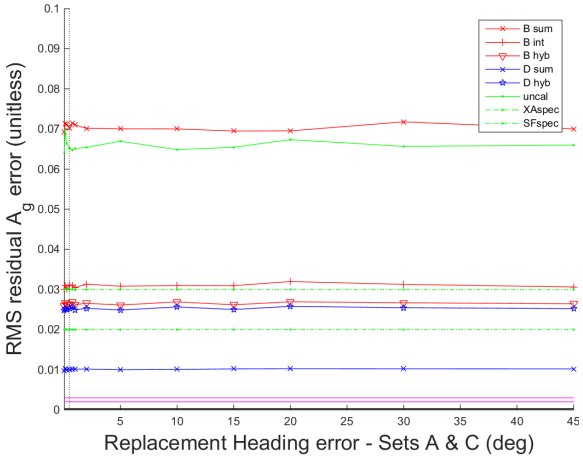
(b) Gyroscope bias



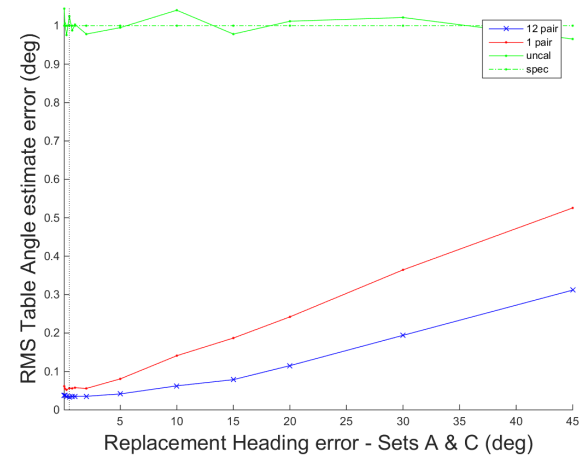
(c) Accelerometer alignment error



(d) Gyroscope g-dependent error



(e) Gyroscope alignment error



(f) Table angle estimate error

Figure 4.15: The effect of the accuracy of the replacement during the static part of the calibration on the residual errors post-calibration.

Replacement heading error sets A & C	0.05°	45°
Accelerometer Bias	ms^{-2}	
Set A T1 RMS	0.0957	0.0890
Set C T1 RMS	0.0569	0.0605
Set A T2 RMS	0.0254	0.0255
Set C T2 RMS	0.0221	0.0222
Gyroscope Bias	rad/s	
Set A T1 RMS	4.51e-5	4.49e-5
Set C T1 RMS	2.37e-5	2.50e-5
Set A T2 RMS	6.80e-5	7.16e-5
Set C T2 RMS	3.90e-5	4.26e-5
Gyro. G-dep. Bias	°/s / g	
Set A RMS	0.004479	0.004531
7 pos. RMS	0.004481	0.004532
Set C RMS	0.002987	0.003117
Accel. $\widehat{\mathbf{A}_\mathbf{a}}$ matrix	unitless	
Set A RMS	0.01412	0.01403
7 pos. RMS	0.01412	0.01404
Set C RMS	0.009747	0.0108
Table angle estimate	°	
1-pair RMS	0.06158	0.5256
12-pairs RMS	0.03704	0.3117

Table 4.1: Selected numerical results of varying the replacement error.

Table 4.1), the 1-pair estimate used for the 7-position calibration technique, unsurprisingly more so than the average of 12-pairs method used for the set C techniques. However, counter-intuitively, this doesn't appear to seriously affect the accuracy of the $\widehat{\boldsymbol{b_a}}$, $\widehat{\boldsymbol{b_g}}$, $\widehat{\mathbf{A}^\mathbf{b}_\mathbf{a}}$ or $\widehat{\mathbf{G}^\mathbf{b}_\mathbf{g}}$ using the set C method. The values at the extreme ends of the range tested are in Table 4.1. As can be seen the loss of accuracy is very small despite the vast range in replacement error. In Section 4.8.1.2 the improvement from the advanced technique was stated to be due more to the better cancellation of non-vertical measured components rather than the use of the table angle estimates. The following paragraphs explain this and also the cancellation's apparent robustness to high replacement errors.

Consider the specific force measurements in the nominal box frame, that is where all the face errors/cube angles are zero.

Consider also the 4 heading orientations of a particular side, for this example w.l.o.g. **ZU**. The orientations are **ZU1**, **ZU2**, **ZU3** and **ZU4**, see Figure 4.4. The table error angles α and β make the true specific force in the table frame (see Section 4.3.6,

$$\boldsymbol{f}^{\boldsymbol{ZU}} = \begin{bmatrix} \sin(\alpha)\cos(\beta)g \\ \cos(\alpha)\sin(\beta)g \\ \cos(\alpha)\cos(\beta)g \end{bmatrix}. \tag{4.88}$$

Additionally, if the matrix $\mathbf{R_h}$ represents rotation about the table-frame z-axis by h degrees, i.e. a reduced form of Equation 4.1,

$$\mathbf{R_h} = \mathbf{C_t^h} = \begin{bmatrix} \cos(h) & \sin(h) & 0 \\ -\sin(h) & \cos(h) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.89}$$

Then allow the IID heading replacement errors to be $h_1, \ldots, h_4$, to define the measurement used (see Equation 4.55 and 4.56), the average over the 4 set C measurements,

$$\begin{aligned} \overline{\boldsymbol{f}^{\boldsymbol{ZU}}} &= \frac{1}{4}\left(\boldsymbol{f}^{\boldsymbol{ZU1}} + \boldsymbol{f}^{\boldsymbol{ZU2}} + \boldsymbol{f}^{\boldsymbol{ZU3}} + \boldsymbol{f}^{\boldsymbol{ZU4}}\right) \\ &= \frac{1}{4}\left(\mathbf{R}_{\mathbf{h_1}}\boldsymbol{f}^{\boldsymbol{ZU1}} + \mathbf{R}_{(\mathbf{90^\circ+h_2})}\boldsymbol{f}^{\boldsymbol{ZU1}} + \mathbf{R}_{(\mathbf{180^\circ+h_3})}\boldsymbol{f}^{\boldsymbol{ZU1}} + \mathbf{R}_{(\mathbf{270^\circ+h_4})}\boldsymbol{f}^{\boldsymbol{ZU1}}\right) \\ &= \frac{1}{4}\left(\begin{bmatrix} \cos(h_1) & \sin(h_1) & 0 \\ -\sin(h_1) & \cos(h_1) & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} \cos(90^\circ+h_2) & \sin(90^\circ+h_2) & 0 \\ -\sin(90^\circ+h_2) & \cos(90^\circ+h_2) & 0 \\ 0 & 0 & 1 \end{bmatrix} + \right. \\ &\left. \begin{bmatrix} \cos(180^\circ+h_3) & \sin(180^\circ+h_3) & 0 \\ -\sin(180^\circ+h_3) & \cos(180^\circ+h_3) & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} \cos(270^\circ+h_4) & \sin(270^\circ+h_4) & 0 \\ -\sin(270^\circ+h_4) & \cos(270^\circ+h_4) & 0 \\ 0 & 0 & 1 \end{bmatrix}\right)\boldsymbol{f}^{\boldsymbol{ZU}} \end{aligned} \tag{4.90}$$

As,

$$\begin{aligned} &\sin(h_1) + \sin(90^\circ+h_2) + \sin(180^\circ+h_3) + \sin(270^\circ+h_4) \\ =&\sin(h_1)+\sin(90^\circ)\cos(h_2)+\cos(90^\circ)\sin(h_2)+\sin(180^\circ)\cos(h_3)+\cos(180^\circ)\sin(h_3)+\sin(270^\circ)\cos(h_4)+\cos(270^\circ)\sin(h_4) \\ &= \sin(h_1) + \cos(h_2) - \sin(h_3) - \cos(h_4) \\ &\approx h_1 - \frac{h_1^3}{6} + \ldots + 1 - \frac{h_2^2}{2} + \ldots - h_3 + \frac{h_3^3}{6} - \ldots - 1 + \frac{h_4^2}{2} - \ldots \approx h_1 - h_3 \end{aligned} \tag{4.91}$$

$$\begin{aligned} &\cos(h_1) + \cos(90^\circ+h_2) + \cos(180^\circ+h_3) + \cos(270^\circ+h_4) \\ =&\cos(h_1)+\cos(90^\circ)\cos(h_2)-\sin(90^\circ)\sin(h_2)+\cos(180^\circ)\cos(h_3)-\sin(180^\circ)\sin(h_4)+\cos(270^\circ)\cos(h_4)-\sin(270^\circ)\sin(h_4) \\ &= \cos(h_1) - \sin(h_2) - \cos(h_3) + \sin(h_4) \approx h_4 - h_2 \end{aligned} \tag{4.92}$$

By substituting these into 4.90,

$$\begin{aligned} \overline{\boldsymbol{f}^{\boldsymbol{ZU}}} =& \frac{1}{4}\begin{bmatrix} \cos(h_1)-\sin(h_2)-\cos(h_3)+\sin(h_4) & \sin(h_1)+\cos(h_2)-\sin(h_3)-\cos(h_4) & 0 \\ -\sin(h_1)-\cos(h_2)+\sin(h_3)+\cos(h_4) & \cos(h_1)-\sin(h_2)-\cos(h_3)+\sin(h_4) & 0 \\ 0 & 0 & 4 \end{bmatrix}\begin{bmatrix} \sin(\alpha)\cos(\beta)g \\ \cos(\alpha)\sin(\beta)g \\ \cos(\alpha)\cos(\beta)g \end{bmatrix} \\ \approx& \frac{1}{4}\begin{bmatrix} h_1-h_3 & h_2-h_4 & 0 \\ -h_2+h_4 & h_1-h_3 & 0 \\ 0 & 0 & 4 \end{bmatrix}\begin{bmatrix} \sin(\alpha)\cos(\beta)g \\ \cos(\alpha)\sin(\beta)g \\ \cos(\alpha)\cos(\beta)g \end{bmatrix} \\ =& \underbrace{\begin{bmatrix} 0 \\ 0 \\ \cos(\alpha)\cos(\beta)g \end{bmatrix}}_{\text{expected result}} + \underbrace{\frac{1}{4}\begin{bmatrix} h_1-h_3 \\ -h_2+h_4 \\ 0 \end{bmatrix}\sin(\alpha)\cos(\beta)g + \frac{1}{4}\begin{bmatrix} h_2-h_4 \\ h_1-h_3 \\ 0 \end{bmatrix}\cos(\alpha)\sin(\beta)g}_{\text{Error due to heading replacement}} \end{aligned} \tag{4.93}$$

In the Technique 2 method only the third ('z') column of the ZU or ZD is used and so there is no effect of the heading replacement error on the Technique 2 bias estimation method. For the same reason the diagonal entries of $\widehat{\mathbf{G_g}}$ and $\widehat{\mathbf{A_a}}$ are also completely unaffected by the heading replacement error.

The Technique 1 bias estimation method uses the mean of the six faces' measurements. Thus if $\varepsilon_{b_a|_{h_i}}$ is the error in the measurement due to the heading replacement error. That is, the error in Equation 4.63 coming from the second part of Equation 4.93. Note that, as there are 24 replacements, there are 24 IID h_i's.

$$\varepsilon_{b_a|_{h_i}} = \frac{1}{6}\left(\frac{1}{4}\begin{bmatrix} h_1 - h_3 + \ldots_{\text{(8 terms)}} \\ -h_2 + h_4 - \ldots_{\text{(8 terms)}} \\ h_5 - h_7 + \ldots_{\text{(8 terms)}} \end{bmatrix} \sin(\alpha)\cos(\beta)g + \frac{1}{4}\begin{bmatrix} h_2 - h_4 + \ldots_{\text{(8 terms)}} \\ h_1 - h_3 + \ldots_{\text{(8 terms)}} \\ h_6 - h_8 + \ldots_{\text{(8 terms)}} \end{bmatrix} \cos(\alpha)\sin(\beta)g\right) \quad (4.94)$$

So, if $h_i \sim (0, \sigma_h)$ then for all rows,

$$\begin{aligned}\mathbf{Var}\left(\varepsilon_{b_a|_{h_i}}\right) &= \left(\frac{\sin(\alpha)\cos(\beta)g}{24}\right)^2 8{\sigma_h}^2 + \left(\frac{\cos(\alpha)\sin(\beta)g}{24}\right)^2 8{\sigma_h}^2 \\ &= \left(g\sqrt{\sin^2(\alpha)\cos^2(\beta) + \cos^2(\alpha)\sin^2(\beta)}\frac{\sqrt{2}}{12}\sigma_h\right)^2 \end{aligned} \quad (4.95)$$

If α and β are taken to be $1^\circ \approx 0.0175$rad (default SD so correct order of magnitude), then

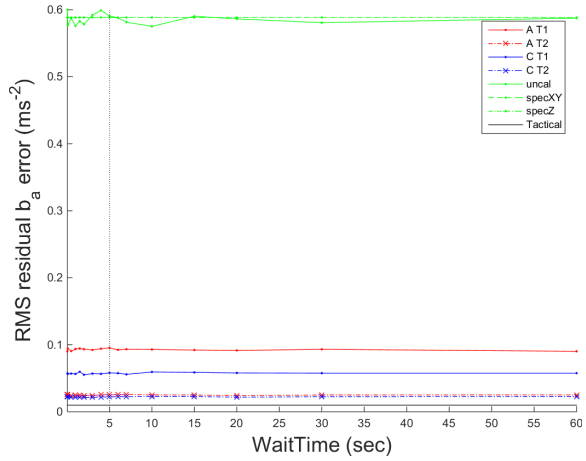
$$\begin{aligned}\mathbf{SD}\left(\varepsilon_{b_a|_{h_i}}\right) &= g\sqrt{\sin^2(\alpha)\cos^2(\beta) + \cos^2(\alpha)\sin^2(\beta)}\frac{\sqrt{2}}{12}\sigma_h \\ &= 0.0247\frac{g\sqrt{2}}{12}\sigma_h = 0.0285\ \sigma_h\,\text{ms}^{-2} \end{aligned} \quad (4.96)$$

Note that σ_h must be expressed in radians for the above to work so for $5^\circ \approx 0.0873$rad, then $\mathbf{SD}\left(\varepsilon_{b_a|_{h_i}}\right) = 0.0025\,\text{ms}^{-2}$, compared to the total RMS error for accelerometer bias seen in Figure 4.15a of around 0.05ms^{-2}; this is clearly not a significant error source.

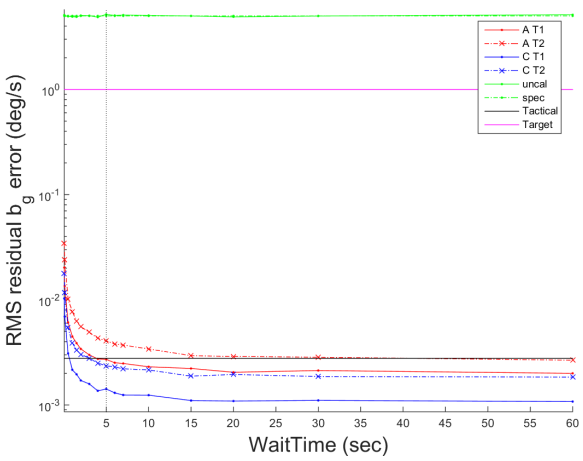
As this error source only applies to the static part of the procedure, any effect on the dynamically determined $\widehat{\mathbf{A}^{\mathbf{b}}_{\mathbf{g}}}$ term is indirect, and because the effect of this error is so small, its indirect effect is not visible in Figure 4.15e.

4.8.1.4 Static averaging time

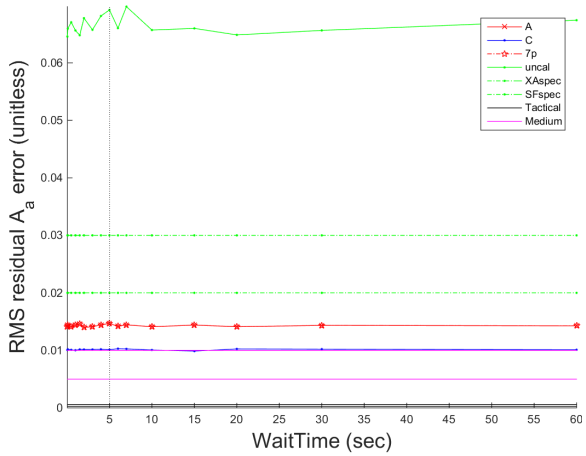
Waiting longer on each face of the cube, gives a longer averaging time, which reduces the effect of the sensor noise on the eventual results, for both the specific force and angular rate. In Figure 4.16, results with a wide range of averaging times from 0.05 to 60 seconds are presented. However, the extreme values are not realistic because only waiting for very small amounts of time is not practical, and very long waiting times would mean that the modelling assumption that the systematic errors are fixed over the whole experiment may not be valid (see Chapter 5). A more realistic range would be 1 to 10 seconds, but the greater range is presented because this is more instructive of the error behaviour.



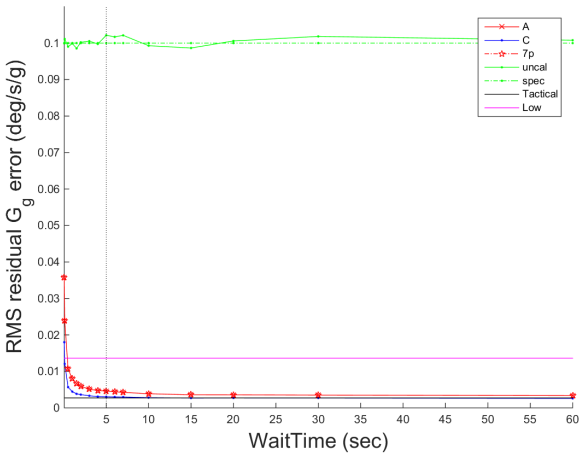
(a) Accelerometer bias



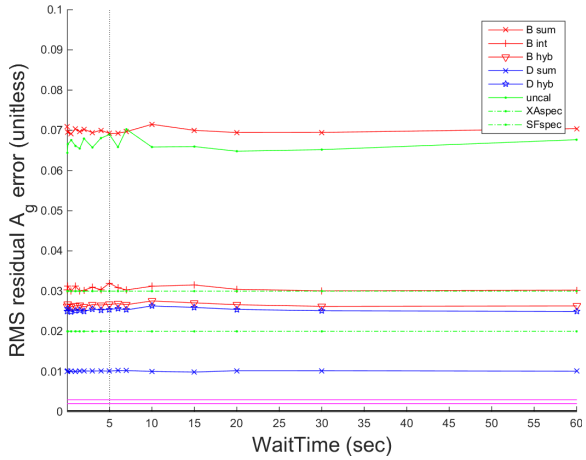
(b) Gyroscope bias



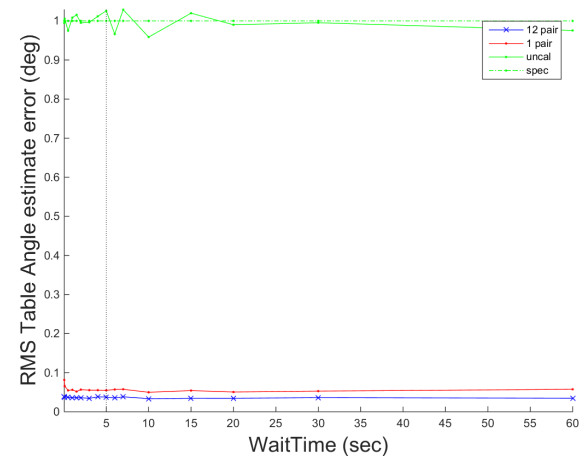
(c) Accelerometer alignment error



(d) Gyroscope g-dependent error



(e) Gyroscope alignment error



(f) Table angle estimate error

Figure 4.16: The effect of the static averaging time on the residual errors post-calibration.

The assumption that over a sufficient time interval the noise term is insignificant and may be ignored that takes Equation 4.47 to 4.49 and 4.48 to 4.50. That is, assuming the average of the noise over the sampling interval for both the accelerometer $\overline{\boldsymbol{w_a}} \approx 0$ and gyroscope $\overline{\boldsymbol{w_g}} \approx 0$. This approximation can be expanded as follows. The average of the noise over the sampling interval,

$$\overline{\boldsymbol{w_a}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{w_a}(i) \implies$$

$$\mathbf{E}(\overline{\boldsymbol{w_a}}) = \frac{1}{n}\left(n\cancelto{0}{\mathbf{E}(\boldsymbol{w_a})}\right) = 0$$

$$\text{and } \mathbf{E}(\overline{\boldsymbol{w_a}}^2) = \frac{\sigma^2_{\boldsymbol{w_a}}}{n}$$

$$\implies \overline{\boldsymbol{w_a}} \sim \mathbf{N}\left(0, \left(\frac{\sigma_{\boldsymbol{w_a}}}{\sqrt{n}}\right)^2\right) \quad (4.97)$$

where $\sigma_{\boldsymbol{w_a}}$ is the standard deviation of the noise process at each of the n epochs. Given that these two quantities are determined from the noise PSD (S_a), the output data rate (R, Hz) and the averaging time (T_{av}, sec). The SD of the effect of the average noise ($\sigma_{\overline{\boldsymbol{w_a}}}$) is

$$\sigma_{\overline{\boldsymbol{w_a}}} = \frac{\sigma_{\boldsymbol{w_a}}}{\sqrt{n}} = \frac{\sqrt{S_a}\sqrt{R}}{\sqrt{T_{av}R}} = \frac{\sqrt{S_a}}{\sqrt{T_{av}}} \quad (4.98)$$

For default settings, i.e. 5 second averaging, this becomes

$$\sigma_{\overline{\boldsymbol{w_a}}} = \frac{\sqrt{S_a}}{\sqrt{T_{av}}} = \frac{300\,\mu\text{g}/\sqrt{Hz}}{\sqrt{(5\text{ sec})}} = \frac{0.0029\text{ ms}^{-2}/\sqrt{Hz}}{\sqrt{(5\text{ sec})}} = 0.0013\,\text{ms}^{-2} \quad (4.99)$$

$$\sigma_{\overline{\boldsymbol{w_g}}} = \frac{\sqrt{S_g}}{\sqrt{T_{av}}} = \frac{0.01\,^\circ/\text{s}/\sqrt{\text{Hz}}}{\sqrt{(5\text{ sec})}} = \frac{1.7\times10^{-4}\text{ rad}/\sqrt{Hz}}{\sqrt{(5\text{ sec})}} = 7.8\times10^{-5}\text{ rad/s} = 0.0045\,^\circ/\text{s} \quad (4.100)$$

Comparing the number from Equation 4.99 with the axes in Figures 4.16a and 4.16c it becomes clear why the contribution of the noise these errors is insignificant.

Consider also that for the measurements of bias in set A, 6 measurements are averaged for Technique 1 and 2 measurements in the Technique 2 algorithm, and for the set C measurements each of the 3 bias estimates in Technique 1 is the average of 24 measurements or 8 for Technique 2. So the actual contribution to the bias is between $\sqrt{2} \approx 1.4$ and $\sqrt{24} \approx 4.9$ times smaller than calculated in Equation 4.99. For a similar reason the $\widehat{\mathbf{A}_\mathbf{a}}$ and $\widehat{\mathbf{G}_\mathbf{g}}$ are also either $\sqrt{2} \approx 1.4$ or $\sqrt{8} \approx 2.8$ times less affected by the noise than calculated. Similarly the table angles which are estimated using accelerometer measurements are barely affected by the averaging time (Figure 4.16f).

For the gyroscope static estimates partly because the effect of the cube inaccuracy is so much (numerically) lower, the contribution to the overall error from the noise is more significant. The value calculated in Equation 4.100 is much closer to the residual error seen in Figures 4.16b and 4.16d. Thus reducing or lengthening the averaging time has more of an effect, even considering the reduction due to averaging several measurements discussed above.

This input only effects the static part of the calibration procedure, so any effect on the dynamically determined $\widehat{\mathbf{A}_\mathbf{g}}$ is indirect. As the direct effect is very small, there is no visible indirect effect in Figure 4.16e.

4.8.1.5 Replacement heading error for the dynamic part of the calibration procedure (sets B and D)

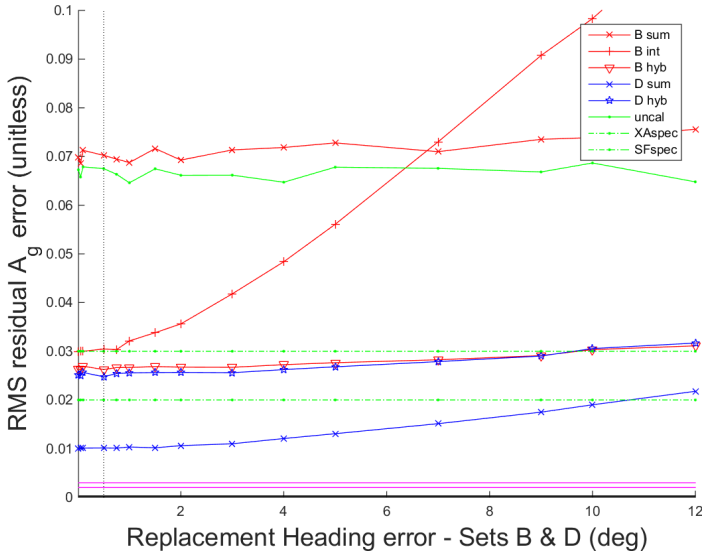


Figure 4.17: The effect of the accuracy of the replacement during the dynamic part of the calibration on the residual gyroscope alignment errors post-calibration.

The most fundamental assumption of the dynamic calibration method is that it is a zero-attitude update. In other words that after each manoeuvre the IMU is replaced *at the same attitude*. The position is not used, but in a real system the position of the cube before and after must be within a few millimetres as the 'table' will not be a perfect plane, and so a zero-position update is needed to make the zero-attitude update valid. However, replacing the IMU on the table after a manoeuvre can create an angular error about the axis normal to the table; this is described in Section 4.3.5. For brevity this will be described as a heading error, despite it being about an axis normal to the table rather than the gravity vector (see Section 4.3.6) In the default settings this heading error is small, with an SD of 0.5°, this is because the assumption is that the IMU is placed in a corner in the manner described in Section 4.5.2.

The greater the heading error, the less valid the zero-attitude update is. This can be seen in Figure 4.17, where the heading errors in sets B and D are varied together. The set D techniques are affected much less, primarily because there are six manoeuvres in set D rather than the 3 in set B, so the effect of heading replacement errors is reduced by the fact that there calculated correction is averaged between two manoeuvres. Additionally the effect on the hybrid method is less strong because only the 3 diagonal entries in the $\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}$ matrix are estimated from the manoeuvres, the others being taken from $\widehat{\mathbf{A}_{\mathbf{a}}^{\mathbf{b}}}$.

It can be concluded that the other errors in the dynamic calibration are large enough that diminishing returns in getting the replacement angle correct apply at around 2°.

Note that, because this error applies in the table-frame z-axis, this error effects sets B and D slightly differently. Its effect on set B is seen mostly in the bottom row, whereas in set D as it applies in the direction of rotation it mostly applies to the diagonal entries of the matrix.

4.8.1.6 Rotation time

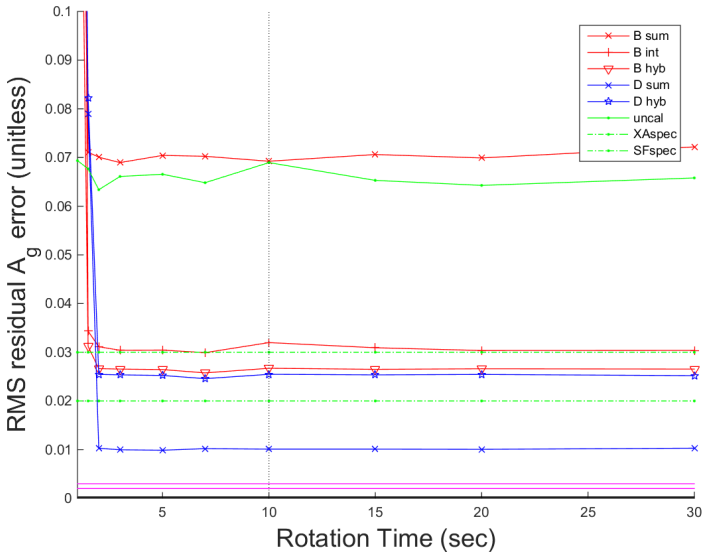


Figure 4.18: The effect of the rotation time during the dynamic part of the calibration on the residual gyroscope alignment errors post-calibration.

Figure 4.18 shows the effect of the time taken to rotate the sensors during the dynamic tests, henceforth the rotation time.

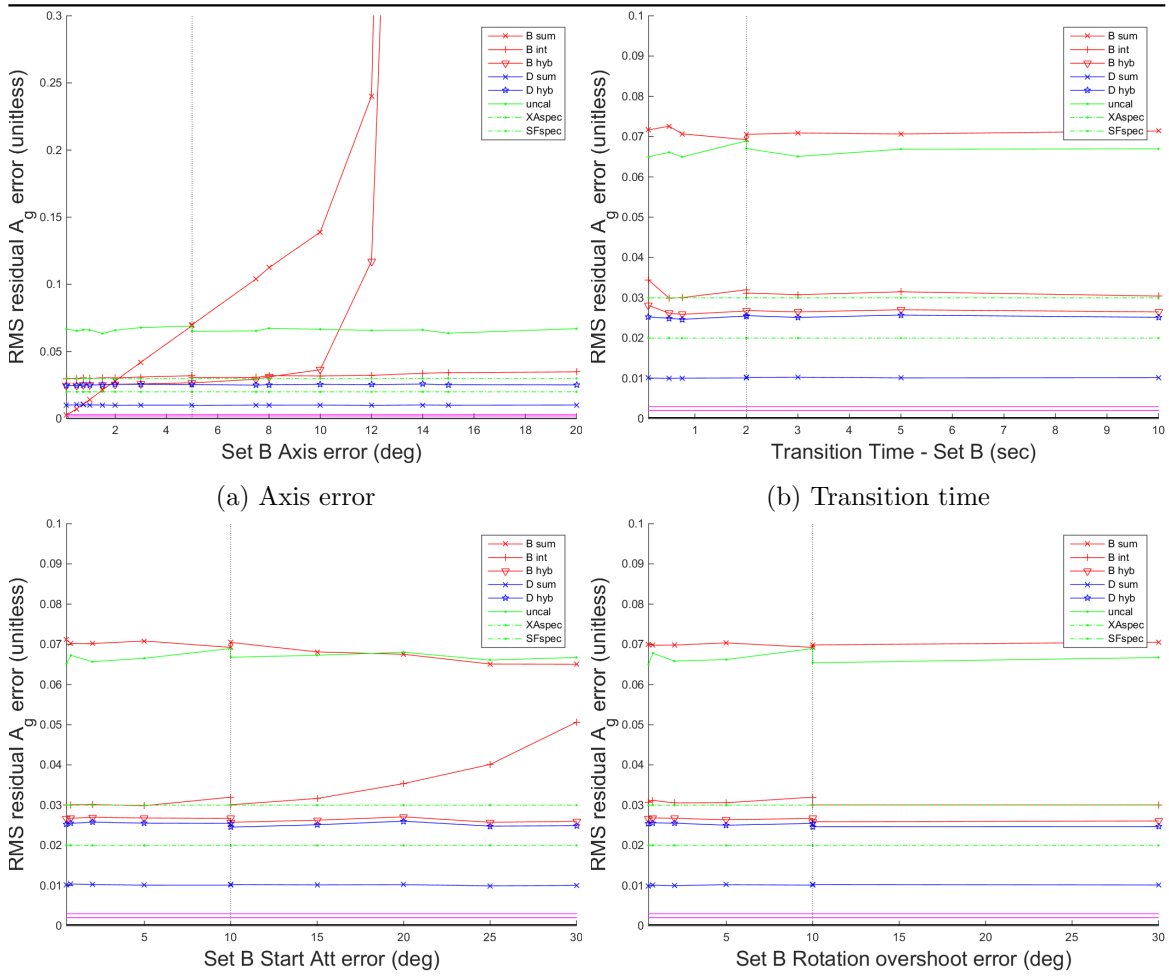
When the rotation times are less than two seconds the performance is extremely poor. This is because the sensors are rotating at speeds in excess of the maximum that they can measure (250 °/s), causing sensor clipping. For times longer than 2 seconds there is almost no visible effect in the graph.

Theoretically a longer rotation time should perform worse than a shorter one, in contrast with the longer static averaging times in Section 4.8.1.4. This is because for longer integration times more sensor noise is integrated, this means the accuracy gets worse in proportion to $\sqrt{n}$, rather than better by a factor of $\frac{1}{\sqrt{n}}$. However the contribution of sensor noise to the overall error is still small in comparison to the other errors, so as can be seen in Figure 4.18 there is no dramatic improvement over the range of times that might be practical.

It is also worthwhile noting that while theoretically faster rotations result in better calibrations, in reality a procedure that is carried out quickly will make it difficult to retain the same level of rotational accuracy, and as rotation speeds are not likely to be as uniform as simulated, the chance of clipping is higher. Therefore a better approach would be to repeat the calibration manoeuvres and take the eventual average, this would reduce the effect of integrated sensor noise as well as several other errors, e.g. the heading replacement (Section 4.8.1.5 and those in Section 4.8.1.7.

4.8.1.7 Other dynamic procedure errors

The procedure for set B as shown in Figure 4.3, is sensitive to other errors sources in addition to the 'rotation time' and 'heading replacement error' it shares with set D. From its start position it is lifted up in the air, to a orientation which is intended to be aligned with the



(a) Axis error

(b) Transition time

(c) Starting attitude error

(d) Rotation overshoot error

Figure 4.19: The effect of the various set B specific errors of the calibration on the residual Gyroscope alignment error post-calibration.

local navigation frame. This could take a variable amount of time (Figure 4.19b), which means that more noise must be integrated. The mid-air starting attitude alignment will not be perfect (Figure 4.19c). At this point the box should be rotated about one of the cardinal axes by 360° and then returned to the starting position, so that the data from that period can be averaged. It is difficult to rotate about an axis in mid-air so that will not be perfectly achieved (Figure 4.19a) and the user may over- or under-shoot 360° (Figure 4.19d), which means that the overshoot must be reversed during the transition back to the start, which exacerbates the order-of-rotations problem that some of the integration methods have (see Section 4.5.2).

As these inputs only affect the set B dynamic method (red lines) there is no effect on the set D results, so only the blue lines in Figure 4.19 are only shown for comparative purposes.

The figures illustrate that the axis-error (Figure 4.19a) is the most critical of these parameters. The accuracy obtained at the default level, which is already a reasonably difficult level of accuracy to obtain by hand, improves almost directly linearly as the axis error is reduced. Note that, the set D equivalent of the axis error is from the cube face error, so when the axis error is 1° they are approximately equivalent. At 1° axis error level the set B method

is approximately 40% worse than the set D, which is about the level that might be expected given the fact that set D has the average of two (essentially independent) measurements in comparison with set Bs one. When the axis error gets much above 10°, the method breaks completely, due to feedback in the iterative procedure.

The effect on $\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}$ of the axis error is that a correction that should be made to one entry in the $\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}$-matrix is in fact made to another element in the same row, because motion which was assumed to be about one of the body-frame axes was not. As $\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}$ is calculated in an iterative process, these wrongly applied corrections then feed back to the next iteration, which makes the estimate progressively worse leading to the very poor results seen in Figure 4.19a at above 10°. It is clear that this is the dominant error source, being much greater than the error from a poor zero-attitude update (Figure 4.17) or additional gyroscope noise (Figures 4.18, 4.19b, 4.29c etc.). It can also be seen that the axis error required for the set B manoeuvres to be achieve target accuracy (magenta line) is much less than 0.5°, which is not realistically possible with mid-air rotations, but using a guide like the table's surface (as in set D), might make it realistic.

The rotation overshoot and transition time, appear to have little effect in the presence of the large axis error, except where the transition times are so short that clipping occurs (left-hand-side of Figure 4.19b). However, the starting attitude error, does appear to have some noticeable effect on the integrated method particularly (labelled 'B int').

4.8.1.8 Other procedural errors

Other error sources, which come from the 'TESTspec' part of the inputs were also varied. Some of these, such as the transition time between positions in the static part, were tested to verify that they had no effect on the results, which they should not and did not have. These are not presented for reasons of space. The results for error specifications which generate the mounting angle errors and thus generate each run's $\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}$, are not presented here but in Section 4.8.2.3 where they fit more naturally.

4.8.2 The effect of the IMU specification

The IMU performance specifications which are included in the Monte Carlo simulations, are included as part of 'IMUspec' (see Section 4.6.1). This includes the starting or uncalibrated values of the systematic errors which are estimated by the calibration procedure, and also the remaining IMU errors, including the stochastic terms and higher-order sensor errors, which might be expected to affect the accuracy of the procedure. The default values for these inputs are based on a MPU9250 IMU's specification [23], and are assumed to be drawn from zero-mean Gaussian distributions.

The direct effect of most of these input parameters should be fairly predictable, often a reasonable estimate can be made by means of a simple calculation. Note that, the results presented in this section represent the same calibration procedure being applied to IMUs of different specification and that the procedure in question (with default 'TESTspec' inputs)

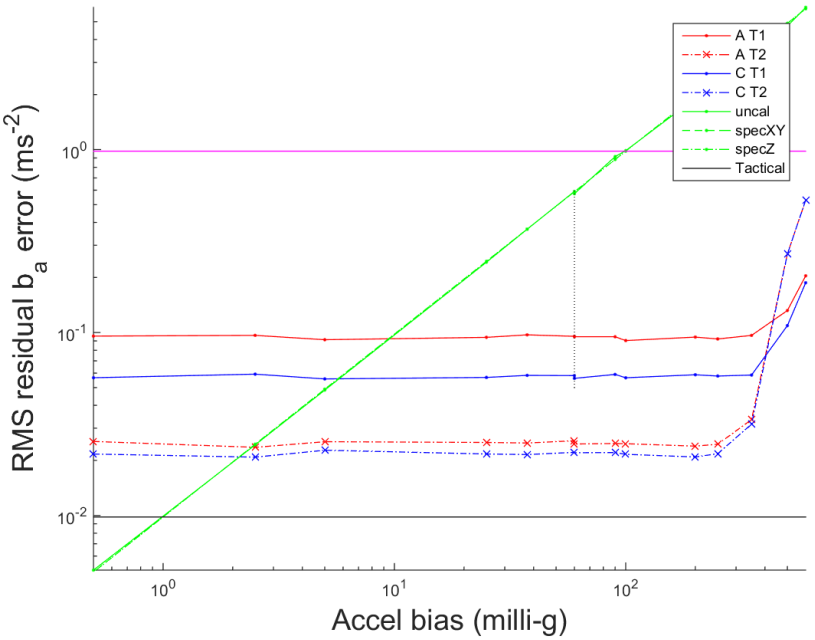


Figure 4.20: The effect of the pre-calibration accelerometer bias on the accelerometer bias remaining post-calibration.

is dominated by errors representing the cube's deviation from being cubic, as was shown in Section 4.8.1.

As the individual effects of some of these sensor errors are very small on their own, in order to make the effect of the error visible, a range of possible inputs for each one has been tested which is, in some cases, several orders of magnitude worse than might exist in real sensors. This can lead to some odd effects, for example if a ±2g-range accelerometer's bias reaches 1g, the sensor will crop sensor output during static tests. Odd effects should be expected when things are taken to extremes, and these can be safely ignored when they are far enough from the realistic range of values. It should be noted that the MPU9250 is a consumer-grade IMU, and thus realistic range of values will generally be the default input or lower.

This section examines the effect of each of the 'IMUspec' inputs in turn starting with the errors that are being calibrated.

4.8.2.1 Sensor biases

The effect of varying the input accelerometer and gyroscope biases on the respective final post-calibration residual bias is shown in Figures 4.20 and 4.21. It is shown that there is no effect from these biases being several orders of magnitude smaller or greater, at least until the point where the bias is so large that the sensor clips under normal use (e.g. rightmost points in Figure 4.20). As any indirect effect depends on the residual sensor biases, there is no significant effect on the other calibration outputs, so they are not shown. These results show that the starting level of the bias has no effect on the accuracy of the resulting calibration, which is instead governed by different factors such as the level of sensor noise and the accuracy of the procedure.

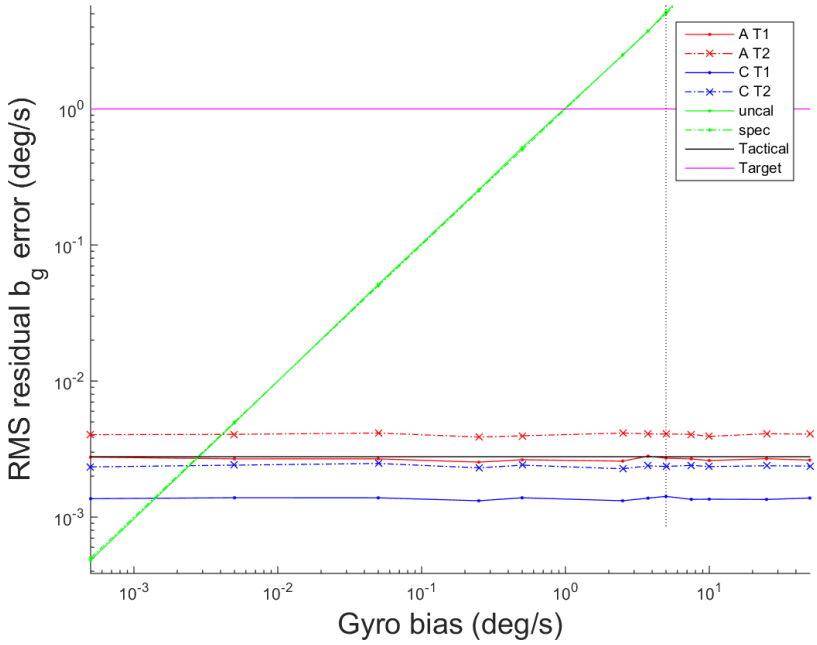
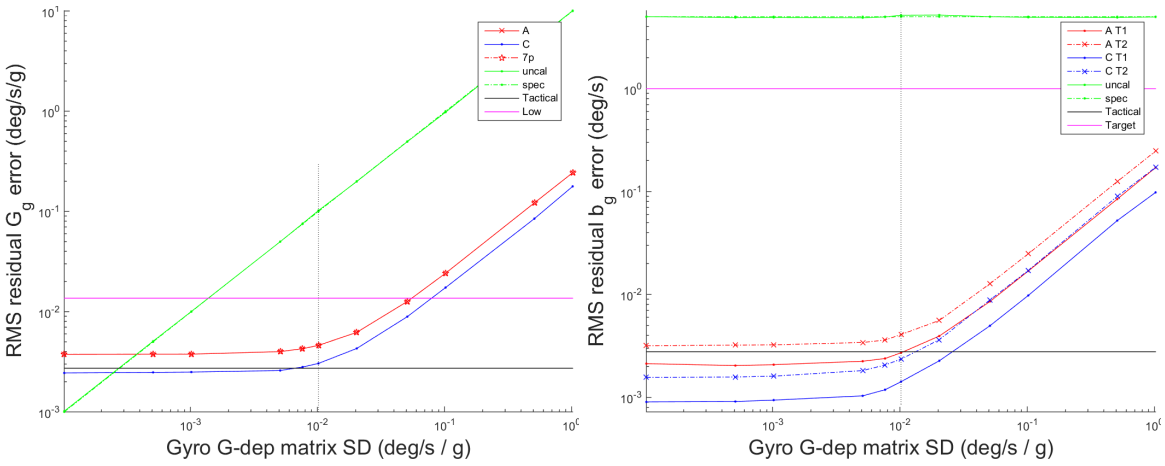


Figure 4.21: The effect of the pre-calibration gyroscope bias on the gyroscope bias remaining post-calibration.

4.8.2.2 Gyroscope g-dependent error



(a) Gyroscope g-dependent error

(b) Gyroscope bias

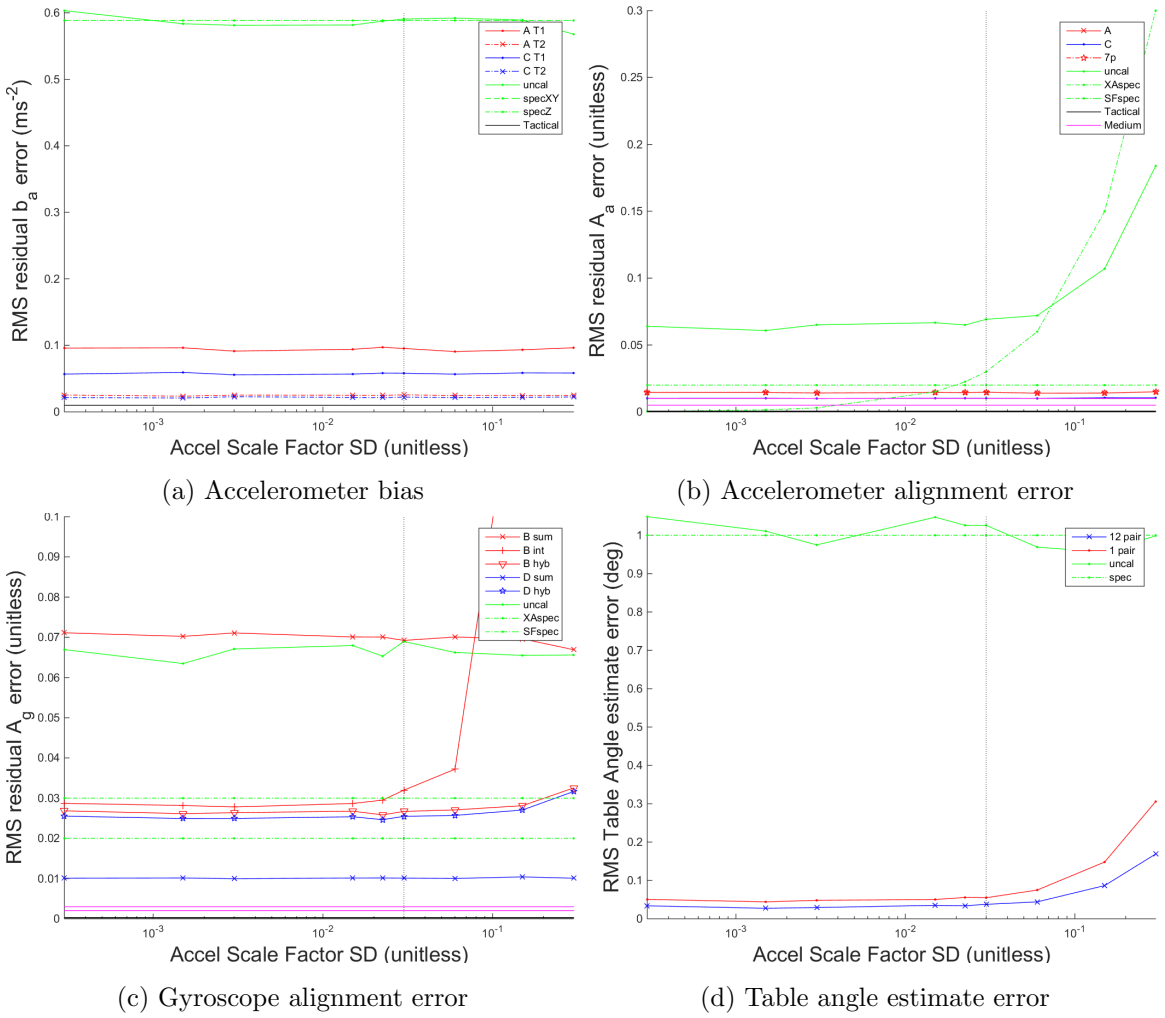
Figure 4.22: The effect of the pre-calibration gyroscope g-dependent bias on the errors remaining post-calibration.

The effect of varying the gyroscope g-dependent error is shown in Figure 4.22; note that the levels of error shown on the right-hand-side of the graph are far greater than is realistic, even the default level of 0.1 °/s/g is conservative (experimentally determined values were around 10× smaller, see Chapter 5). However, from Figure 4.22a it can be seen that the algorithm reduces the g-dependent error to ≈ 2% of the starting error or down to a particular level ($\approx 2 \times 10^{-3}$ °/s / g) whichever is the greater, for larger and smaller starting errors respectively. It would be reasonable to propose that the proportional behaviour is due to the angular error of the cube and/or table, and the fixed floor is due to the averaging time and gyroscope noise level. This is supported by the fact that 2% deviation is approximately

equivalent to 1 degree of angular misalignment[9] and by Equation 4.100. The calculation of $\widehat{\boldsymbol{b_g}}$ uses the same measurements ($\widetilde{\widetilde{\boldsymbol{\omega^{XU}}}}$, etc.) as that for $\widehat{\mathbf{G}^{\mathbf{b}}_{\mathbf{g}}}$, see Equations 4.65 and 4.72. As such it is not surprising that when there is a noticeable effect on $\widehat{\mathbf{G}^{\mathbf{b}}_{\mathbf{g}}}$, there is also a similar effect on $\widehat{\boldsymbol{b_g}}$, as is shown in Figure 4.22b. Additionally, both errors remain well within the target values (magenta line) even with 5× greater than default starting error.

However, this level of increased residual gyroscope bias and g-dependent bias is still too small to make any significant difference in the dynamic calibration. The gyroscope measurements do not effect the accelerometer calibration, so these are not shown.

4.8.2.3 Accelerometer and gyroscope alignment and IMU mounting errors



(a) Accelerometer bias

(b) Accelerometer alignment error

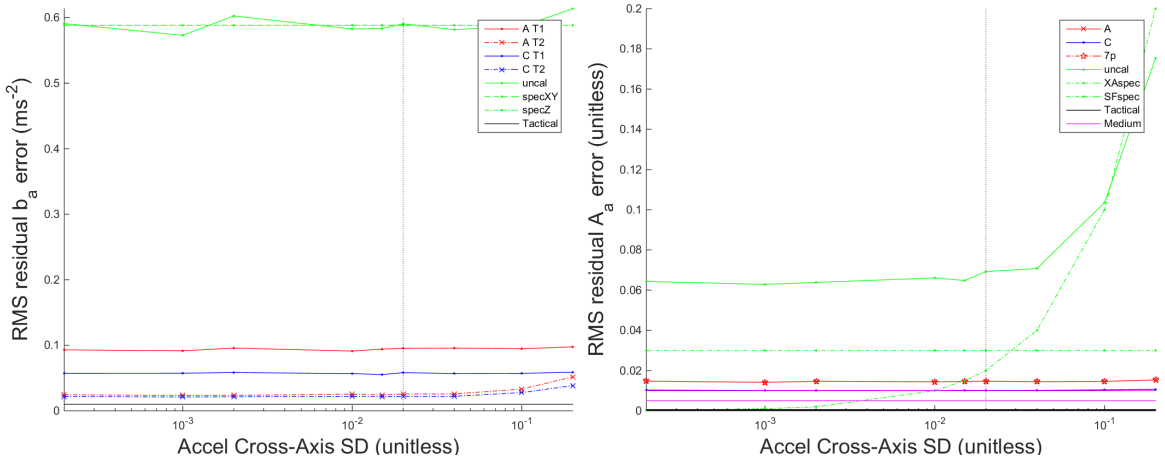
(c) Gyroscope alignment error

(d) Table angle estimate error

Figure 4.23: The effect of accelerometer scale factor error on the residual errors post-calibration.

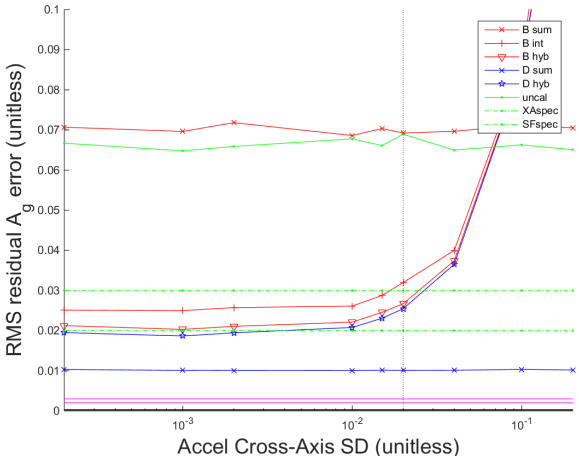
The sensor alignment matrix estimates ($\widehat{\mathbf{A}^{\mathbf{b}}_{\mathbf{a}}}$ and $\widehat{\mathbf{A}^{\mathbf{b}}_{\mathbf{g}}}$) are unusual compared with the other estimates made during this calibration procedure in that their target values are each determined by several separate input parameters. Both $\mathbf{A}^{\mathbf{s}}_{\mathbf{a}}$ and $\mathbf{A}^{\mathbf{s}}_{\mathbf{g}}$ are determined by a 'scale factor error SD' parameter for their diagonal entries, and a 'cross-coupling error SD' parameter for their off-diagonal entries. Additionally the target values for their estimates

[9] $\sin(1°) = 0.0175$

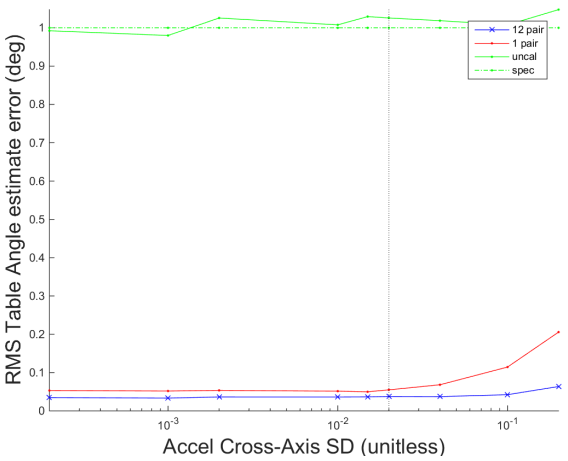


(a) Accelerometer bias

(b) Accelerometer alignment error



(c) Gyroscope alignment error



(d) Table angle estimate error

Figure 4.24: The effect of accelerometer cross coupling error on the residual errors post-calibration.

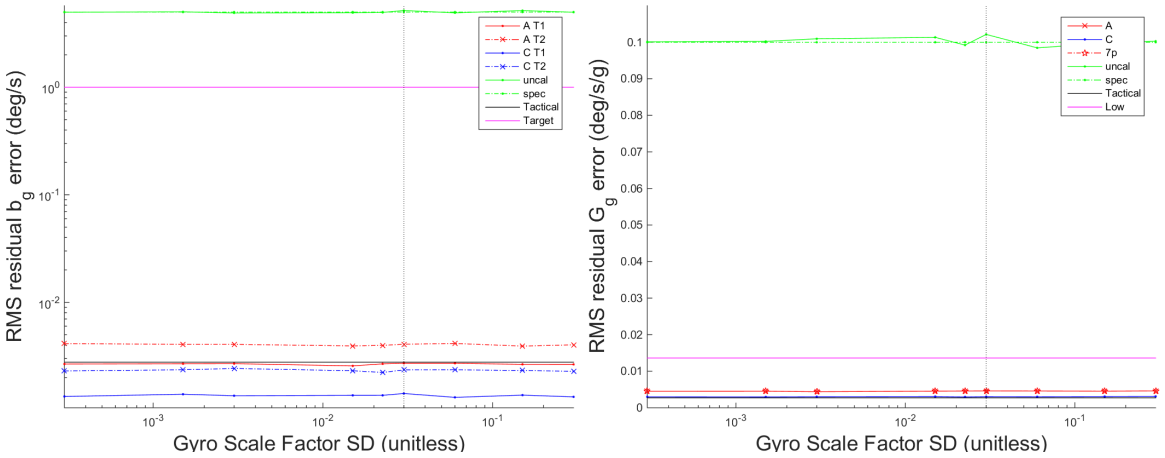
include the attitude of the IMU within the box, $\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}$, see Equations 4.20 and 4.25. Thus the mounting error parameters that determine $\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}$ also affect the uncalibrated values of $\widehat{\mathbf{A}_{\mathbf{a}}^{\mathbf{b}}}$ and $\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}$ simultaneously. As such when considering the effect of varying each of these parameters individually it is worth paying attention to the overall effect of the three, depicted by the solid green line in the $\widehat{\mathbf{A}_{\mathbf{a}}^{\mathbf{b}}}$ or $\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}$ graphs.

As these error sources have certain similarities in their effects, their effect on the statically determined errors will be discussed first and their effect on the dynamically determined $\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}$ afterwards.

Given that the calculation methods for $\widehat{\mathbf{A}_{\mathbf{a}}^{\mathbf{b}}}$ and $\widehat{\boldsymbol{b}_{\boldsymbol{a}}}$ do not involve the gyroscope readings and similarly $\widehat{\mathbf{G}_{\mathbf{g}}^{\mathbf{b}}}$ and $\widehat{\boldsymbol{b}_{\boldsymbol{g}}^{\boldsymbol{b}}}$ do not involve the accelerometer readings [10], these estimates would not be expected to be affected by those inputs. They are not, and so the graphs are not presented for reasons of space.

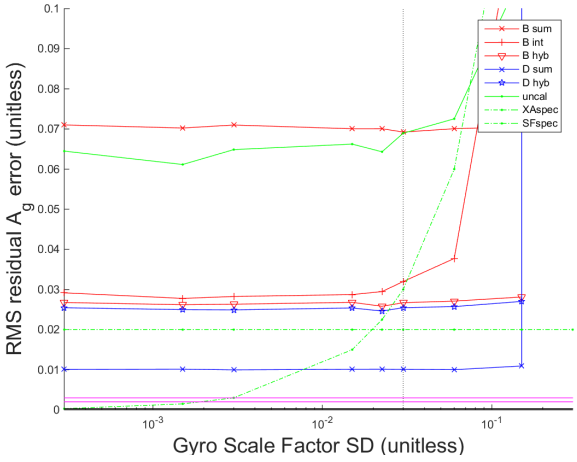
Figure 4.23 and 4.24 show the effect of the accelerometer scale factor and cross coupling errors, respectively, on the residual errors in $\widehat{\boldsymbol{b}_{\boldsymbol{a}}}$, $\widehat{\mathbf{A}_{\mathbf{a}}^{\mathbf{b}}}$, $\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}$ and the table angle estimates.

[10] Except for the estimation of the table angles α and β, which was shown in Section 4.8.1.2 not to be a significant source of error.



(a) Gyroscope bias

(b) Gyroscope g-dependent error



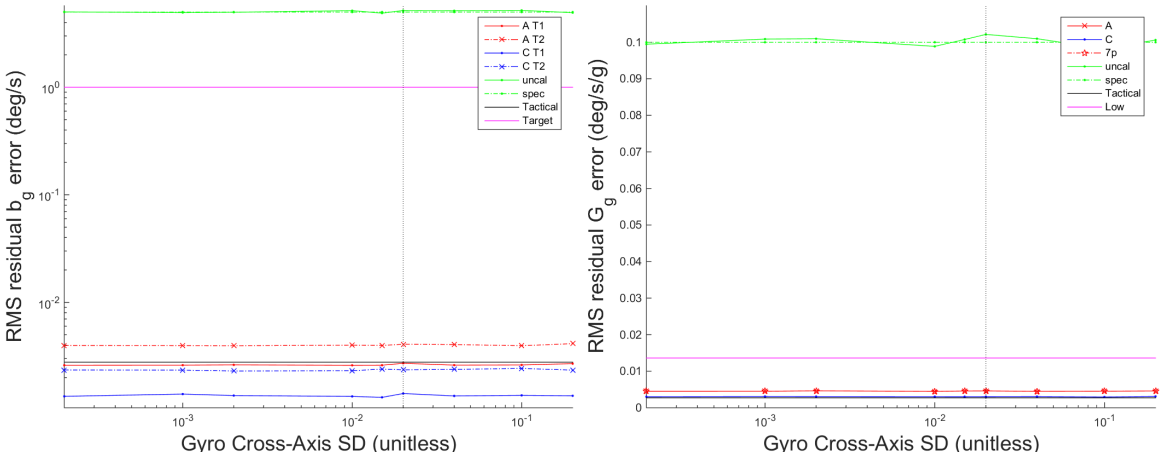
(c) Gyroscope alignment error

Figure 4.25: The effect of gyroscope scale factor error on the residual errors post-calibration.

Figure 4.25 and 4.26 show the effect of the gyroscope scale factor and cross coupling errors, respectively, on the residual errors in $\widehat{\boldsymbol{b_g}}$, $\widehat{\mathbf{G}^{\mathbf{b}}_{\mathbf{g}}}$ and $\widehat{\mathbf{A}^{\mathbf{b}}_{\mathbf{g}}}$. Figure 4.27 shows the effect of the mounting error parameters (which determine $\mathbf{C}^{\mathbf{s}}_{\mathbf{b}}$) on all the estimated quantities.

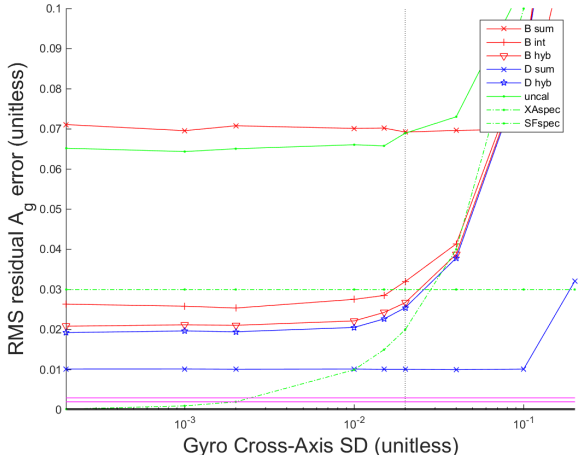
The graphs presented show that the static-procedure error estimates ($\widehat{\boldsymbol{b_a}}$, $\widehat{\boldsymbol{b_g}}$, $\widehat{\mathbf{A}^{\mathbf{b}}_{\mathbf{a}}}$ and $\widehat{\mathbf{G}^{\mathbf{b}}_{\mathbf{g}}}$) are essentially unaffected by these errors, despite the change in their uncalibrated level, i.e. how bad these errors are before calibration. Figures 4.23a, 4.23b, 4.24a and 4.24b show that the starting level of accelerometer scale factor or cross coupling error has virtually no effect on the accuracy of the accelerometer bias or alignment matrix estimates. Similarly, Figures 4.25a, 4.25b, 4.26a and 4.26b show that the statically determined g-dependent error matrix and gyroscope bias are also unaffected by the starting level of error in $\mathbf{A}^{\mathbf{s}}_{\mathbf{g}}$. As the *uncalibrated* accelerometer readings are used to estimate the table angles these are slightly affected, see Figures 4.23d and 4.24d. However, even with 10% starting errors in $\mathbf{A}^{\mathbf{s}}_{\mathbf{a}}$, the residual table angle error is still 10× smaller than the starting value, so this error is not particularly significant.

Varying the mounting angle parameters, i.e. $\mathbf{C}^{\mathbf{s}}_{\mathbf{b}}$, simultaneously changes the starting accuracy of both $\widehat{\mathbf{A}^{\mathbf{b}}_{\mathbf{a}}}$ and $\widehat{\mathbf{A}^{\mathbf{b}}_{\mathbf{g}}}$. The most notable effect that can be seen in the statically determined parts of Figure 4.27 are that the Technique 2 method for calculating $\widehat{\boldsymbol{b_a}}$, pre-



(a) Gyroscope bias

(b) Gyroscope g-dependent error



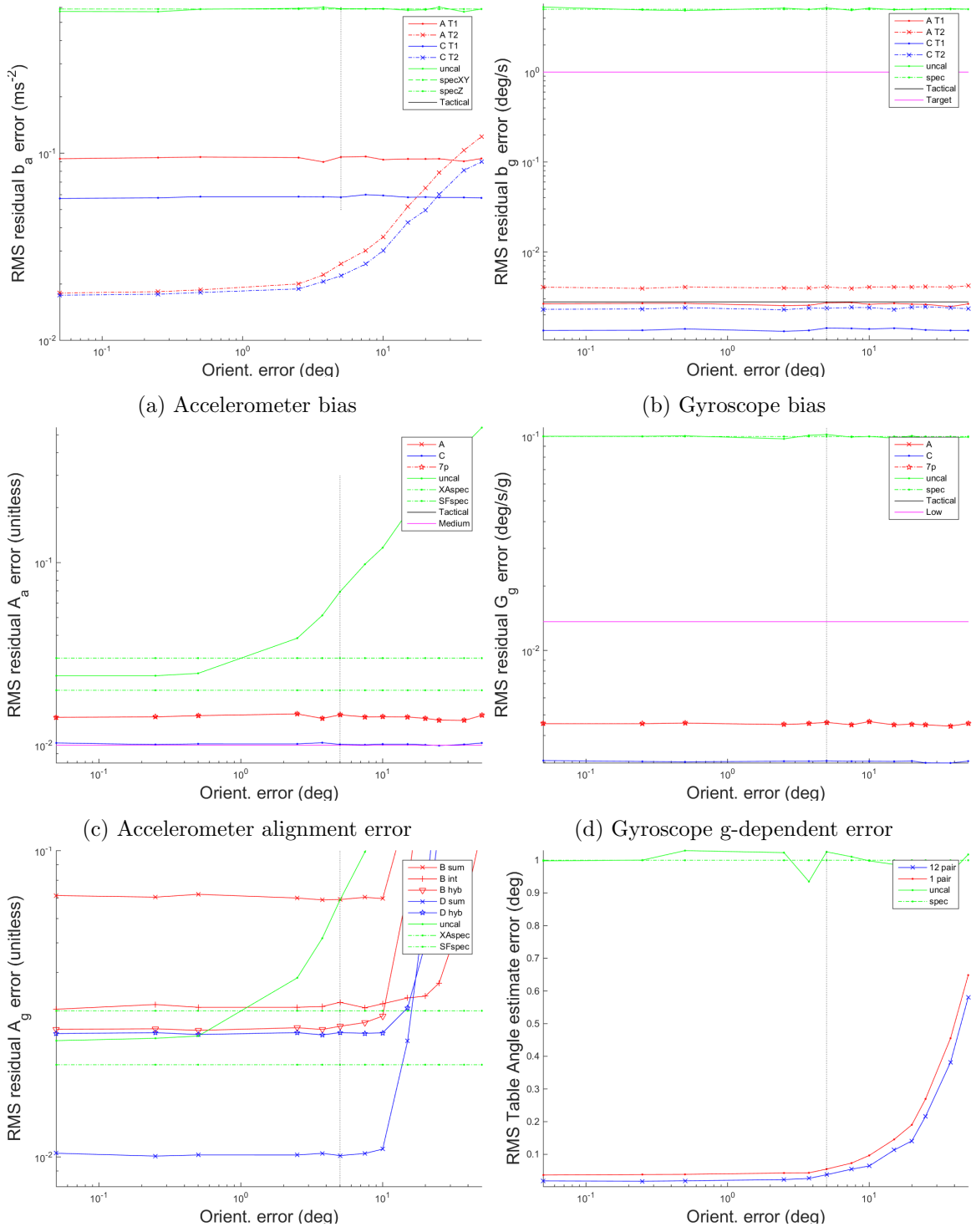
(c) Gyroscope alignment error

Figure 4.26: The effect of gyroscope cross coupling error on the residual errors post-calibration.

sented in Equation 4.67 and discussed in Section 4.8.1.1 becomes less effective, where the Technique 1 method of Equation 4.63 is unaffected. This is due to the vertical reading being less co-incident with the gravity vector with greater values of $\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}$, and thus being less effective at reducing the cube-face errors. Also, as it relies on the uncalibrated readings, the estimate of the table angles gets worse as the mounting angle error increases, but this does not translate into a significant increase in the IMU errors that are estimated using it ($\widehat{\boldsymbol{b_a}}$, $\widehat{\boldsymbol{b_g}}$, $\widehat{\mathbf{A}_{\mathbf{a}}^{\mathbf{b}}}$ and $\widehat{\mathbf{G}_{\mathbf{g}}^{\mathbf{b}}}$). The estimation errors for two statically-determined gyroscope parameters, $\widehat{\boldsymbol{b_g}}$ and $\widehat{\mathbf{G}_{\mathbf{g}}^{\mathbf{b}}}$, remain unaffected by the mounting angle.

The effect of the scale-factor, cross-coupling and mounting errors on the dynamic part of the calculation is more complex. For instance, as all of the estimates are made from iterative processes, if the starting errors in the $\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}$ matrix are large then the entire process can become unstable.

The hybrid method uses the assumption that $\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}} = \widehat{\mathbf{A}_{\mathbf{a}}^{\mathbf{b}}}$ for all the off-diagonal estimates with only the diagonal elements of $\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}$ being determined dynamically. Equation 4.75 shows the assumption that allows the hybrid method to work, simply put the mounting error must be significantly bigger than either the accelerometer or gyroscope cross-coupling errors. The



(a) Accelerometer bias

(b) Gyroscope bias

(c) Accelerometer alignment error

(d) Gyroscope g-dependent error

(e) Gyroscope alignment error

(f) Table angle estimate error

Figure 4.27: The effect of IMU mounting error on the residual errors post-calibration.

accelerometer's cross coupling error, Figure 4.24c, very strongly affects the hybrid methods. Whereas, the accelerometer scale factor, Figure 4.23c, has only a very slight effect on the two hybrid methods; this supports the assumption. The other $\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}$ estimation methods are unaffected by these two accelerometer inputs, except for the integrated method which appears to become unstable due to the poor initialisation of $\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}$, when the starting level of $\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}$ is more than 5× larger than the default.

In a similar manner, when the starting gyroscope scale factor (Figure 4.25c) and cross-coupling (Figure 4.26c) errors become larger the hybrid method becomes less effective. However, the other methods excepting the integrated method are unaffected scale factor or cross coupling SDs of less than 10%. At this point, the errors is so large that the correction method becomes unstable, and all the methods produce very poor estimates, due to feedback in their iterative processes. The hybrid method is affected by starting gyroscope $\mathbf{A}_{\mathbf{g}}^{\mathbf{s}}$ levels similarly to the way it was for the starting accelerometer $\mathbf{A}_{\mathbf{a}}^{\mathbf{s}}$ levels.

Figure 4.27e shows the effect of the mounting angle errors on the $\widehat{\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}}$. The mounting angle makes virtually no difference for techniques other than the hybrid method until the angle has moved to more than double the default. When the mounting error becomes greater than this the estimation system quickly becomes ineffective, leading to very inaccurate estimates. This leads one to conclude that the mounting error must be kept below 10° for the estimation techniques to work.

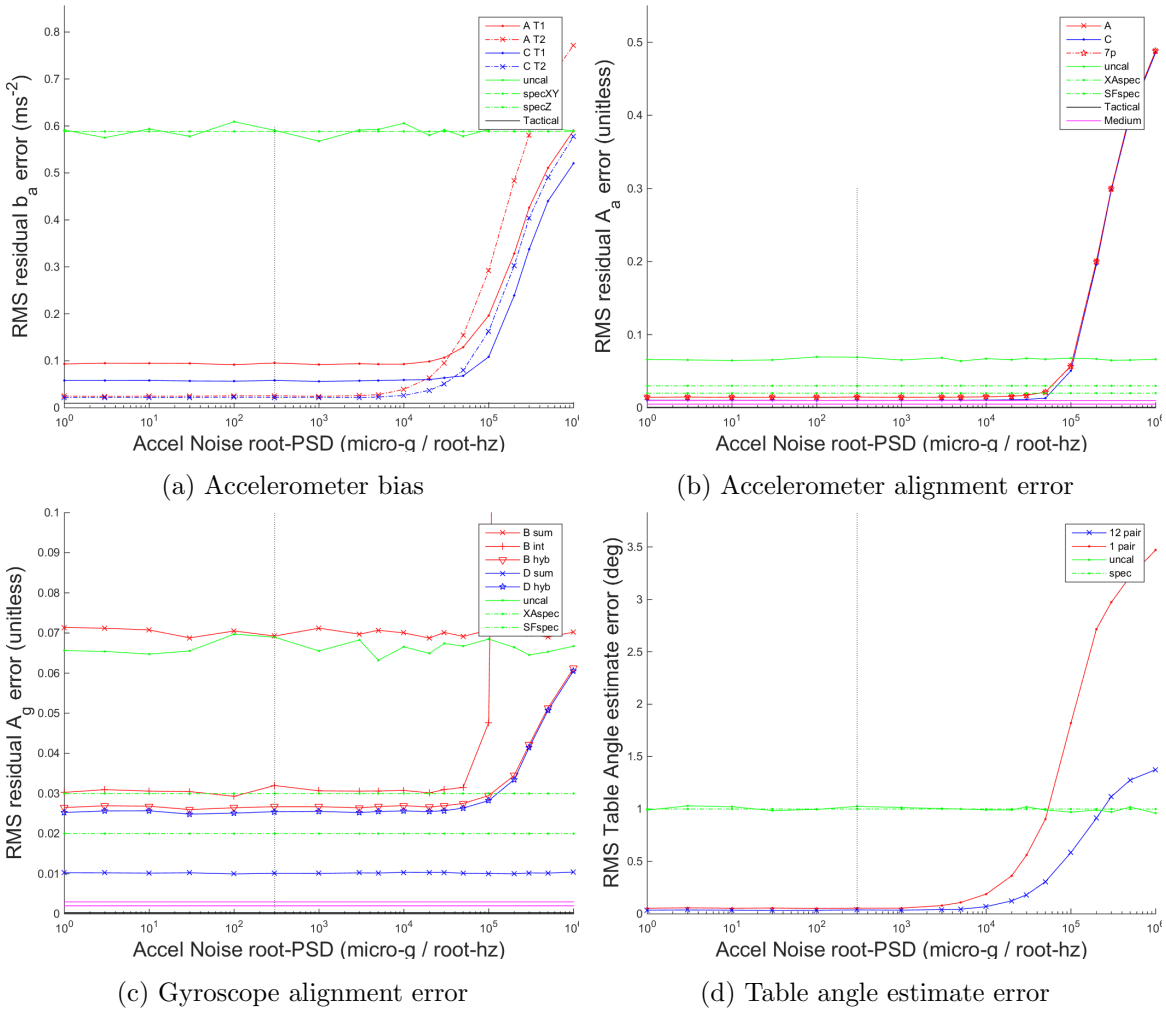
The hybrid technique's assumption (Equation 4.75), that the diagonal elements of $\mathbf{A}_{\mathbf{g}}^{\mathbf{b}}$ and $\mathbf{A}_{\mathbf{a}}^{\mathbf{b}}$ are highly correlated (in the *box* frame), is not applicable when the mounting angle error is small. So when the mounting error is under 10× the default settings, simply using $\mathbf{I_3}$, is a better estimate than the hybrid method (the solid green no-calibration line is below the hybrid method lines in Figure 4.27e)

4.8.2.4 Accelerometer and gyroscope noise

In order to examine the effect of sensor noise on the calibration results a very wide range of possible inputs were tested, ranging several orders of magnitude better and worse than real sensors, so the results must be considered in the context of the default values which represent a typical consumer-grade MEMS IMU and noise levels more than 1.5 to 2 × worse than that are unrealistically high.

The sensor noise will have a direct effect on both the dynamic and static calibration. We must consider particularly for the static calibration that its effect is fairly low partly because of the long averaging times (5 sec for set A, effectively 20 sec for set C).

The effect on the accelerometer's calibration is clear. Decreasing the noise root-PSD, even by several orders of magnitude from the default 300 micro-g/$\sqrt{\text{Hz}}$ makes no appreciable difference to the results, and in order for there to be a noticeable worsening of the results the accelerometer's noise parameter must be increased by more than an order of magnitude (5000 micro-g/$\sqrt{\text{Hz}}$). Note that the Technique 1 method is considerably less sensitive to noise than the Technique 2 method. This small (relative to the residual error) numerical value calculated in Equation 4.99 explains this.



(a) Accelerometer bias

(b) Accelerometer alignment error

(c) Gyroscope alignment error

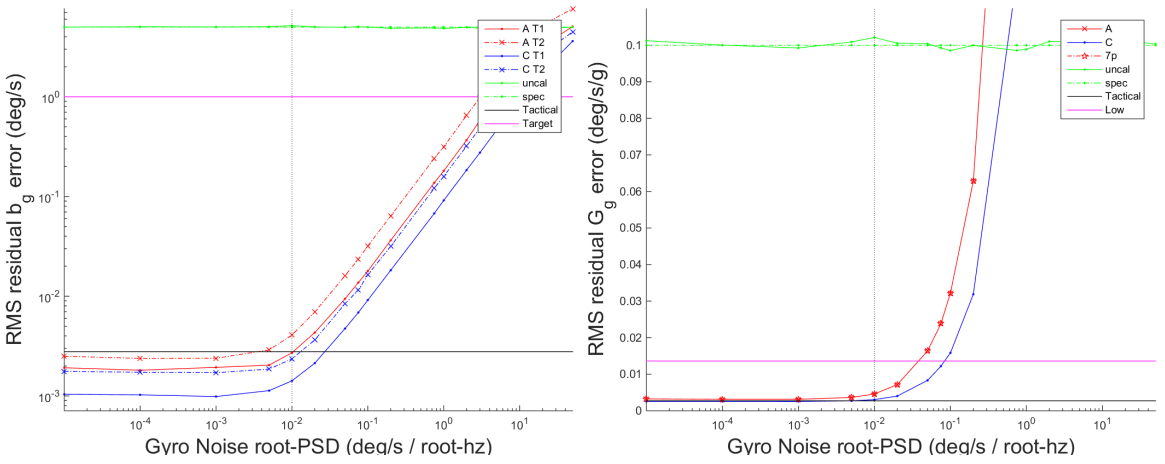
(d) Table angle estimate error

Figure 4.28: The effect of accelerometer noise on the residual errors post-calibration.

By contrast, the gyroscope's statically determined errors ($\widehat{\boldsymbol{b_g}}$ and $\widehat{\mathbf{G_g^b}}$) are improved by reducing its noise root-PSD below the default $0.01\,°/s/\sqrt{Hz}$. Although no further improvement is seen below $0.001\,°/s/\sqrt{Hz}$, note that this level is still above that of a tactical-grade IMU. This shows that the gyroscope noise is still having a significant effect even with the 5 second averaging times. At the default level, the gyroscope noise is not the completely dominant error source, it shares that with the cube error. So the line on Figure 4.29a is not straight (log-log scale) until well above (10×) the default level. Again this can be explained by considering that the numerical value for the default settings calculated in Equation 4.100 is significant in comparison to but still smaller than the residual errors.

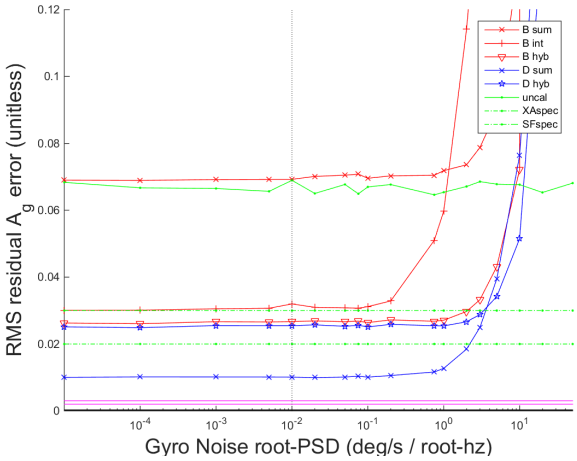
When we examine the accelerometer noise's effect on the dynamic calibration, aside from the hybrid method unsurprisingly becoming less accurate as the $\widehat{\mathbf{A_a}}$ calibration worsens. There is little effect until the noise is above 50,000 micro-g/$\sqrt{Hz}$, which is extremely high.

The effect of the gyroscope noise on the dynamic calibration is more complex. Between the default level and about $1\,°/s/\sqrt{Hz}$, there is a slow worsening of the calibration accuracy, likely the indirect effect of less well compensated bias. Above that level, the solution rapidly decreases in quality, as the bias and integrated noise become more significant than the other errors affecting $\widehat{\mathbf{A_g^b}}$. Notably, the integrated method fails rapidly at about an order of



(a) Gyroscope bias

(b) Gyroscope g-dependent error



(c) Gyroscope alignment error

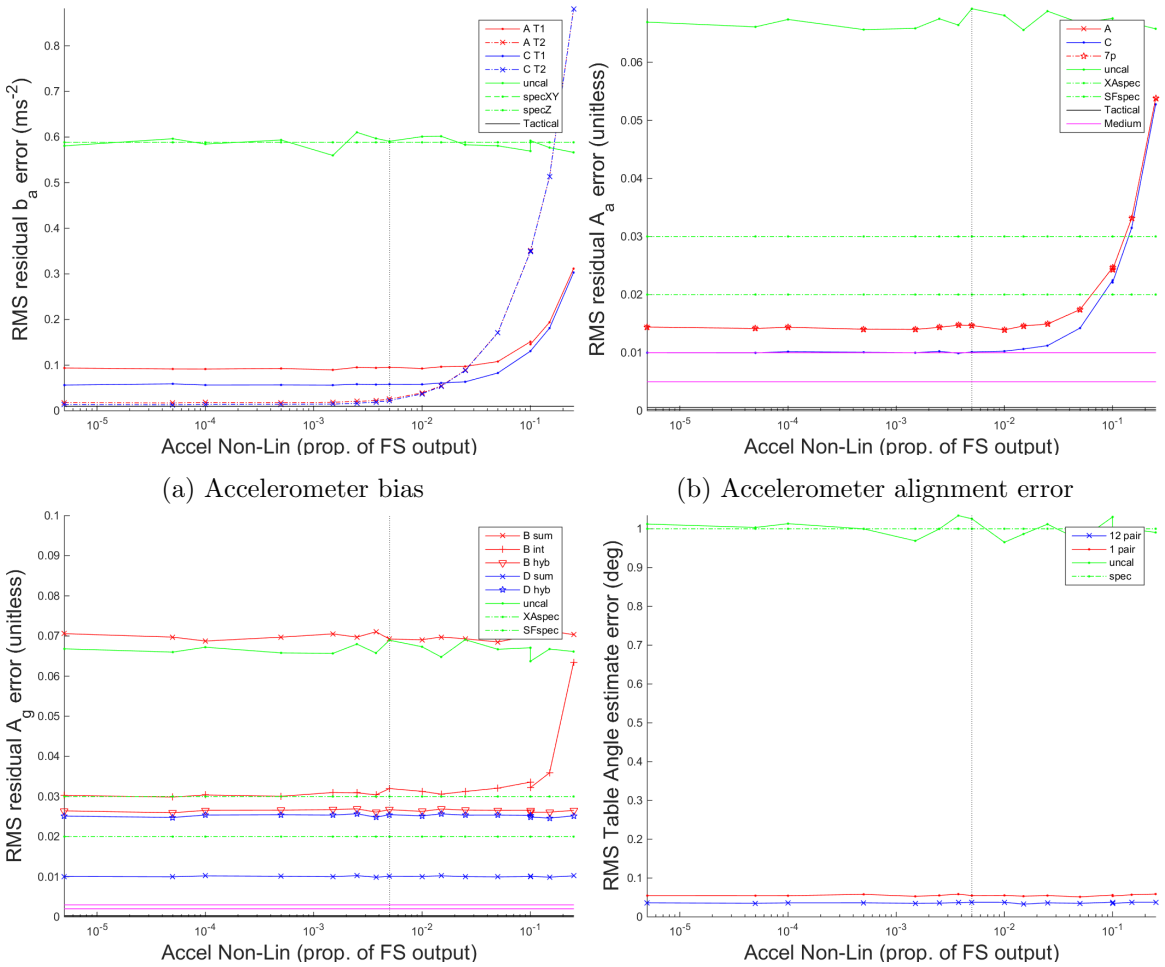
Figure 4.29: The effect of gyroscope noise on the residual errors post-calibration.

magnitude lower noise than the other methods.

In addition to the other sensor specifications tested different accelerometer and gyroscope quantisation levels were tested. The level spanned the range of 8- to 24-bit ADC-words, compared to the default level of 16-bit. However, no variation in the accuracy of any of the output calibrations was found. The IMU model used for this simulation keeps track of quantisation residuals, adding them to the next epoch's reading, and so the maximum effect on the mean value of the static readings is the quantisation level divided by the number of epochs (e.g. 5 sec. $\times$ 100 Hz $=$ 5000 epochs), making this an insignificant error source (e.g. for the gyroscope at default level $\frac{1.3\times10^{-4}}{5000} = 2.6 \times 10^{-8}$ rad/sec). The graphs are not shown to conserve space. It has been observed (e.g. in [8]) that many low-cost sensors are simply discarding quantisation residuals, in which case quantisation errors would generally behave in a similar manner to the sensor noise, investigated in this section.

4.8.2.5 Sensor non-linearity

The model used to simulate sensor non-linearity is quadratic, but with a *separate coefficient* for positive and negative true values (Equation 4.9). This is expressed as part of Equations 4.7 (accelerometer) and 4.8 (gyroscope). Non-linearity is typically expressed on datasheets as



(a) Accelerometer bias

(b) Accelerometer alignment error

(c) Gyroscope alignment error

(d) Table angle estimate error

Figure 4.30: The effect of accelerometer non-linearity on the residual errors post-calibration.

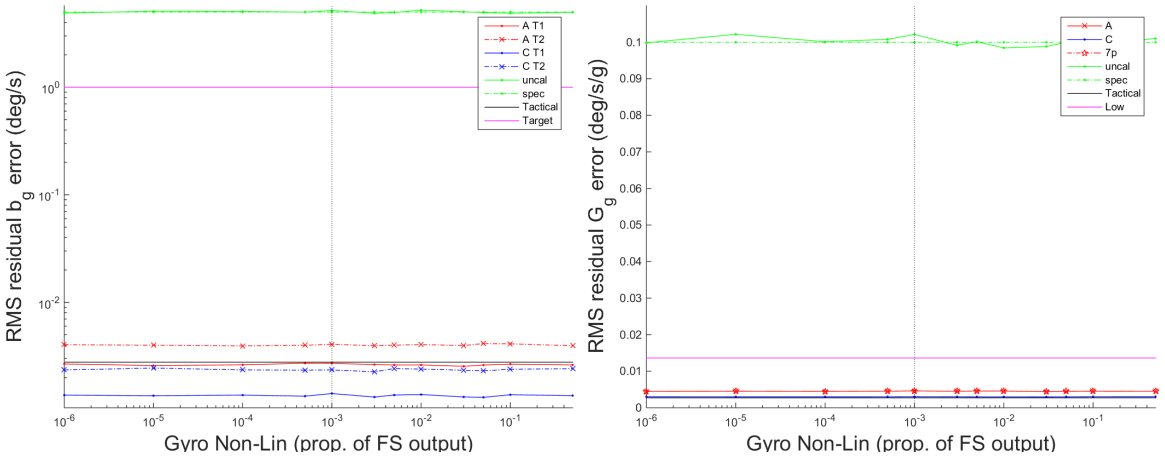
'% of full scale output' (% FS) thus the default settings, which are meant to represent [23], are 0.1% FS for the gyroscope and 0.5% FS for the accelerometer. If we ignore sensor mounting error (so using Equations 4.5 and 4.6), a single output (here the 'x' output) of the accelerometer becomes

$$\widetilde{f^s_{isx}} = b_{ax} + (\mathbf{A^s_a} \boldsymbol{f^s_{is}})_x + w_a + n(\boldsymbol{l_a}, \boldsymbol{f^s_{is}}) \tag{4.101}$$

$$\widetilde{f^s_{isx}} = b_{ax} + A^s_{a11} f^s_{isx} + A^s_{a12} f^s_{isy} + A^s_{a13} f^s_{isz} + w_a + \begin{cases} l_a(1,1) \times {f^s_{isx}}^2, & \text{when } f^s_{isx} \geq 0 \\ l_a(1,2) \times {f^s_{isx}}^2, & \text{otherwise} \end{cases} \tag{4.102}$$

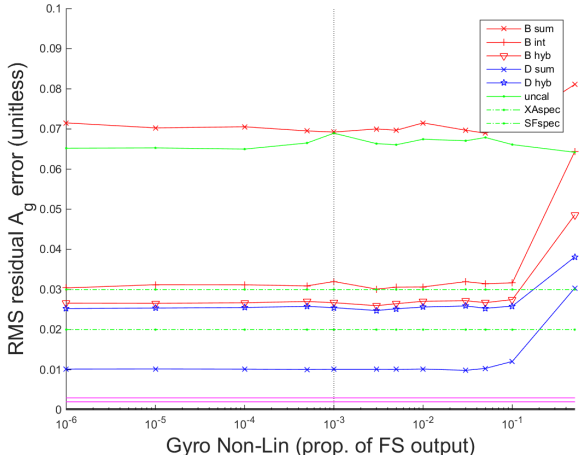
If $X \sim \mathbf{N}(a, b^2)$ is (temporarily) abbreviated to $X = a \pm b$. Then, noting that '2g' is the full-scale measurement range, with default settings this becomes

$$\widetilde{f^s_{isx}} = \pm 60 \text{ milli-g} + (1 \pm 0.03) f^s_{isx} \pm 0.02 f^s_{isy} \pm 0.02 f^s_{isz} + w_a + \begin{cases} \frac{\pm 0.005}{2g} {f^s_{isx}}^2, & \text{when } f^s_{isx} \geq 0 \\ \frac{\pm 0.005}{2g} {f^s_{isx}}^2, & \text{otherwise.} \end{cases} \tag{4.103}$$



(a) Gyroscope bias

(b) Gyroscope g-dependent error



(c) Gyroscope alignment error

Figure 4.31: The effect of the gyroscope non-linearity on the residual errors post-calibration.

The equivalent for the gyroscope's x-output, noting that '250 °/s' is the full-scale measurement range, is

$$\widetilde{\omega^s_{isx}} = \pm 5\,^\circ/\mathrm{s} + (1 \pm 0.03)\omega^s_{isx} \pm 0.02\omega^s_{isy} \pm 0.02\omega^s_{isz} + w_g \pm 0.01\frac{f^s_{isx}}{g} \pm 0.01\frac{f^s_{isy}}{g} \pm 0.01\frac{f^s_{isz}}{g} + \begin{cases} \frac{\pm 0.001}{250\,^\circ/\mathrm{s}}\omega^{s\,2}_{isx}, & \text{when } \omega^s_{isx} \geq 0 \\ \frac{\pm 0.001}{250\,^\circ/\mathrm{s}}\omega^{s\,2}_{isx}, & \text{otherwise.} \end{cases} \quad (4.104)$$

This section tests an important assumption of all the calibration techniques in this Chapter, namely that both gyroscope and accelerometer non-linearity are small enough to be ignored. Note that, because the model's non-linearity terms depend on the square of *true* input and are related to the sensors' range, during the static procedure, the gyroscope is barely affected by the non-linearity at all, as the true angular rate is nearly zero, but the non-linearity of the accelerometer is considerably excited because the true specific force is approximately g and thus 50% the default measurement range.

Accelerometer non-linearities were simulated from 0.0005% to 25% of FS. Examining the behaviour of the table estimate (Figure 4.30d) one can see no significant effect of the non-linearity. This is because the horizontal readings that this method uses (Equation 4.45) are close to 0 ms^{-2}, and thus not significantly affected by the non-linearity. As the accelerometer

non-linearity does not affect the gyroscope static calibration there is unsurprisingly little effect on the dynamic calibration(Figure 4.30c), except for the hybrid method, which is very slightly affected as $\widehat{\mathbf{A_a}}$ gets worse.

Reducing the level of accelerometer non-linearity does not have any significant effect on the accuracy of the $\widehat{\mathbf{A_a}}$ calibration, indicating that, at default level, it does not have a significant effect on the results. If the sensors' non-linearity is approximately an order of magnitude greater ($\approx 5\%$) then this starts to be more significant. Note that, as the level of non-linearity rises above this then the calibration continues to worsen and the 24-position and 6-position methods tend towards each other. This is because neither of the improvements from set C, namely the insensitivity to table levelling and the increased static averaging time, help to mitigate this source of error. So, as the non-linearity increases, it makes less and less difference whether set A or set C methods are used.

Figure 4.30a shows the behaviour of the accelerometer bias calibration. The convergence of the two methods as mentioned above is also present. The Technique 1 method, is insensitive to non-linearities below about 3% so, at the default level, the non-linearity is making virtually no difference. However, the Technique 2 method is more sensitive to non-linearities. Reducing the non-linearity below the default level slightly improves the accuracy, thus there is a significant effect at default level. Increasing the non-linearity results in the error growing more quickly for the Technique 2 method than the Technique 1 method. Beyond 2% non-linearity, it is better to use Technique 1. This behaviour can be explained as follows. In the Technique 1 method, each result is the average of six readings, four readings with magnitudes near zero, one around '+g' and another around '-g'. The Technique 2 method is just the average of the last two. Thus if the variance of non-linearity errors is, ${\sigma_{NL}}^2$, then the variance in the bias calibration due to non-linearity for the Technique 1 method is

$$\boldsymbol{b_a} - \widehat{\boldsymbol{b_a}} = \frac{1}{6}\left(0+0+0+0+\frac{g^2}{2g}l_1+\frac{(-g)^2}{2g}l_2\right) = \frac{g}{12}\left(l_1+l_2\right)$$

$$\implies \mathbf{Var}(\mathrm{T1}) = \left(\frac{g}{12}\right)^2\left({\sigma_{NL}}^2+{\sigma_{NL}}^2\right) = \left(\frac{g\sqrt{2}}{12}\sigma_{NL}\right)^2 \quad (4.105)$$

Similarly, for the Technique 2 method,

$$\boldsymbol{b_a} - \widehat{\boldsymbol{b_a}} = \frac{1}{2}\left(\frac{g^2}{2g}l_1+\frac{(-g)^2}{2g}l_2\right) = \frac{g}{4}\left(l_1+l_2\right)$$

$$\implies \mathbf{Var}(\mathrm{T2}) = \left(\frac{g}{4}\right)^2\left({\sigma_{NL}}^2+{\sigma_{NL}}^2\right) = \left(\frac{g\sqrt{2}}{4}\sigma_{NL}\right)^2 \quad (4.106)$$

These equations show why the Technique 2 method is more susceptible.

Simulations were carried out with gyroscope non-linearity from 0.0001% to 50% of FS (Figure 4.31). These values are greatly in excess of what might be considered reasonable values (1% of FS is poor linearity even for a consumer grade sensor). However, as pointed out above, the static calibration is insensitive to gyroscope non-linearity, so unsurprisingly there is no visible effect on $\widehat{\boldsymbol{b_g}}$ or $\widehat{\mathbf{G_g}}$ (Figures 4.31a and 4.31b). Perhaps more surprising is that there is also no visible effect on $\widehat{\mathbf{A_g}}$ (Figure 4.31c), although this may be due to the fact that the default rotation time is a relatively slow 10 seconds, so $\approx 36\,^\circ/\mathrm{s}$ which is a fairly

small faction of the measurement range of 250 °/s. So even the very large non-linearity is not significantly excited until the non-linearity is above 10% of full-scale. Although, if the rotation speed were greater, there would be more of an effect.

4.9 Summary

This chapter presents a method for user-conducted calibration procedures. The motivation for and theory behind the type of calibration procedure used are presented in Sections 4.1 and 4.2. Reference frames needed to describe and simulate the calibrations are in Section 4.3, and the model of IMU's errors that is assumed is shown in Section 4.4.1.

A few alternative methods for calculating estimates of the IMU's calibration parameters from the data recorded during are presented in Section 4.5. The accelerometer errors and some of the gyroscope errors can be estimated from static measurements ($\boldsymbol{b_a}$, $\boldsymbol{b_g}$, $\mathbf{A_a}$ and $\mathbf{G_g}$). However, the gyroscope scale factor and alignment matrix, $\mathbf{A_g}$, requires rotation for it to be observable. To this end there are both static and dynamic parts of the calibration procedure.

For the static part of the calibration procedure, measurements are collected on all six sides of the cube, either one per face (set A), the average of four per face at 90° heading rotation (set C), or one per face and one additional measurement at 180° heading rotation for one of the faces (7 position). In the six-position version the table is assumed to be flat, for the other two an estimate of the table's deviation from level is made using the 1 pair or 12 pairs of measurements at 180° heading rotation for each face. Then the sum of the measurements opposite faces' measurements is used to eliminate components depending on specific force and thus estimate the bias for both the gyroscope and the accelerometer, using either of two possible techniques. The difference between opposite faces' measurements can be used to estimate one column of the $\mathbf{A_a}$ or $\mathbf{G_g}$ for accelerometer and gyroscope measurements, respectively. This estimate requires division by double the (vertical) specific force due to the reaction to gravity, which can be corrected for the table's deviation from level, if known. The 6-position method has been presented before (e.g. [158]), but the 24-position method tested here does not appear in the existing literature.

The dynamic part of the calibration procedure is based on performing three complete 360° rotations of the IMU returning it to the same attitude (and position) between each one. These are known as a zero-attitude-updates. In order for there to be a simple analytical solution the algorithm the rotation each rotation must be about one of the (box-frame) cardinal axes. Under this condition each of the rotations provides one column of the $\widehat{\mathbf{A_g}}$-matrix. The simulations test two versions of this, one where the required rotations are carried out in mid-air, and another where six-rotations are carried out, one with each face in contact with the table, and then there are two rotations to average between for each column of the matrix. Also tested are several alternative methods to calculate $\widehat{\mathbf{A_g}}$ from the measurements. This approach to calculating $\widehat{\mathbf{A_g}}$ has not previously been examined. Compared to existing techniques (see Section 2.5.1.3), it is simple to carry out the necessary manoeuvres, computationally fast and not dependent on the signals of an external sensor.

Monte Carlo simulations of this calibration procedure were carried out in order to determine how accurate these methods are, and which error sources are the most important for this accuracy. The method for this is presented in Section 4.6. However, in short, randomly generated error values, based on input values for these errors' SDs, for each of 500 runs were generated. These error values were used to generate imperfections in the way the procedure was carried out and using these imperfections a true trajectory is calculated. From this trajectory the output values that a perfect IMU would give are calculated. More of the randomly generated error values generate the IMU's errors. These errors are used, with the perfect IMU output to generate an as-measured IMU output. This output is given to the calibration algorithms and used to estimate the IMU errors. These estimates are compared to the real values to determine the accuracy of the calibration, which is the output of the simulation (see Figure 4.12). User calibration techniques presented in the existing literature typically only demonstrate that repeatable results are produced (see Section 2.5.1.3), which is a necessary but not sufficient condition. This chapter presents a detailed study of the calibration accuracy and how the accuracy of each estimated IMU error is affected by factors ranging from the specifications of the IMU to the care with which the user conducts the calibration. This allows different calibration algorithms and manoeuvres to be compared in terms of their accuracy.

The results of these simulations are presented in Section 4.8. These results show that the most important error for the accuracy of the calibration is the angular accuracy of the calibration box. This error is so important that techniques which reduce the effect of this error on the calibration's accuracy, even at the cost of increasing the effect of the other errors, are viable. An example of this is the Technique 2 accelerometer bias calibration method (Equation 4.67). After this, the most significant source of error for the four statically-determined error parameters is the gyroscope noise, or in the case of the six-position (set A) calibration the table's error-from-level. These are followed by the accelerometer non-linearity.

The number of seconds of measurements recorded at each static position has a surprisingly small effect in relation to the other error sources (when they are at default level), such that for realistic amounts of time ($\geq$ 1 second) it makes little difference to the accuracy of the calibration.

In terms of the different manoeuvre sets and calibration algorithms tested the following recommendations are made.

The set C (24-position) manoeuvres always produce better results than the set A (6-position). There are two reasons for this. First, they virtually eliminate inaccuracy due to the table not being level, second the total sample period for each face is effectively quadrupled which approximately halves inaccuracies due to sensor noise. This manoeuvre set is recommended in every case where the table's deviation from level is of a similar or greater magnitude as the box's deviation from being orthogonal sided. If the table is *very close* to level the set C manoeuvres will perform similarly to set A with a longer static recording on each face.

For calculating $\widehat{\boldsymbol{b_a}}$ the Technique 2 algorithm (Equation 4.67) is recommended for box-deviation-from-orthogonal SDs in excess of 0.25°, due to its quadratic ($\mathcal{O}(\theta^2)$) rather than linear ($\mathcal{O}(\theta)$) sensitivity to these (and table levelling) errors (θ radians), unless the accelerometer noise or non-linearity specification are significantly worse. For highly accurate cubes of 0.25° SD and lower, the Technique 1 method is recommended instead due to its reduced sensitivity to accelerometer noise and other errors.

Technique 1 is always recommended for $\widehat{\boldsymbol{b_g}}$ as sensor noise is a more significant error source the cube- or table-angular errors. However, all the techniques tested can easily bring the $\widehat{\boldsymbol{b_g}}$ calibration within the level required for stability determined in Chapter 3.

For $\widehat{\mathbf{A_a}}$ and $\widehat{\mathbf{G_g}}$ the 6- and 7-position techniques perform almost identically, and both perform worse than the 24-position method, which is recommended. Even with the relatively conservative 1° box-deviation-from-orthogonal SD the required medium quality of $\widehat{\mathbf{A_a}}$ calibration required for KF stability can be achieved. Similarly the low level of residual $\widehat{\mathbf{G_g}}$ error required by Chapter 3 is easily exceeded at default levels, where the calibration is nearly at tactical level. The definition of these levels are in Table 3.3.

For the dynamic part of the calibration procedure, the most significant is the rotation-axis error, which is a separate input for set B and for set D is included in the cube error, this error causes corrections for the columns of $\widehat{\mathbf{A_g}}$ to be mixed up. The next most important errors is the replacement heading error, i.e. the accuracy with which the zero-attitude update is carried out.

This strong dependence on the rotation axis being correct, combined with the challenging low target residual error (see Table 3.3), means that hand-rotated set B manoeuvres will not reach a sufficient level of accuracy for stable KF integration. The set D manoeuvres do not need to reach such a good level of axis accuracy This is due to the repetition (i.e. 6 rotations rather than the minimum 3) and is achievable as the rotation axis is more accurate due to the use of the face as a guide.

In Chapter 3 it was shown that for low levels of gyroscope higher-order errors ($\mathbf{G_g}$ and $\mathbf{A_g}$) the KF integration was stable for all or nearly all levels of bias tested, and for medium levels it was only stable in some circumstances (see Figure 3.12). The results of the $\widehat{\mathbf{A}_\mathbf{g}^\mathbf{b}}$ calibration presented in this chapter show that the calibration accuracy mostly lies between these two levels. For example, for the $\widehat{\mathbf{A}_\mathbf{g}^\mathbf{b}}$ residual error for the set D manoeuvres to be below 1% (medium level) requires a box-deviation-from-orthogonal SD of <1.25°, which is relatively achievable, whereas for it to be below 0.3% (low level) requires it to be $\leq 0.25°$. Further tests would be needed to determine the target level more precisely.

Chapter 5

User Calibration Experiments

This chapter presents some experiments conducted with the array of MPU-9250 MEMS IMU sensors described in Section A.2. The main purpose these experiments serve is twofold. First the calibration procedures proposed in Section 4.5 are tested on the data recorded during a manually conducted calibration procedure with real sensors. Second, to validate the simulations presented in Chapter 4 by comparison of their estimated accuracy with that calibration derived from real sensors. A secondary aim of this chapter are to characterise the performance of the MPU9250 sensors in ways that can be used for the sensor array techniques which will be presented in Chapter 6. The experiments presented here use data recorded on an optical table using the hardware described in Appendix A, an illustrative video still is shown in Figure 5.1.

In order to conduct the calibration effectively this chapter begins by determining the optimal noise settings (Section 5.1) for the MPU9250 IMUs by testing at a variety of bandwidth settings, output rates and measurement ranges. In Section 5.2 the repeatability of the IMU calibration parameter estimates is examined by recording 12 complete calibration procedures with the array and comparing the estimates. In Section 5.3, in order to test the calibration simulations, a repeatability simulation is conducted with inputs to match the experiments precision and IMU specifications, this is compared with the repeatability in the previous section. Next, the effect of changing the IMUs measurement range setting on the IMU systematic error parameters is examined (Section 5.4) and also how much these parameters vary over the course of 3 calibrations several weeks apart (Section 5.5).

Experiments are run where the IMUs measurements, both with and without calibration, are integrated with the inertial navigation equations to give a position velocity and attitude (PVA) solution. These experiments included both static, and pedestrian motion tests and is presented in Section 5.6. This PVA solution can provide an indirect measurement of the calibration accuracy by comparison between sensors and/or a known truth. The chapters findings are briefly summarised in Section 5.7.

5.1 IMU noise characterisation

Before running more sophisticated experiments some simple noise characterisation experiments were carried out to determine the actual noise performance of the sensors on the IMU array described in Section A.2, when their built in digital low pass filter (DLPF), is

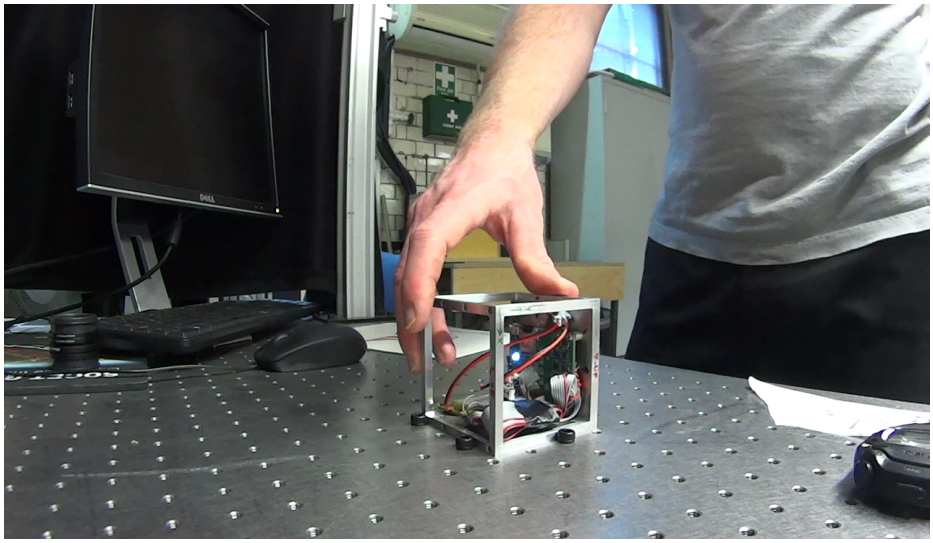


Figure 5.1: Still from a video of the calibration procedure.

Error	Specified Value	SI equivalent
Gyro. Total RMS Noise with DLPFCFG=2 (92 Hz)	0.1 °/s (Typical)	0.0017 rad/s
Gyro. Rate Noise Spectral Density	0.01 °/s/$\sqrt{\text{Hz}}$ (Typical)	0.00017 rad/s /$\sqrt{\text{Hz}}$
Accel. Total RMS Noise with DLPFCFG=2 (94Hz) [sic]	8 milli-g (Maximum)	0.079 ms^{-2}
Accel. Noise Power Spectral Density (Low noise mode)	300μg/$\sqrt{\text{Hz}}$ (Typical)	0.0029 ms^{-2}/$\sqrt{\text{Hz}}$

Table 5.1: The noise performance specifications of an Invensense MPU-9250 IMU, from [23]. Note that the DLPF config setting (DLPFCFG) of 2 defines a filter bandwidth of 92Hz for both accelerometers and gyroscopes according to the register map [182]

used with different bandwidth settings and different output rates, and thus determine the optimal values for these settings in the context required in this thesis.

The specified noise performance of the Invensense MPU9250 IMU according to the datasheet [23] is given in Table 5.1. Note that the total RMS noise is only defined for one of the DLPF settings, and the root-PSD specification does not state at what filter settings this is defined.

Several experiments were run to verify that the MPU9250 sensors behaved as would be expected from reading the datasheet [23] and register map [182]. In these experiments samples of static data were collected from the sensor array for at least 10 minutes and then analysing the final 100 seconds. This allows the sensors to warm up to operating temperature, which is important as several errors, e.g. the biases, vary with temperature. The data is analysed by calculating the SD of the recorded data, from which the implied or apparent root-PSD can be calculated.

$$\text{apparent root-PSD} = \frac{\text{SD of data sample}}{\sqrt{\text{sample rate}}} \tag{5.1}$$

The results of these experiments are presented in Tables 5.2 to 5.6.

Reading Rate (Hz)	1000	100	10
Nom. Output (Hz)	4000	4000	4000
Gyro bandw. (Hz)	8800	8800	8800
Gyro range (°/s)	250	250	250
Accel bandw. (Hz)	1130	1130	1130
Accel range	2g	2g	2g
SD of sensor output			
X Accel (ms^{-2})	0.049	0.044	0.044
Y Accel (ms^{-2})	0.047	0.044	0.044
Z Accel (ms^{-2})	0.071	0.071	0.072
X Gyro (°/s)	1.613	1.715	1.724
Y Gyro (°/s)	4.129	4.065	4.127
Z Gyro (°/s)	4.704	5.035	5.016
Apparent root-PSD			
X Accel ($ms^{-2}/\sqrt{Hz}$)	0.0015	0.0044	0.0139
Y Accel ($ms^{-2}/\sqrt{Hz}$)	0.0015	0.0044	0.0139
Z Accel ($ms^{-2}/\sqrt{Hz}$)	0.0022	0.0071	0.0226
X Gyro (°/s/$\sqrt{Hz}$)	0.0510	0.1715	0.5451
Y Gyro (°/s/$\sqrt{Hz}$)	0.1306	0.4065	1.3051
Z Gyro (°/s/$\sqrt{Hz}$)	0.1487	0.5035	1.5861

Table 5.2: Static noise tests with no low-pass filter.

The results presented in Table 5.2 show that the SD of the unfiltered output does not (significantly) change for sampling rates of 10, 100 or 1000 Hz. This is not surprising as the nominal output rate without the DLPF is 4000Hz. The array cannot sample faster than 1000Hz. However, if the SD were also the same at 4000Hz then the apparent root-PSD would be roughly half that presented for 1000 Hz sampling rate, this would be well within the specified root-PSD for the accelerometer but 5× the specified noise for the gyroscope. This experiment implies that if unfiltered data is used it should be collected at as high a rate as possible.

There are two elements of the internal DLPF that can be set, the "nominal output" governed by one byte and the 'bandwidth' setting which is determined separately for the gyroscope and accelerometer with 8 levels each (3 bits), this is abbreviated to 'bandw.' in Tables 5.2 to 5.6. The nominal output rate is determined by

$$\text{Nominal output rate} = \frac{1000}{1+n} \text{ Hz} \tag{5.2}$$

where n is the setting of a single register byte, i.e. an integer from 0 to 255. If the DLPF is switched off the nominal output rate is 4000Hz.

What is not clear without experiment is the optimal way to use these filters. This is because changing the nominal output rate also changes the effective bandwidth. Selecting a low bandwidth filter (20Hz) and leaving the nominal output rate at maximum (1000Hz) while reading at different rates (Table 5.3) shows that the sensor SD is not affected by reading faster than the effective Nyquist frequency of the low-pass filter. However, if reading slower

Reading Rate (Hz)	83.33	40	20	10
Nom. Output (Hz)	1000	1000	1000	1000
Gyro bandw. (Hz)	20	20	20	20
Gyro range (°/s)	250	250	250	250
Accel bandw. (Hz)	20	20	20	20
Accel range	2g	2g	2g	2g
SD of sensor output				
X Accel (ms^{-2})	0.008	0.010	0.009	0.010
Y Accel (ms^{-2})	0.008	0.010	0.008	0.009
Z Accel (ms^{-2})	0.013	0.016	0.014	0.015
X Gyro (°/s)	0.064	0.063	0.064	0.065
Y Gyro (°/s)	0.059	0.060	0.059	0.060
Z Gyro (°/s)	0.051	0.051	0.051	0.052

Table 5.3: Static noise tests with 20Hz low-pass filter.

Reading Rate (Hz)	125	250	125	62.5	45.45
Nom. Output (Hz)	1000	125	125	125	125
Gyro bandw. (Hz)	92	92	92	92	92
Gyro range (°/s)	250	250	250	250	250
Accel bandw. (Hz)	92	92	92	92	92
Accel range	2g	2g	2g	2g	2g
SD of sensor output					
X Accel (ms^{-2})	0.020	0.016	0.015	0.015	0.016
Y Accel (ms^{-2})	0.020	0.016	0.015	0.015	0.016
Z Accel (ms^{-2})	0.026	0.025	0.024	0.025	0.025
X Gyro (°/s)	0.132	0.078	0.078	0.076	0.076
Y Gyro (°/s)	0.118	0.074	0.073	0.072	0.073
Z Gyro (°/s)	0.105	0.065	0.063	0.064	0.065

Table 5.4: Static noise tests comparing filter output rate and actual sampling rate.

than the Nyquist frequency, e.g. Table 5.4 reading at 125Hz with at 92Hz bandwidth then reducing the output rate to match the reading rate, does reduce the noise. Although reading at rates other than the sensor output rate does not affect the SD of the results (also Table 5.4), it was observed that reading faster does result in duplicate identical readings.

Noise characterisation results for the 8 possible bandwidth settings are shown in Table 5.5. As might be expected during this *static test*, the lower the bandwidth the lower the noise SD. A more interesting result is that at the bandwidth settings that are much too high to be readily seen by the 125Hz output and reading rate, there is still considerable difference in the resulting SD, which is consistent with smoothing reducing the noise SD. It is clear therefore that reducing the nominal output rate of Equation 5.2 is *not* achieved by taking the average of the last $(n + 1)$ samples because in this case the bandwidth settings above $\approx$ 62Hz would have been indistinguishable. However, the difference in SD between reading at 125Hz with a nominal output of 125Hz and 1000Hz (1^{st} and 3^{rd} columns of Table

Reading Rate (Hz)	125	125	125	125	125	125	125	125
Nom. Output (Hz)	125	125	125	125	125	125	125	125
Gyro bandw. (Hz)	3600	184	92	41	20	10	5	250
Gyro range (°/s)	250	250	250	250	250	250	250	250
Accel bandw. (Hz)	1130	460	92	41	20	10	5	1130
Accel range	2g	2g	2g	2g	2g	2g	2g	2g
bandw. setting int.	0	1	2	3	4	5	6	7
SD of sensor output								
X Accel (ms^{-2})	1.629	0.020	0.015	0.010	0.008	0.006	0.004	0.020
Y Accel (ms^{-2})	1.203	0.020	0.015	0.010	0.008	0.005	0.004	0.020
Z Accel (ms^{-2})	0.188	0.033	0.024	0.017	0.012	0.009	0.006	0.033
X Gyro (°/s)	0.894	0.087	0.076	0.066	0.052	0.041	0.028	0.239
Y Gyro (°/s)	3.315	0.081	0.071	0.062	0.049	0.037	0.026	0.201
Z Gyro (°/s)	2.120	0.070	0.064	0.054	0.043	0.032	0.023	0.182
Apparent root-PSD								
X Accel ($ms^{-2}/\sqrt{Hz}$)	0.1457	0.0018	0.0013	0.0009	0.0007	0.0005	0.0004	0.0018
Y Accel ($ms^{-2}/\sqrt{Hz}$)	0.1076	0.0018	0.0013	0.0009	0.0007	0.0005	0.0004	0.0018
Z Accel ($ms^{-2}/\sqrt{Hz}$)	0.0168	0.0030	0.0021	0.0015	0.0011	0.0008	0.0006	0.0029
X Gyro (°/s/$\sqrt{Hz}$)	0.0799	0.0078	0.0068	0.0059	0.0047	0.0036	0.0025	0.0214
Y Gyro (°/s/$\sqrt{Hz}$)	0.2965	0.0073	0.0064	0.0056	0.0044	0.0033	0.0023	0.0180
Z Gyro (°/s/$\sqrt{Hz}$)	0.1896	0.0063	0.0057	0.0048	0.0038	0.0029	0.0021	0.0163

Table 5.5: Static noise tests with the 8 possible low-pass filter bandwidth settings.

5.4), implies that it is not simply discarding the intermediate readings. The bandwidth required from the IMU is application specific, although the inertial navigation equations are typically run at at least 100Hz in most navigation applications to calculate the attitude position correctly [1], implying that the filter bandwidth must be at least 50Hz. However, it is clear from this experiment that it should be matched to an appropriate output rate to optimise performance. For the experiments in the rest of this chapter a bandwidth of 92Hz is used, to make it applicable to a wide range of navigation applications including pedestrian motion.

In addition, static tests were conducted at different maximum output range settings, as shown in Table 5.6. As can be seen, the range setting makes no significant difference to the noise, the gyroscope is surprisingly even has slightly worse SD at the lowest range. The same 16-bit ADC is used at the different ranges to encode a higher possible range of outputs there is a loss of sensitivity as the range increases. This theoretically increases the level of quantisation noise. However, this effect is small and does not have a significant effect on this experiment. This is because the noise SD is considerably larger than the smallest increment of specific force/angular rate that can be measured, both of which are presented in Table 5.6. This experiment is not sufficient to determine if quantisation residuals are carried-over or discarded, see Section 4.8.2.4, this is left for future work, although it does show that they are insignificant.

In addition to the direct aims of the experiments presented here, several other observations about sensor noise performance can be made. The in-plane x- and y-axis sensors must be of

Reading Rate (Hz)	500	500	500	500
Nom. Output (Hz)	500	500	500	500
Gyro range (°/s)	250	500	1000	2000
Gyro bandw. (Hz)	92	92	92	92
Accel range	2g	4g	8g	16g
Accel bandw. (Hz)	92	92	92	92
SD of sensor output				
X Accel (ms^{-2})	0.0144	0.0143	0.0144	0.0145
Y Accel (ms^{-2})	0.0144	0.0144	0.0144	0.0145
Z Accel (ms^{-2})	0.0240	0.0236	0.0237	0.0237
X Gyro (°/s)	0.1235	0.1123	0.1104	0.1112
Y Gyro (°/s)	0.1117	0.1063	0.1045	0.1050
Z Gyro (°/s)	0.0993	0.0976	0.0970	0.0975
Apparent root-PSD				
X Accel ($ms^{-2}/\sqrt{Hz}$)	0.00065	0.00064	0.00064	0.00065
Y Accel ($ms^{-2}/\sqrt{Hz}$)	0.00065	0.00064	0.00065	0.00065
Z Accel ($ms^{-2}/\sqrt{Hz}$)	0.00108	0.00106	0.00106	0.00106
X Gyro (°/s/$\sqrt{Hz}$)	0.00552	0.00502	0.00494	0.00497
Y Gyro (°/s/$\sqrt{Hz}$)	0.00500	0.00475	0.00467	0.00469
Z Gyro (°/s/$\sqrt{Hz}$)	0.00444	0.00436	0.00434	0.00436
Quantisation level (calculated from [23])				
Accel (ms^{-2}/LSB)	0.0006	0.0012	0.0024	0.0048
Gyro (°/s/LSB)	0.0076	0.0153	0.0305	0.0610

Table 5.6: Static noise tests with the 4 possible sensor range settings.

a different design to the out-of-plane z-axis sensors, this will lead to different performance characteristics, see Section 6.3.3. It is clearly the case that the noise performance is not the same for the different axes. The z-axis accelerometers have approximately 160% the noise SD of the other two axes, and the z-axis gyroscopes has approximately 85% the noise SD of the other two axes. Also the y-axis gyroscope has slightly less noise than the x-axis gyroscope, which is more surprising.

Based on the results of this section, the IMU data, used in the rest of this chapter and Chapter 6, are collected at 250Hz with the output rate set to match and both sensors internal DLPF bandwidths are set to 92Hz.

5.2 Initial user-calibration experimental results

In order to test the calibration procedures described in Section 4.2 measurements several different experiments were conducted, during which complete examples of each of the four sets of calibration procedures, were recorded using the array of 14 MPU9250 IMUs described in Section A.2. This data was recorded on Newport RP Reliance optical table, which is constructed to be particularly flat ("±0.1mm over 600mm square" [183]) and was close to

level. The 'corner' was constructed by attaching 3 bolts to the table which is an effective way to create a corner for the calibration cube (see Figure 4.11).

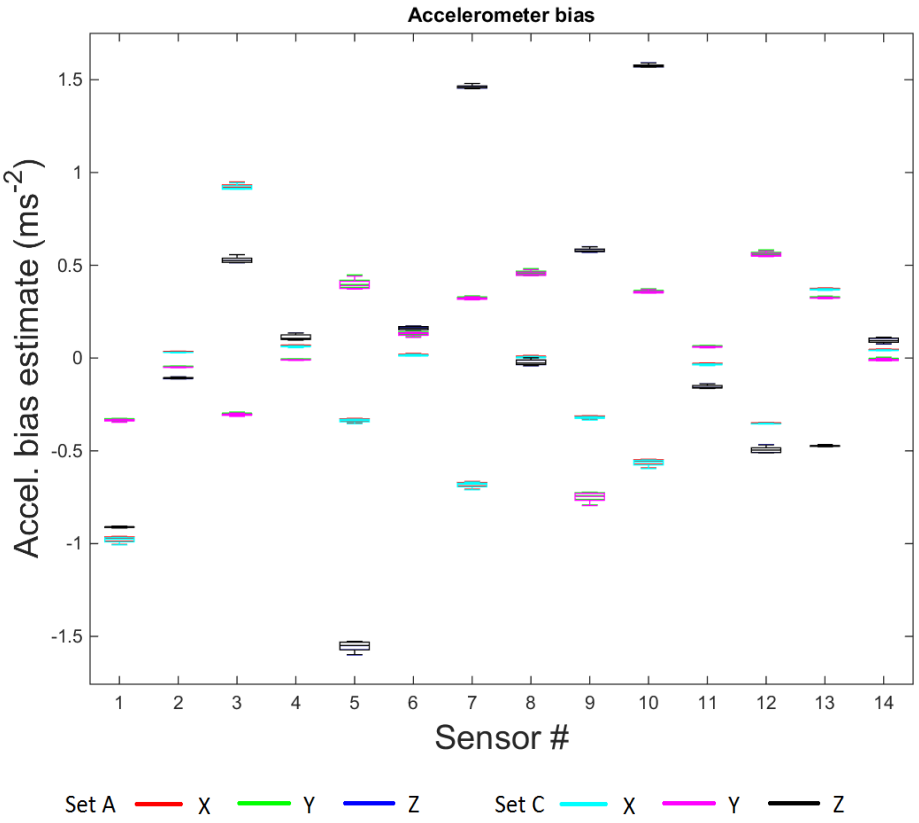


Figure 5.2: The spread of accelerometer bias estimates over 12 calibration procedures. On this scale the results of set A and C are so close as to be almost indistinguishable.

In order to save time over manual identification of start and finish points, the epochs representing the beginning and the end of the static periods were identified with a simple stationarity detection algorithm. This algorithm compares the range and SD of values of the next or previous second of data with threshold values to find the start or end of the static period, respectively. The manoeuvres conducted could then be easily identified using the uncalibrated (raw) sensor outputs. The accelerometer output for identification of the face, and a simple sum of angular increments over the period between the identified static periods for discriminating between 90°, 360° and other rotations. This algorithm also detects gross human errors (e.g. mixing up the order), these are then manually corrected, although a future production algorithm could do this automatically. This algorithm is not a focus of this work or its contribution, so it shall not be discussed further.

Human errors in conducting the manoeuvres (e.g. rotation in the wrong direction) or misidentification of manoeuvres are readily detectable in the results and even an automatic check of the results would remove the vast majority of them. An example is diagonal entries of $\widehat{\mathbf{A}_\mathbf{g}}$ which are ≈ 0.5 when ≈ 1.0 would be expected, caused for example by mixing up the axis of rotation.

Separate data sets were recorded a different output ranges, which will be presented in Section 5.4, this section assesses the consistency of the estimates using data recorded for

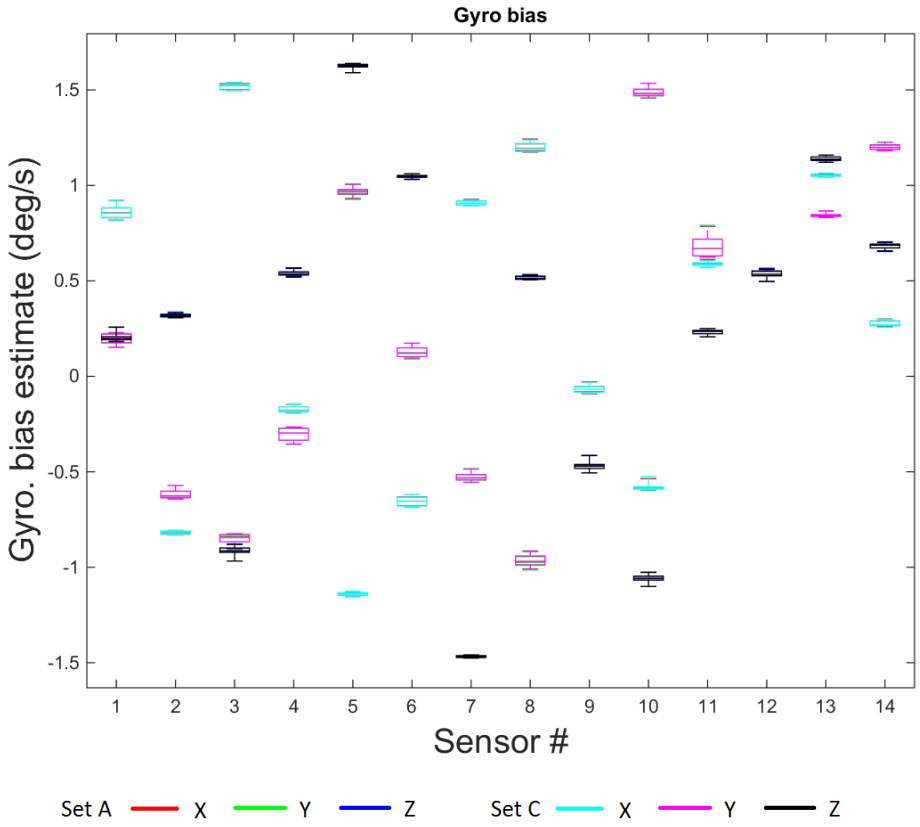


Figure 5.3: The spread of gyroscope bias estimates over 12 calibration procedures. On this scale the results of set A and C are so close as to be almost indistinguishable.

12 complete calibration procedures with the accelerometers and gyroscopes set at ±2g and ±250 °/s, respectively.

12 complete sets of the calibration procedures were recorded, to examine any variation induced by stochastic sensor errors and user or manoeuvre related errors. Figures 5.2 to 5.7 show the spread of the IMU error estimates obtained from 12 complete calibration procedures as box-and-whisker diagrams. In these plots, the outer line represents the range, the box the inter-quartile range and the central line the median. The x-axes show the different sensors and the y-axes the numerical results. The different colours represent the different errors of that type for the particular sensor, e.g. x-, y- and z-axis gyroscope bias. Summary statistics for the experimental repeatability are presented in Table 5.7. These are SDs across the estimation of a single IMU error parameter, rather than the statistics of the distribution of that parameter among the 14 sensors which will be presented in Table 6.1, and discussed in Section 6.6.

A close spread of calibration results shows that the calibration is consistent. This means that it is not sensitive to stochastic sensor errors or how well the user conducts the procedure. However, consistency is not the same as accuracy, as was discussed in Sections 2.5.1 and 4.1. Any biases in the calibration procedure due to the uncalibrated values of the INS's systematic errors, the construction accuracy of the calibration cube, or levelness of the table will not be seen. The results of Section 4.8.2 show that the first of these will not be significant, and Section 4.8.1 suggests that the latter two of these could be significant but

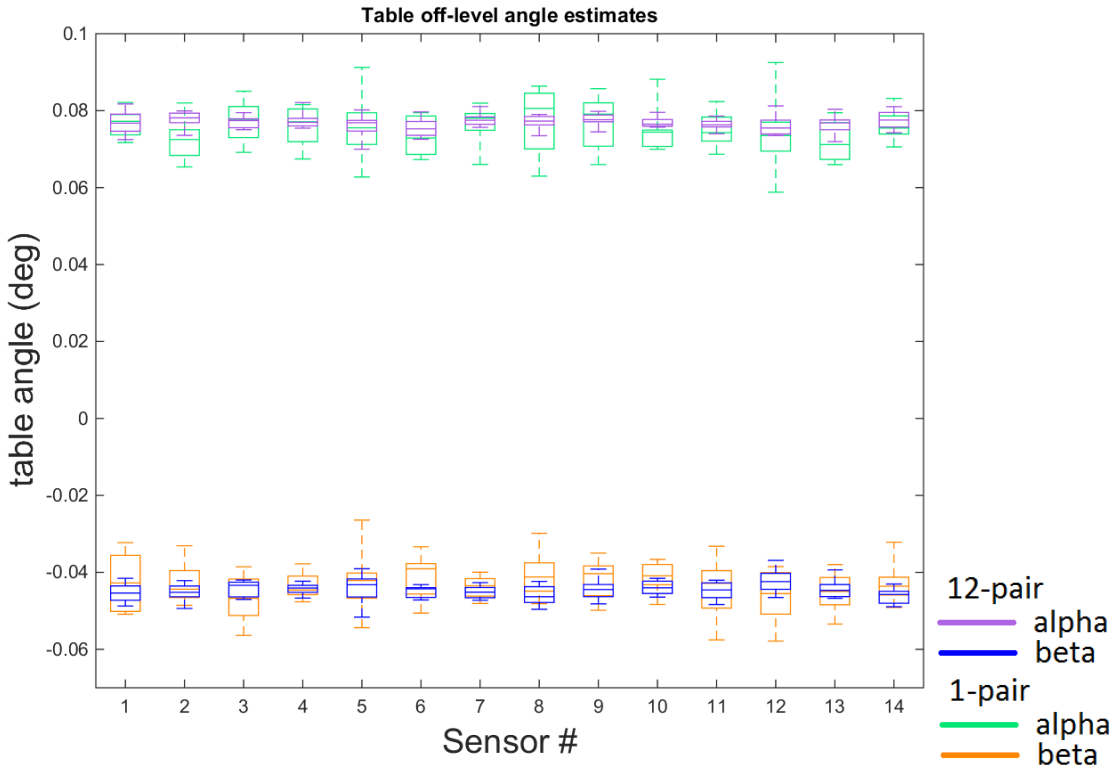


Figure 5.4: The spread of table angle estimates over 12 calibration procedures.

the values expected on an *optical table* and with a precisely made box this is less likely, but this will be examined in Section 5.3.

In Figures 5.2 and 5.3, the accelerometer and gyroscope biases are presented. The colours red, green and blue are used for the six-position (set A) results and cyan, magenta and black for the 24-position (set C) results, for the x- y- and z-axis sensors respectively. As can be seen the spread of each sensor's bias estimates is much smaller than the set of possible biases, which is a reassuring result. The specified bias performance is ±60 milli-g (x and y, ≈ 0.59 ms^{-2}) and ±80 milli-g (z, ≈ 0.79 ms^{-2}) for the accelerometer and ±5 °/s for the gyroscope [23]. If one believes the result of this calibration then for this sample of 14 sensors the accelerometers are slightly under-performing but the gyroscopes are significantly better than specified.

As can be seen in nearly all cases, the 6-position and 24-position results are virtually identical. This in conjunction with the results of Chapter 4 (particularly Section 4.8.1.2) would imply in that there is no significant effect from the table's deviation from level. The estimates of the two table angles (deviation from level, see Section 4.3.6) from each sensor's readings presented in Figure 5.4 show consistently, both across the 12 calibrations and the 14 sensors that the table is indeed very close to level. The angle α being represented by the violet (12-pair) and turquoise (1-pair) being approximately 0.075° and β being represented by the blue (12-pair) and orange (1-pair) being approximately -0.045°. It should be noted that the table angle estimation method (see Section 4.5.1 and Equation 4.45) uses the uncalibrated sensor outputs, so the correlation between sensors is also an indication of table angle estimate accuracy.

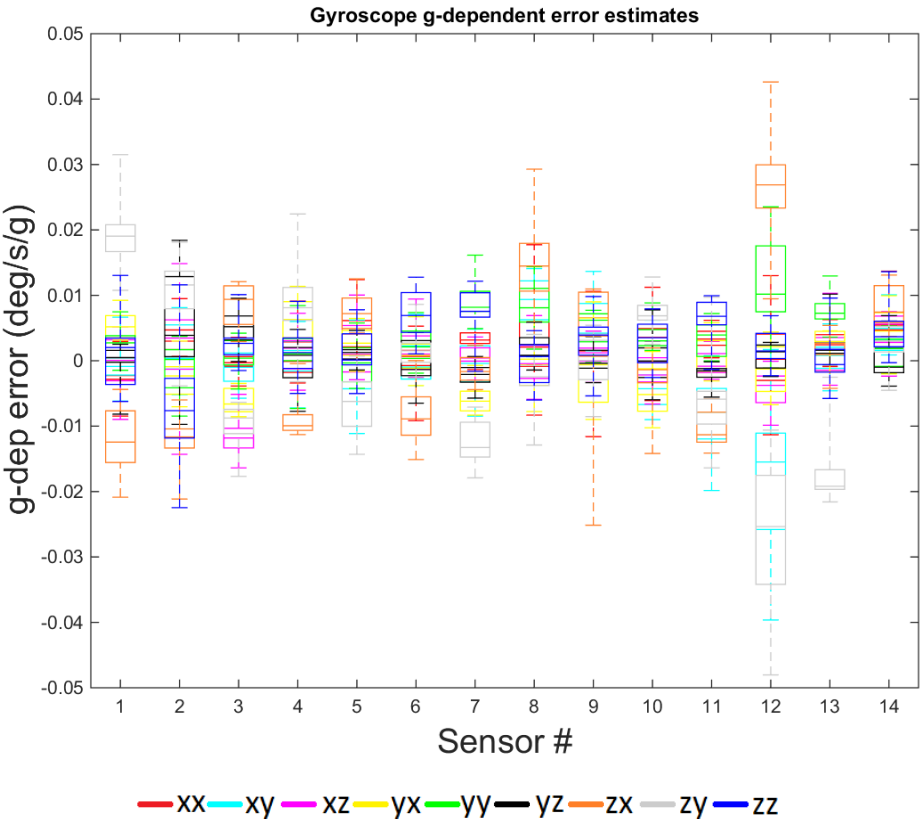
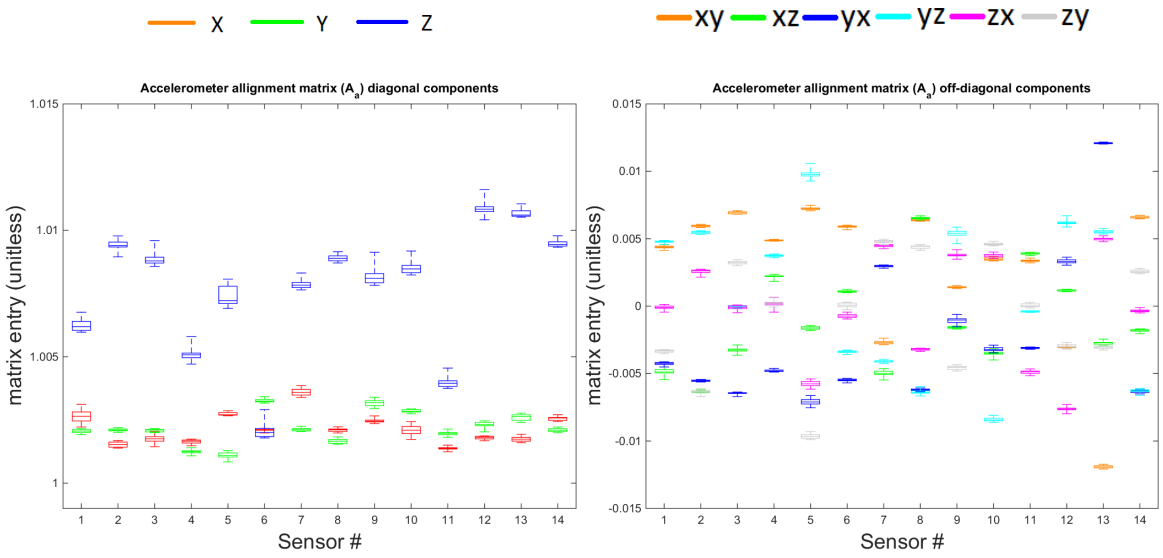


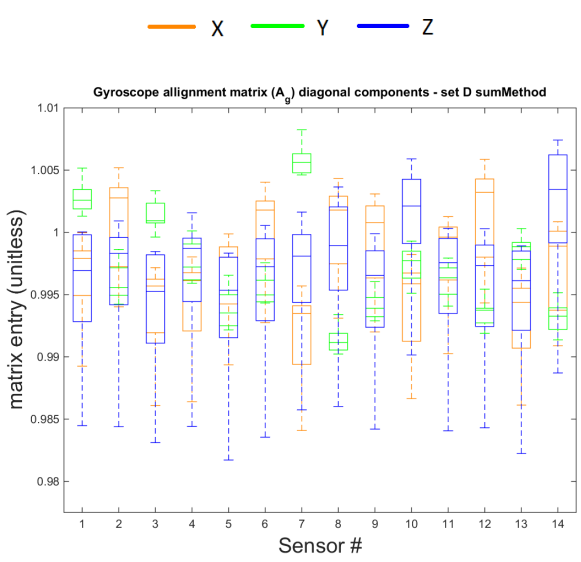
Figure 5.5: The spread of gyroscope g-dependent error estimates over 12 calibration procedures.



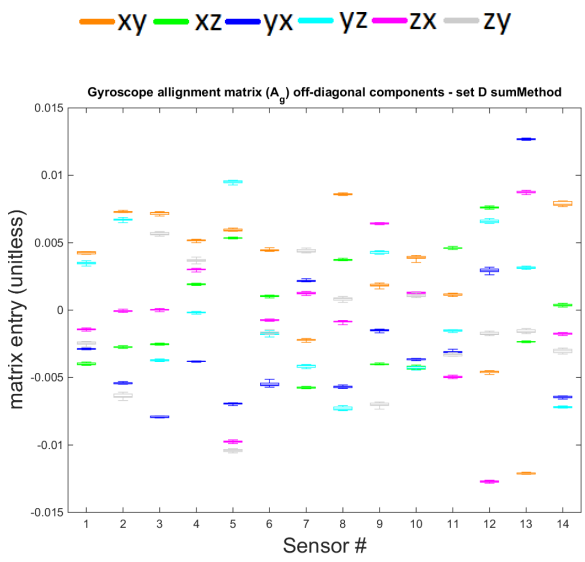
(a) Diagonal elements

(b) Off-diagonal elements

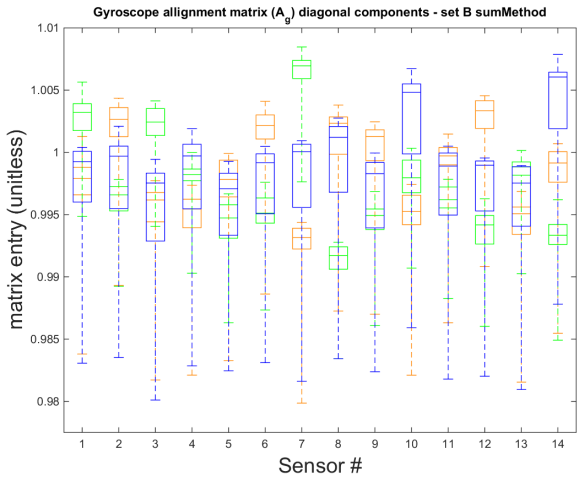
Figure 5.6: The spread of accelerometer alignment and scale factor estimates over 12 calibration procedures.



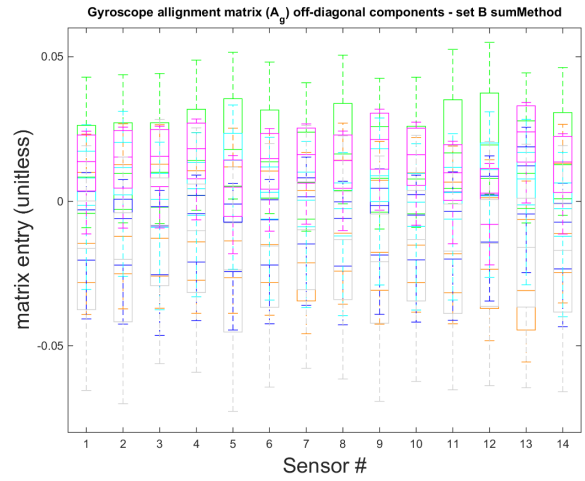
(a) Diagonal elements, set D



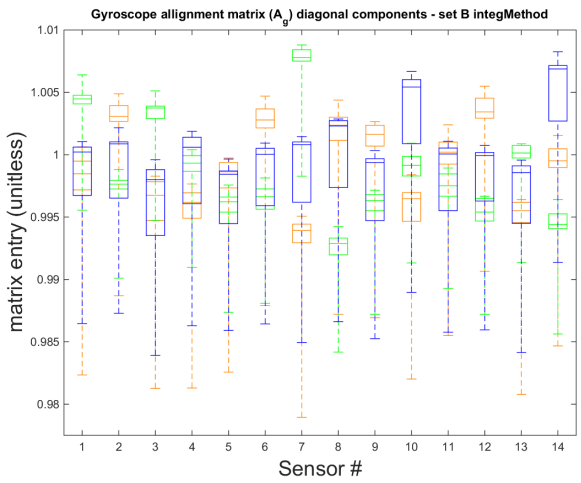
(b) Off-diagonal elements, set D



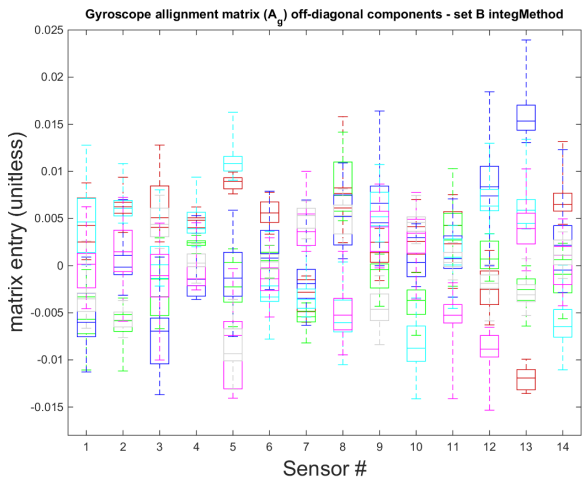
(c) Diagonal elements, set B



(d) Off-diagonal elements, set B



(e) Diagonal elements, set B integ. method



(f) Off-diagonal elements, set B integ. method

Figure 5.7: The spread of gyroscope alignment and scale factor estimates over 12 calibration procedures, using set D and B manoeuvres.

Coefficient	Repeatability: SD over 12 experiments (averaged over 14 sensors)								
	X (A)	Y (A)	Z (A)	X (C)	Y (C)	Z (C)			
$\boldsymbol{b_a}$ (ms^{-2})	0.0064	0.0080	0.0105	0.0064	0.0080	0.0106			
$\boldsymbol{b_g}$ (°/s)	0.0186	0.0305	0.0152	0.0187	0.0306	0.0151			
	xx	xy	xz	yx	**yy**	yz	zx	zy	**zz**
$\mathbf{A_a}$ (C)($\times 10^{-3}$)	**0.1107**	0.0892	0.1290	0.1158	**0.0886**	0.1392	0.1595	0.1405	**0.2551**
$\mathbf{A_a}$ (A)($\times 10^{-3}$)	**0.1239**	0.1023	0.1379	0.1427	**0.1007**	0.1395	0.1834	0.1549	**0.2515**
$\mathbf{A_g}$ (D)	**0.00391**	8.98e-05	6.64e-05	8.01e-05	**0.00127**	9.66e-05	7.53e-05	1.27e-04	**0.00587**
$\mathbf{A_g}$ (B sum)	**0.0042**	0.0209	0.0169	0.0161	**0.0027**	0.0181	0.0113	0.0259	**0.0065**
$\mathbf{A_g}$ (B int)	**0.0045**	0.0020	0.0021	0.0033	**0.0026**	0.0026	0.0029	0.0016	**0.0057**
$\mathbf{G_g}$ (C)(°/s/g)	0.0036	0.0044	0.0035	0.0032	0.0032	0.0030	0.0045	0.0050	0.0041
$\mathbf{G_g}$ (A)(°/s/g)	0.0044	0.0049	0.0048	0.0044	0.0042	0.0055	0.0048	0.0055	0.0047
Table	12-pair α	12-pair β	1-pair α	1-pair β					
off-level (°)	0.0022	0.0019	0.0053	0.0055					

Table 5.7: Summary statistics of experimental repeatability for the estimation of each IMU error coefficient over 12 recorded experiments. SD recorded is mean SD over the 14 sensors. $\mathbf{A_a}$ and $\mathbf{A_g}$ scale factors in bold.

Figure 5.5 shows the results of the g-dependent error matrix estimates. This error source is not specified on the MPU9250's datasheet [23], so to include it in Chapter 4 it was assumed to be the distributed with an SD of 0.1 °/s/g ($\approx$ 1.78e-04 rad/s / ms^{-2}), based on the specifications of the similarly priced Bosch BMI055 [45]. The estimates of the results are around 0.01 - 0.02 °/s/g, so it would appear that the MPU9250s perform much better than was assumed in the simulations of of Chapter 4.

Figure 5.6 presents the estimates of the accelerometer alignment-and-scale-factor matrix ($\widehat{\mathbf{A_a}}$). Note that the for the diagonal elements (Figure 5.6a) the x- and y-axis sensors (red and green) have a much tighter distribution across the sensors, and a lower mean, than the z-axis sensors (blue). This points to a difference in design or construction as discussed in Section 6.3.3, the slightly greater spread is likely due to the increased noise on the z-axis sensor (see Section 5.1). The fact that all the calibrated values for the diagonal elements are greater than one could be explained by different values of g used by the calibration procedure (g= 9.80665 ms^{-2}) and the true value of g in southern England, 9.811832 ms^{-2} [184, 185], and the unknown g value used by the manufacturer when they quote the sensitivity in LSB/g.

Figure 5.7 presents the estimates of the gyroscope alignment/cross-coupling and scale-factor matrix ($\widehat{\mathbf{A_g}}$). There are three different methods presented: set D using 6-rotations one per face while maintaining contact with the table (Figures 5.7a and 5.7b) ; set B sum method which uses 3 rotations (one per axis) conducted in the air by hand and uses a correction based on sum of angular increments (Figures 5.7c and 5.7d); and set B integrated method which uses the same 3 rotations but runs the attitude update part of the inertial navigation equations (Figures 5.7e and 5.7f). These methods are fully detailed in Section 4.5.2. It should be noted that all these method are estimating of the same quantities and the colouring of the matrix components are the same to allow a direct comparison. Note also, the set D estimate is completely independent of the other two as it uses different recorded data.

The simulations in Chapter 4 concluded that errors in $\widehat{\mathbf{A_g}}$ were dominated by errors in the procedure and noise, thus consistent results are likely to represent accurate results.

First consider the diagonal elements of $\widehat{\mathbf{A_g}}$ shown in Figures 5.7a, 5.7c and 5.7e. Note that the vertical axes have the same scale and that the xx- yy- and zz-components are represented by orange, green and blue. It can be seen that all the methods reach approximately similar answers for the individual sensors, e.g. for sensor 7 all the methods have a mean of about 1.006 for the yy-component (green). The two set B methods behave similarly but both are not as good as the set D method. Overall, none of the methods could be said to be markedly better, as the true values are not known.

In the case of the off-diagonal elements, there are much more significant differences. Note that the vertical axis scales are not the same. The six-rotation (set D) method produces remarkably consistent results (Figure 5.7b). The set B integrated method in Figure 5.7f seems to produce broadly similar results to set D note the yx-component (blue) of sensor 13 at approximately 0.013 in set D and 0.015 in set B. However, this figure shows markedly less consistent results than set D, and might only be considered a marginal improvement on no calibration. For the set B sum method, in Figure 5.7d, the estimates are much less consistent. Note the change of scale shows that the spread of results is around 5× that of either of the other two. The results with this method appear almost uncorrelated with those of the other two, and the range of results so broad that the sum method for set B is shown not to work. Also note that the hybrid method (see Section 4.5.2.4), which uses the off-diagonal elements of $\widehat{\mathbf{A_a}}$ as an approximation for those of would certainly work better than the sum method as can be seen by comparing Figure 5.6b with Figure 5.7b.

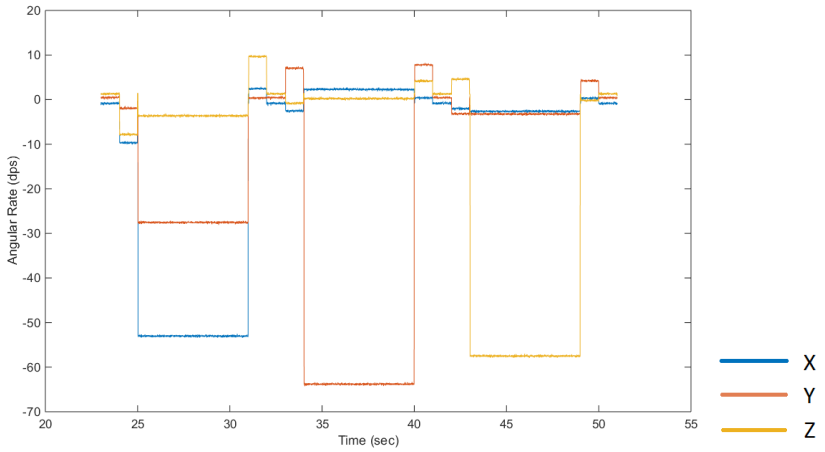


Figure 5.8: Angular rate during a simulated example of manoeuvre set B.

To understand why this difference between the two set B methods was not shown in Chapter 4, it should be noted that this difference arises from the non-commutativity of rotations. Briefly, the reality of conducting a rotation in mid-air makes it very difficult to rotate about a single fixed axis. Even simulating that rotation axis not precisely the intended axis does not adequately capture the complexity of a real free-hand rotation, as can be seen by comparing the simulated Figure 5.8 with the real angular rate output in Figure 5.9. As can be seen the proportions of the angular rates about the different axes are not fixed, which implies the instantaneous axis of rotation is not fixed, thus it becomes very important to treat the

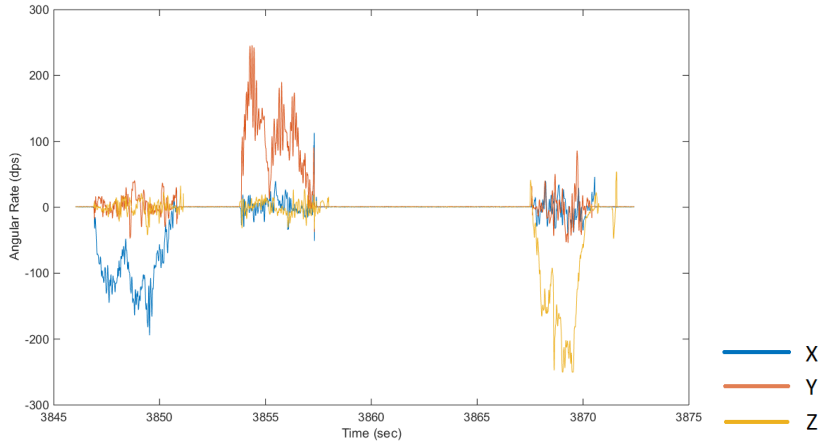


Figure 5.9: Angular rate during a real example of manoeuvre set B.

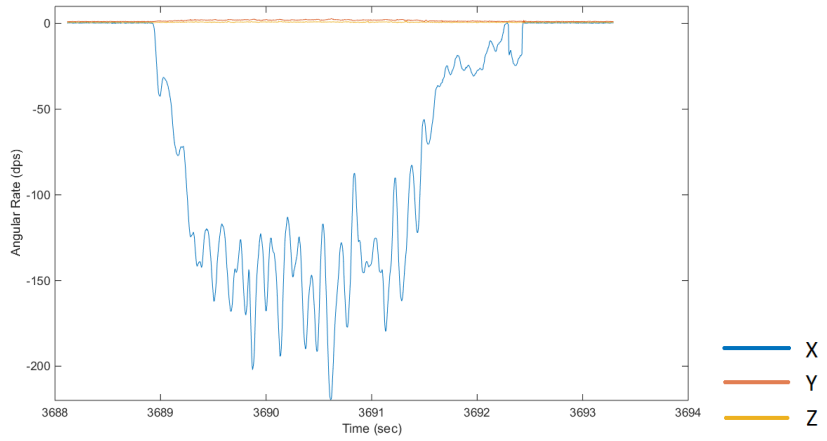


Figure 5.10: Angular rate during a real example of one part (2^{nd}) of manoeuvre set D.

rotations correctly.

In contrast, keeping the box in contact with the table, as in manoeuvre set D, produces a very consistent rotation axis. This can be seen in Figure 5.10. This is real measured angular rate during one of the set D manoeuvres the rotation is primarily about the x-axis (blue) but a small rotation about y (red) can be seen at a couple of degrees per second, roughly in proportion to the x-axis angular rate. This means that the axis of rotation is almost fixed so the fact that only the sum-of-angular-increments method, rather than the integrated method, is used to calculate the result does not matter.

5.3 Simulations to determine expected consistency and accuracy of calibration procedure results

It is possible to test the validity of the simulations presented in Chapter 4 by comparing the simulated results with the empirically obtained results. However, the default settings used in Chapter 4 were chosen to represent a typical navigation system housing and a roughly level surface (e.g. dining room table), rather than the high-precision calibration frame and optical table used in this Chapter.

ODR	200 Hz
Wait Time	2 sec
cube face-error	0.0438°
cube half-length	0.0375m
table angle	0.1°
replacement Head Error	1° all sets
position error	0.5mm all sets
rotation time	6 sec
part B midair start Att Error	10°
part B midair axis Error	8°
part B midair spin Error	10°

Table 5.8: The parts of 'TESTspec' changed from default, see Section 4.6.1.

In Chapter 4, it was shown that the initial distribution of the quantities being calibrated had very little or no effect on the accuracy with which they were estimated. However what did have the most significant effect on the accuracy was the cube's angular accuracy followed by the table's levelness and the gyroscope noise. Thus to run a simulation that realistically represents the experiments presented in this chapter these are the most important factors to get correct.

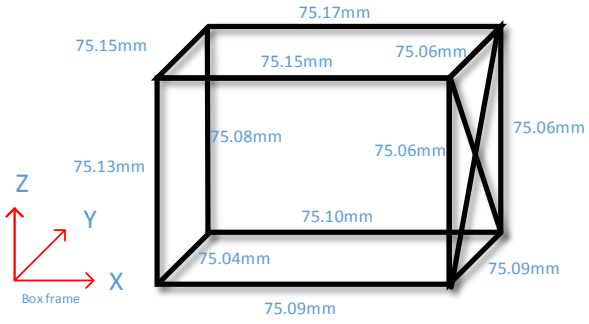


Figure 5.11: The dimensions of the aluminium calibration frame as manufactured.

The calibration cube was precision manufactured. However, it is not perfectly cubic. In the absence of a better technique for measuring its deviation from cubic its twelve edges were measured with digital vernier callipers accurate to ±0.01mm. This is shown in Figure 5.11. The edges have mean length 75.1017 mm and SD 0.0406 mm. So the difference between two lengths has SD $\sqrt{0.0406^2 + 0.0406^2} = 0.0575$mm and if they are 75.1017 mm apart then the angular SD is about $\arcsin(\frac{0.0575}{75.1017}) = 0.0438°$.

The table angles were consistently estimated by the calibration procedure to be $\alpha = 0.075°$ and $\beta = -0.045°$ (see Figure 5.4 and Table 5.7). These and the other values presented in Sections 5.1 and 5.2 lead to changing some of the default settings used in Chapter 4 to the values presented in Tables 5.8 and 5.9.

There are a number of assumptions upon which the simulation is based. Some of these are

Gyro noise PSD	$0.01\,^\circ/\mathrm{s}/\sqrt{\mathrm{Hz}}$
Gyro bias SD	$1.5\,^\circ/\mathrm{s}$
Gyro g-dep SD	1×10^{-5} rad / ms^{-2}
Accel noise PSD	300 micro-g/$\sqrt{Hz} \approx 0.0029\mathrm{ms}^{-2}/\sqrt{\mathrm{Hz}}$)
Accel bias SD	0.6 ms^{-2}

Table 5.9: The parts of 'IMUspec' changed from default, see Section 4.6.1.

also assumptions of the calibration method. For example, the IMU model assumes that the IMU's systematic errors comprise a particular set of factors (see Equation 4.5 to 4.8) are their coefficients have a fixed value, and sensor noise is white. These assumptions are valid in the simulation, by design, and may or may not sufficiently close to valid in reality that they have a significant effect on the results. There are also calibration algorithm assumptions, such as non-linearity being insignificant, which may be less valid for the real data than the simulation, for instance the variation in angular rate during the set B or D manoeuvres is greater in the real case (see Figures 5.9 and 5.10) than in the simulation (see Figure 5.8), which excites gyroscope non-linearity more. There are also some limitations in how well the simulated manoeuvres reflect a real calibration procedure. For example in the set B simulation the rotation axis is effectively fixed rather than instantaneously varying, see Figures 5.8 and 5.9 and the discussion in Section 5.2, which makes the sum method behave better in the simulation than reality. Most of these effects would cause the simulation to appear more accurate/repeatable than the real experiment.

Noting these provisos, in this section, two simulations are presented. The intention of these simulations are: first, to validate the simulation method in Chapter 4 by comparing the repeatability derived from a simulation with the experimental repeatability presented in Section 5.2; second, to estimate the accuracy of the experimental calibration, under the assumption that the simulation is valid.

In order to test the validity of the simulation by comparing simulated to experimental repeatability, one simulation was required, where all the IMU errors (i.e. the values of the systematic errors) were fixed and many of the test errors were also fixed (e.g. the deviation from cubic of the calibration box, and the table levelling error), to simulate repeatedly calibrating the same IMU in the same box. A second simulation is presented with the normal methodology simulating many different IMUs with the same specifications being calibrated in different boxes, to estimate the accuracy of the calibration procedure. The latter of these is a repeat of the work in Chapter 4 with a different set of inputs.

5.3.1 Simulation estimated repeatability

The central column of Table 5.10 presents the results of a simulation were many of the randomly generated errors were fixed to a single (randomly generated) value for all the 500 simulation runs. These comprised all the IMU's systematic errors, the cube's deviation from cubic and its length error, the IMU-within-the-box mounting error, and the table's deviation-from-level error. This means that the differences that are observed between the

IMU error	Repeatability (SD)	Accuracy (RMS)
$\widehat{\boldsymbol{b_g}}$ set A T1	0.00287 °/s	0.00302 °/s
$\widehat{\boldsymbol{b_g}}$ set C T1	0.00145 °/s	0.00145 °/s
$\widehat{\boldsymbol{b_g}}$ set A T2	0.00513 °/s	0.00513 °/s
$\widehat{\boldsymbol{b_g}}$ set C T2	0.00250 °/s	0.00250 °/s
Uncalibrated $\boldsymbol{b_g}$	–	1.49 °/s
$\widehat{\boldsymbol{b_a}}$ set A bias T1	0.00111 ms^{-2}	0.0101 ms^{-2}
$\widehat{\boldsymbol{b_a}}$ set C bias T1	0.000427 ms^{-2}	0.00626 ms^{-2}
$\widehat{\boldsymbol{b_a}}$ set A bias T2	0.00148 ms^{-2}	0.0174 ms^{-2}
$\widehat{\boldsymbol{b_a}}$ set C bias T2	0.00074 ms^{-2}	0.0173 ms^{-2}
Uncalibrated $\boldsymbol{b_a}$	–	0.681 ms^{-2}
$\widehat{\mathbf{A_a}}$ set A	0.000196	0.00152
$\widehat{\mathbf{A_a}}$ set C	0.0000759	0.00112
$\widehat{\mathbf{A_a}}$ 7-pos.	0.000196	0.00152
Uncalibrated $\mathbf{A_a}$	–	0.0661
$\widehat{\mathbf{G_g}}$ set A	0.00498 °/s/g	0.00540 °/s/g
$\widehat{\mathbf{G_g}}$ set C	0.00252 °/s/g	0.00317 °/s/g
$\widehat{\mathbf{G_g}}$ 7-position	0.00498 °/s/g	0.00540 °/s/g
Uncalibrated $\mathbf{G_g}$	–	0.00563 °/s/g
$\widehat{\mathbf{A_g}}$ set B sum	0.0930	0.1066
$\widehat{\mathbf{A_g}}$ set B int	0.0114	0.0291
$\widehat{\mathbf{A_g}}$ set B hyb	0.00865	0.0281
$\widehat{\mathbf{A_g}}$ set D sum	0.00105	0.00164
$\widehat{\mathbf{A_g}}$ set D hyb	0.000930	0.0232
Uncalibrated $\mathbf{A_g}$	–	0.0656
table angle (12-pair)	0.00693°	0.00445°
table angle (1-pair)	0.0112°	0.0104°
uncalibrated table angle	–	0.0999°

Table 5.10: Simulation-derived consistency measures (standard deviations of estimates) and accuracy measures (RMS of estimates). SD and RMS results derived from separate simulations. SD or RMS for each axis or matrix entry is then averaged across the set for a single number.

calibration runs are the result of IMU noise, and human-related inaccuracies: the heading-replacement-error and inaccuracies in the way the dynamic rotation was carried out.

As discussed in Chapter 4, the main effect of the getting the heading-replacement wrong in the static parts of the calibration is to reduce the effectiveness of the 24-position manoeuvre set C's calibration of the effect of the table error from level. In this case this error is very small so less accurately calibrating it has very little effect. If one observes the difference in SD for each statically-determined error and estimation technique in Table 5.10 the SD is with set C approximately half that for set A in every case due to the 4× longer averaging time, see Section 4.8.1.4, rather than any additional improvement from minimising the effect of the table angles.

Comparing the simulation derived SDs (Table 5.10) with the consistency results presented in Section 5.2 (Table 5.7). The simulation predicts gyroscope bias SDs at around 0.003 °/s, the experimentally observed variation of this was more around the 0.015 to 0.03 °/s level (Table 5.3). For the accelerometer bias the model predicted SD is around 0.001 ms^{-2} this is also lower than can be observed in Figure 5.2 but not by much. The $\widehat{\mathbf{G_g}}$ SD predicted by the model is 0.03 - 0.055 °/s/g, this is a very good match for the levels of variation shown in Figure 5.5. Similarly the experimental results for $\widehat{\mathbf{A_a}}$ are very similar, 0.0009 - 0.00025 SD, to the predicted 0.0001 - 0.00015 SD (unitless), see Figure 5.6. For most components except the zz-component about half as much variability is observed. This better performance could be attributed to the actual (x and y) noise being lower than modelled.

A possible explanation for this is that the gyroscope and accelerometer biases are not fixed at that level for an hour and/or the unavoidable temperature variation over that time (see Figure 5.12) lead to larger changes in the biases. Note that the internal IMUs' internal temperature sensors are only specifications are only given as typical values ("Sensitivity Untrimmed 333.87 LSB/°C, Room Temp Offset (21°C) 0 LSB" [23]) so it is probable that the differences between the outputs of the internal temperature sensors are biases. The effect of temperature on the sensors in the testbed will be examined in Section 6.6.2.

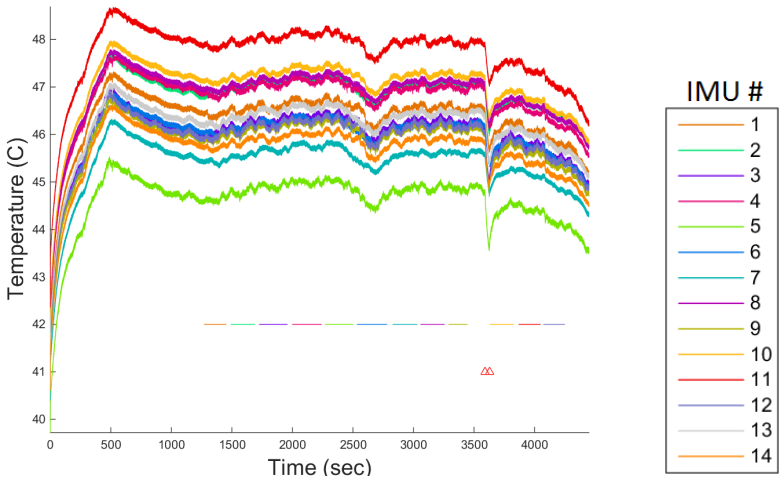


Figure 5.12: The output of the internal temperature sensors during the experiment. Note that the first calibration procedure began at 1273 seconds to allow the sensors to heat up. The last procedure used finished at 4281 seconds. The time during which each of the 12 calibration procedures took place are marked by coloured horizontal lines. The walking test (see Section 5.6.2) start and finish are marked by red triangles. The sharp dip in temperature coinciding with the walking test, may be attributed to convective cooling due to increased airflow over the PCB while moving.

When considering the repeatability of $\widehat{\mathbf{A_g}}$, one needs to consider the problems highlighted above, i.e. the simulation not capturing the full complexity of the motion. However, considering the set D 'advanced' manoeuvres, the simulation estimates repeatability that is very close to that demonstrated in the experiments shown in Figure 5.7, the simulation predicted SD 0.001 (unitless) is about half the variation of the diagonal elements and more than double that exhibited by the off-diagonal elements, so a very good estimate. The $\mathbf{A_g}$ estimates from the set B 'integrated' method in the simulation are at around the 0.03 (3%

level), Figure 5.7e and 5.7f show that the experimental variation is around the 1% level, but the simulation is reasonable given the issues with the capturing of the complexity discussed earlier.

Finally the simulated table angle estimate SDs come out as 1×10^{-4} or 2×10^{-4} radians, equivalent to 0.006 to 0.012 degrees, for 1-pair and 12-pair, respectively. Comparing these results to Figure 5.4, one can see that there is a good agreement.

5.3.2 Simulation estimated accuracy

If one takes the agreement between the results presented in Table 5.10 and the experiments presented in Section 5.2 as enough evidence to believe the simulations, at least for the parts of the procedure other than set B, then one can use simulations to get an idea of the accuracy of the experiments.

One must expect any procedure to be less accurate than it is repeatable, as the accuracy includes both the random errors from noise or human sources, and any errors introduced from sources that are fixed when one repeats the experiments, such as deviations from cubic in the calibration box.

In order to simulate the *accuracy* of the experimental calibration a second simulation was run *without* the special modification used to set most of the errors to the same value. That is, the simulation was run in exactly the same way as those in Chapter 4 the only change being the input values in Tables 5.8 and 5.9.

The third column of Table 5.10 shows the simulation-derived RMS values for the accuracy of the systematic error estimates produced by the calibration algorithm. The initial or uncalibrated values are also shown, for comparison.

There are some results that are worth individually highlighting, first that the Technique 2 bias estimation methods (see Section 4.5.1 and Equation 4.67 and 4.69) are worse performing than the Technique 1 method for both the accelerometer and gyroscope biases. This is related to the cube angle error (deviation from orthogonal-sided), as was discussed in Section 4.8.1.1, specifically with reference to Figure 4.13a the Technique 2 method is better than the standard method when the cube angle error exceeds 0.25°, which is not the case here. Also note that the set B sum and hybrid methods are worse than no calibration.

If we look at the approximate level of improvement for each error under each of the different calibration sets. The gyroscope bias improves by 3 orders of magnitude for the 24-position set C method. The 6-position set A method has, as expected by the 4× longer averaging time, twice the RMS of the 24-position method, supporting the hypothesis that the noise averaging is the primary factor.

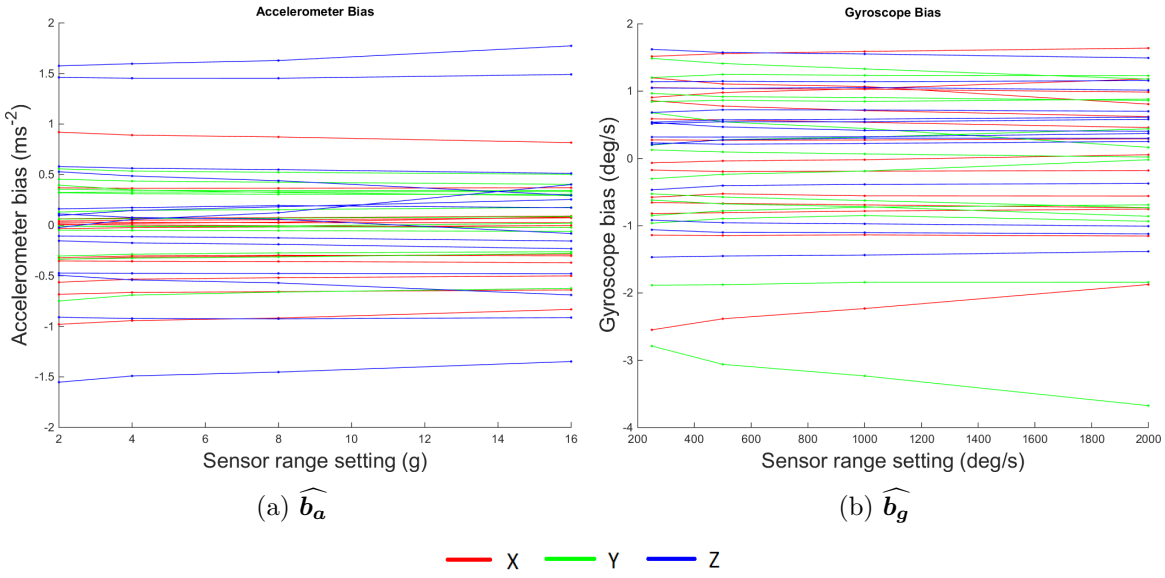
Similarly the accelerometer bias is improved by about 2 orders of magnitude and the set C performance is not twice as good as the set A performance (closer to 1.5× 0.0063 vs 0.010ms^{-2}), so there is still a small effect from removing the effect of the un-level table, even though it is very close to level (0.1° SD). The $\widehat{\mathbf{G}_\mathbf{g}}$ accuracy estimate shows the uncertainty is very close to the actual estimated values, so the calibration is only marginally better

than just ignoring the error (i.e. assuming that it is a matrix full of zeros). However, as around the repeatability SD is around 80% the value of the RMS, this uncertainty is clearly dominated by random errors (i.e. sensor noise). Thus improvements could be made by using longer averaging times, increasing the averaging time from 2 to 50 seconds would make the effect of the noise smaller, until the cube error was the dominant source of error, at the expense of taking a long time for the whole calibration.

The $\widehat{\mathbf{A_a}}$ is significantly improved by the calibration, and the set C method is only slightly better than the set A method, implying that the cube error is the most a significant factor, as the longer noise averaging time has very little effect. For the $\widehat{\mathbf{A_g}}$ errors the part B integrated method shows a small improvement on the no calibration.

The simulation results for the set D manoeuvres are very significantly better than uncalibrated (nearly 50× smaller RMS). The set D hybrid method, which uses the off-diagonal elements from $\widehat{\mathbf{A_a}}$, shows a much smaller improvement. This implies that the off-diagonal elements are well estimated by the set D sum method.

5.4 Effect of sensor range setting on the calibration output



(a) $\widehat{\boldsymbol{b_a}}$ (b) $\widehat{\boldsymbol{b_g}}$

Figure 5.13: The effect of accelerometer and gyroscope range settings on $\widehat{\boldsymbol{b_a}}$ and $\widehat{\boldsymbol{b_g}}$, respectively.

An experiment was carried out to assess the effect of the sensor range settings on the calibration procedure's output. This test is intended to show if there are significant differences in the sensor calibration parameters if the range settings are changed.

Four experiments were carried out to cover the four possible range settings for the accelerometer and gyroscope over a four-hour period. In each of these experiments multiple repeats of the calibration procedure were conducted. First, 5 complete procedures were recorded with the accelerometers set to 16g and the gyroscopes to 2000 °/s. Then, 3 procedures with range settings of 8g and 1000 °/s followed by 4 procedures with range settings of 4g and

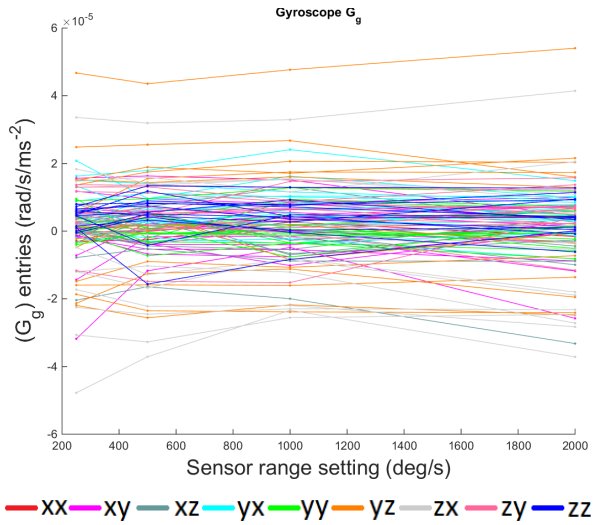
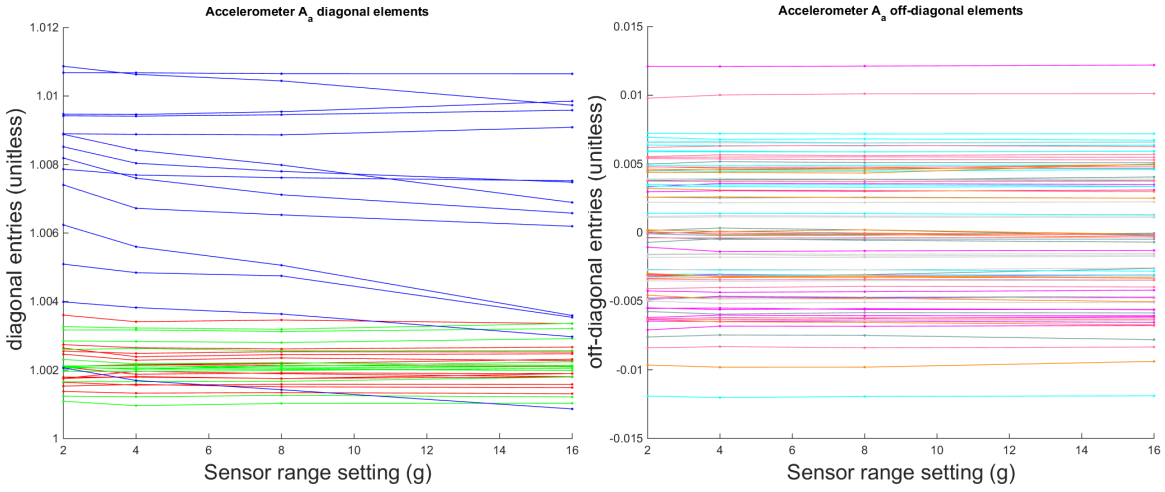


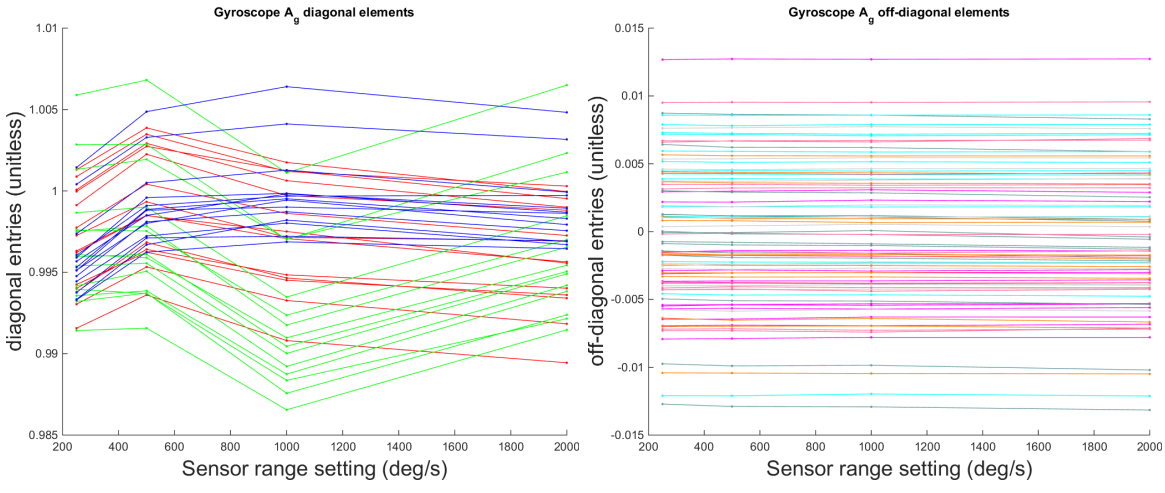
Figure 5.14: The effect of gyroscope range setting on g-dependent error.



(a) Diagonal elements

(b) Off-diagonal elements

Figure 5.15: The effect of accelerometer range setting on alignment and scale factor.



(a) Diagonal elements

(b) Off-diagonal elements

Figure 5.16: The effect of gyroscope range setting on alignment and scale factor.

500 °/s, for the accelerometer and gyroscope respectively. Finally 12 calibration procedures were recorded with 2g accelerometer range and 250 °/s gyroscope range.

In the results presented in Figures 5.13a to 5.16 the mean calibration output over the 3 to 12 procedures for each range setting is shown. In all cases, the results of the advanced method (using manoeuvre set C or D) are presented. The lines refer to all the outputs of the multiple axes across the multiple sensors. Lines of the same colour denote the same axis (or matrix entry) outputs. In all cases red, green and blue refer to the x, y and z (or xx yy zz) outputs, respectively.

Figure 5.13a shows that there is only a limited change in each accelerometer's bias calibration results at different range settings. The change that is there seems to be more pronounced (and positive) for the z-axis sensors. In Figure 5.13b interestingly this tendency is reversed with the x- and y-axis gyroscope biases apparently more sensitive to changes in the range settings.

The variation in g-dependent error (Figure 5.14) is consistent with the precision of the calibration process given in Table 5.10. Therefore there are no effects that can be directly attributed to the range settings.

For the accelerometer and gyroscope alignment matrices it makes sense to consider the diagonal and off-diagonal elements of the matrices separately. This is because the underlying causes, as well as the specifications, of these errors are different. The off-diagonal entries in $\widehat{\mathbf{A_a}}$ and $\widehat{\mathbf{A_g}}$ are primarily misalignment of the sensitive axes relative to the package outline and misalignment of the package outline inside the calibration box. In other words, this it is a physical alignment problem. Thus one would not expect it to be affected by any register setting (e.g. range) of the sensors. Figures 5.15b and 5.16b show the off-diagonal elements are virtually unaffected by range settings, and thus support this hypothesis.

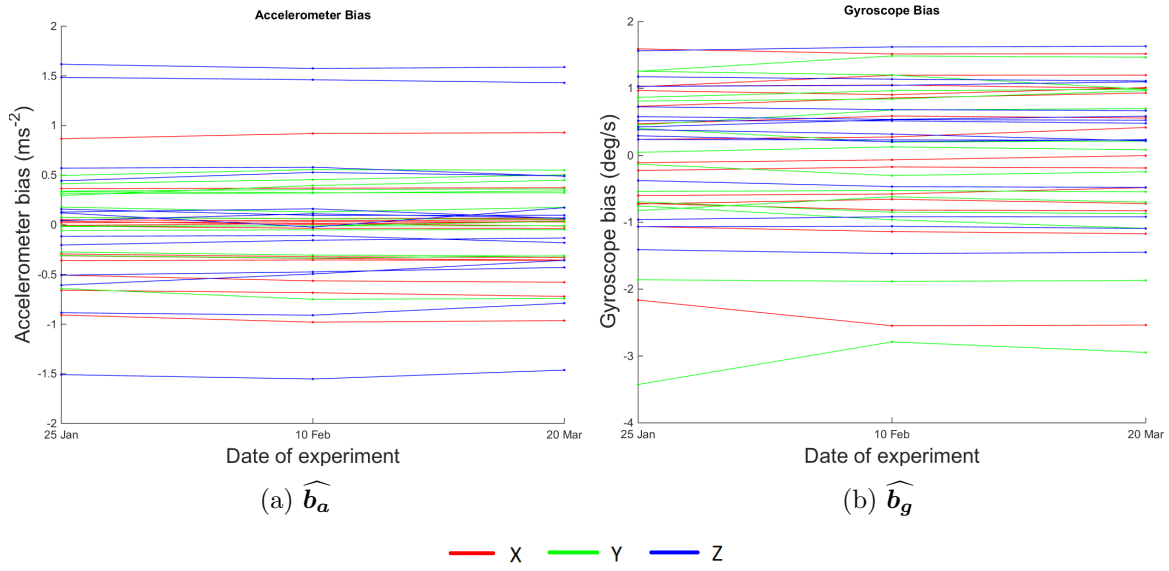
In contrast the diagonal elements of $\widehat{\mathbf{A_a}}$, which is dominated by the accelerometer's scale error, is affected. In Figure 5.15a there is a clear negative trend for the majority of the z-axis accelerometers. The x- and y-axis sensors' sensitivities are not apparently affected by the range settings. This gives more support to the idea that the performance of the z-axis sensor is considerably different, see Section 6.3.3. Note also the much greater spread of calibration results for the z-axis sensor than the x- and y-axis sensors.

The diagonal elements of $\widehat{\mathbf{A_g}}$ are presented in Figure 5.16a. The values for all axes of the gyroscope scale factor are clearly affected by the range setting, and not in the same way. While no particularly obvious trend can be observed it is clear that one can not expect a calibration conducted at one range setting to be particularly useful at another. The fact that there is apparently no dependence on range for the off-diagonal elements (Figure 5.16b) supports this being a genuine phenomenon rather a random occurrence.

The fixed nature of the off-diagonal elements of $\mathbf{A_g}$ and $\mathbf{A_a}$ could be used in two ways. First, a simplified procedure could be designed to estimate only the diagonal elements, for example a single rotation, which might be conducted more frequently. Secondly the expected consistency of the off-diagonal element estimates might be used as a test that the procedure was conducted as expected.

5.5 Temporal validity of calibrated parameters

User calibration procedures were recorded on three separate days several weeks apart with identical sensor settings, including measurement ranges set at 2g and 250 °/s. These occasions were: 25th January 2016, when 6 procedures were recorded as preliminary data collection for developing code; 10th February 2016, when 12 procedures were recorded as presented in Sections 5.2 and 5.4 (on this occasion the other range experiments were also conducted); 20th March 2016 when 4 procedures were recorded before heating the testbed; the 2 post-heating are not included here, these will be presented in more detail in Section 6.6.3. The IMU error estimates calculated from the manoeuvre set C and D of these calibration procedures were averaged for each day and are presented in Figures 5.17 to 5.20.



(a) $\widehat{\boldsymbol{b_a}}$ (b) $\widehat{\boldsymbol{b_g}}$

Figure 5.17: The average bias estimates on three occasions.

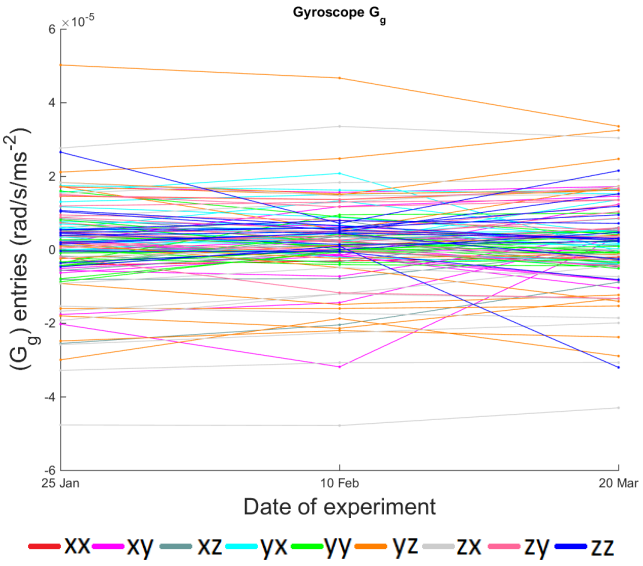
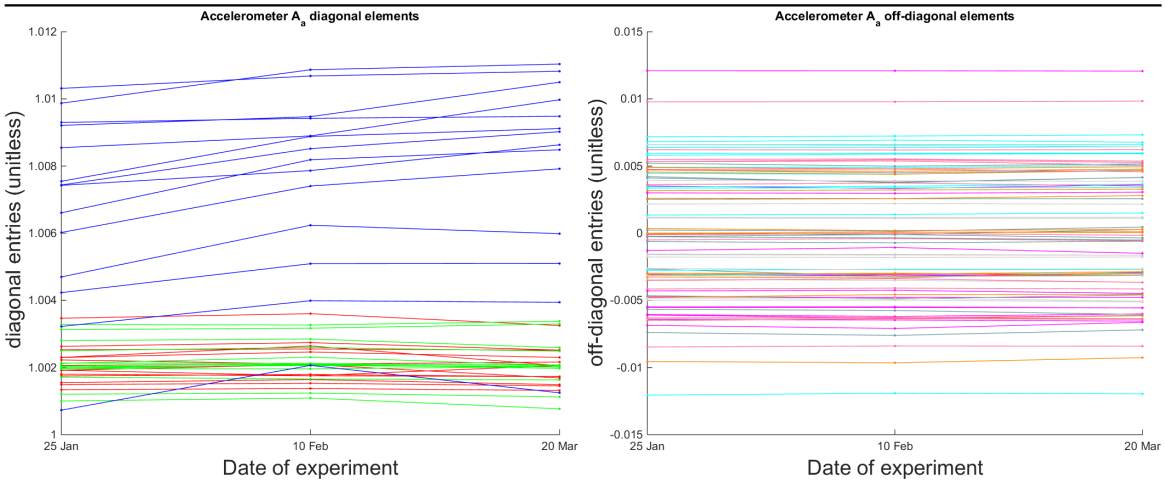


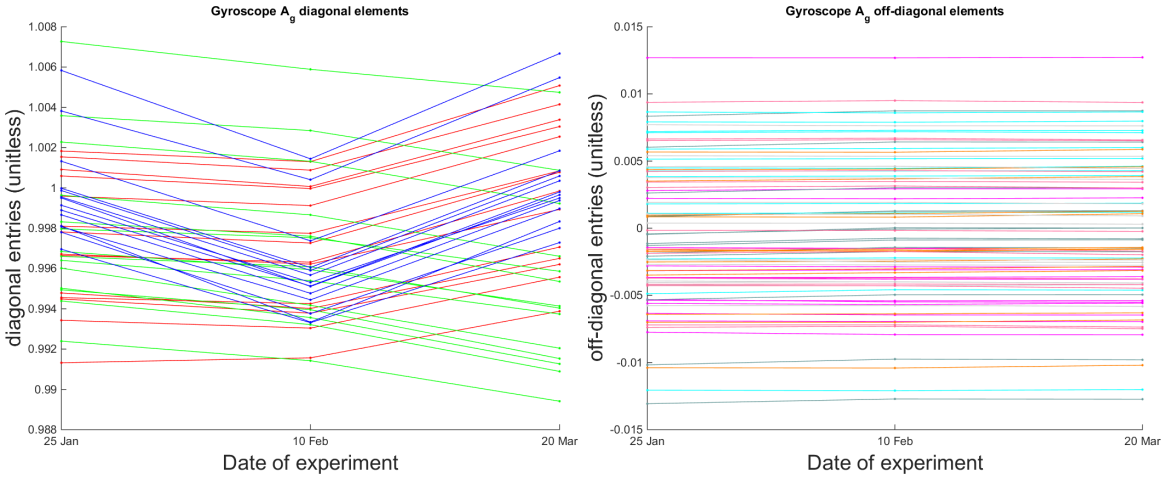
Figure 5.18: The average $\widehat{\mathbf{G_g}}$ on three occasions.

Using this method of comparison, any systematic bias (e.g. from cube deviation-from-orthogonal) would be constant across the multiple procedures and days. Therefore vari-



(a) Diagonal elements (b) Off-diagonal elements

Figure 5.19: The average $\widehat{\mathbf{A_a}}$ on three occasions.



(a) Diagonal elements (b) Off-diagonal elements

Figure 5.20: The average $\widehat{\mathbf{A_g}}$ on three occasions.

ation in the calculated values is due to either random errors of the procedure and sensors, which were shown to be small in Section 5.2 and are further reduced by the averaging across the 4–12 procedures on each occasion, or actual change in the underlying quantity being estimated.

The bias estimates in Figure 5.17 vary slightly day-to-day, but significantly less than the total variation within the population. Part of this variation may be related to differences in ambient temperature on the days in question, bias variation with temperature will be examined in Section 6.6.2. Figure 5.18 shows some correlation in estimates of $\widehat{\mathbf{G_g}}$ between the three days, but there is also considerable variation, given the precision of the calibration process given in Table 5.10 this is not surprising.

The alignment and scale factor matrices show more interesting behaviour. The off-diagonal elements show very little variation day-to-day (both Figures 5.19b and 5.20b). This implies

their calibration, remains valid even after several weeks. The diagonal elements show more variation, some of which can be accounted for by the greater SD of repeatability (particularly $\widehat{\mathbf{A_g}}$, compare Figure 5.7a with Figure 5.20a). However, this repeatability SD, particularly for $\widehat{\mathbf{A_a}}$ is not sufficient to account for all the variation. This implies there is a variation in the true values of the diagonal elements of $\mathbf{A_a}$ and $\mathbf{A_g}$ between the occasions.

The variation in the IMU error estimates calculated on the three occasions in this test are shown to be greatest for the scale factor errors (both accelerometer and gyroscope) and the gyroscope bias. These error terms, as shown in Chapter 3, are among the most significant for KF stability. However, the variation from day-to-day is shown to be considerably smaller than within the population, so an old calibration may have a small enough error from the current value to maintain stability (scale factors of 1% for accelerometer and 0.3% for gyroscope), this is easily the case for the accelerometer as the variation seen in Figure 5.19a is well under 1%. Further experiments are required to calculate detailed statistics on the run-to-run variation, as three occasions is only sufficient for an approximate insight.

5.6 Empirical tests of calibration accuracy

In order to experimentally evaluate the accuracy of the calibration algorithm, as opposed to its consistency, additional measurements were recorded during the experiment presented in Section 5.2. This test comprised starting in a fixed position and orientation (ZU1 in fact, see Section 4.3), then walking around the room, holding the array at waist height for around 30 seconds, and finally return to the same position and orientation on the optical table. Navigation performance is also assessed while static.

In this section, the results of integrating these recorded static and dynamic periods using the inertial navigation equations (see Section 2.2.3) are presented. It should be noted that these navigation tests are not a perfect means to assess the performance of the calibration algorithm. There are a number of limitations. For instance,any given error at a particular moment in time will be the result of a number of different error sources, for example both an attitude error and an accelerometer bias error will cause an increasing velocity error, so this kind of test can only be viewed as a measure of overall performance and the precise cause of poor performance cannot be determined. Also, even a perfectly calibrated INS would drift from the truth due to sensor noise and accumulation of numerical rounding errors so zero-displacement is virtually impossible.

Most of the graphs presented in this section contain a series of sub-graphs each showing one component of the resulting position, velocity and attitude solution (PVA). In these graphs each sensor (or sub-array) is represented by a different colour, e.g. the grey line in all of the sub-graphs represents the calculated PVA of the 13^{th} sensor.

In these experiments the sensors had been running for nearly 60 minutes and so had long since warmed up to long-term operating temperature (see Figure 5.12), which takes less than 10 minutes, and the calibration results used to correct the output was based on a calibration procedure conducted within five minutes of the data recording (i.e. immediately before or after).

This section first presents the results of integrating static data (Section 5.6.1) then a dynamic test (Section 5.6.2)

5.6.1 Static test

This section presents a static test, several were conducted with very similar results so only one is shown for reasons of space.

In a static test such as this the truth is known (no motion) so it is straightforward to assess what is an error. However, the drift observed in the figures is from a combination of sources. In this static case the attitude errors are a combination of (residual) gyroscope bias and gyroscope noise. The position and velocity errors are more complex, they are a combination of (residual) accelerometer bias and noise but also current attitude error, via the compensation for specific force due to gravity, and this is also affected by the accuracy of the attitude initialisation, which is calculated from the table angles in this case. Additionally the accelerometer scale factor of the (in this case) z-axis also effects the position and velocity errors. This means that determining the precisely which error source is the reason for poor performance is not clear, but the relative effectiveness of different calibration techniques can be seen.

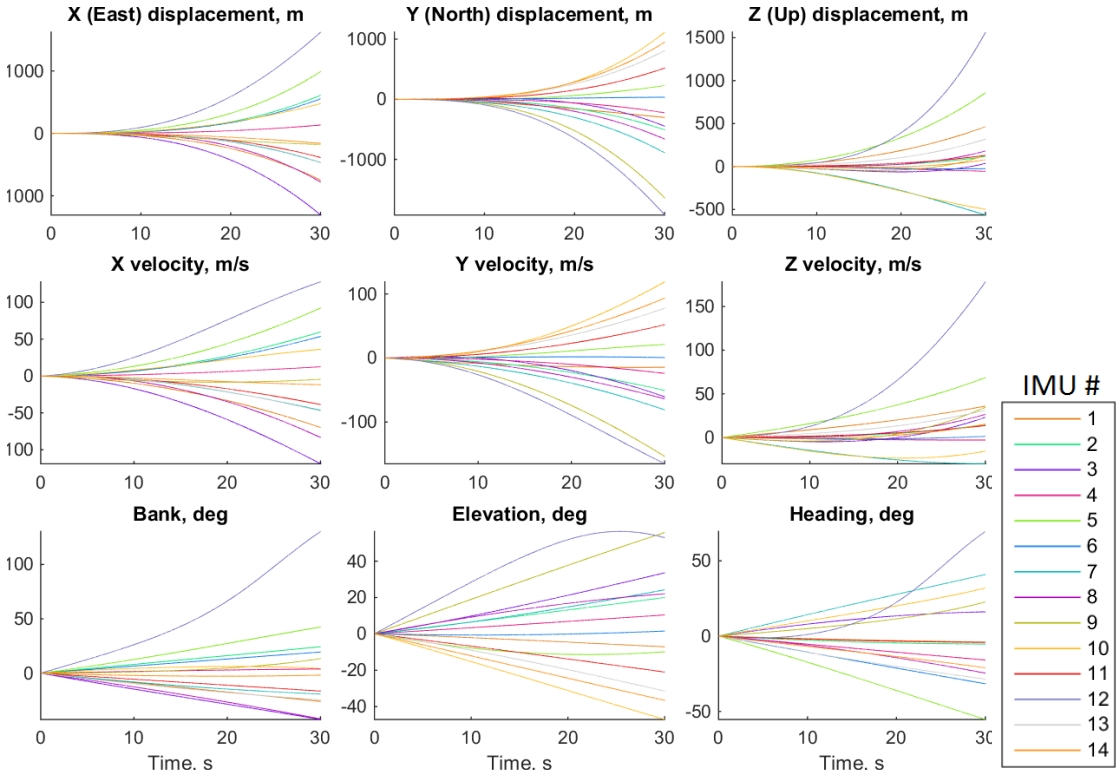


Figure 5.21: Uncorrected IMU measurements integrated over a 30 second static period.

Figure 5.21 shows the inertial navigation PVA solution from the uncorrected IMU measurements. It is clear that the sensors are not going to be particularly useful for inertial navigation in their uncalibrated state. After 30 seconds the attitude solutions are wrong by up to 45° and the positions are wrong more than a kilometre.

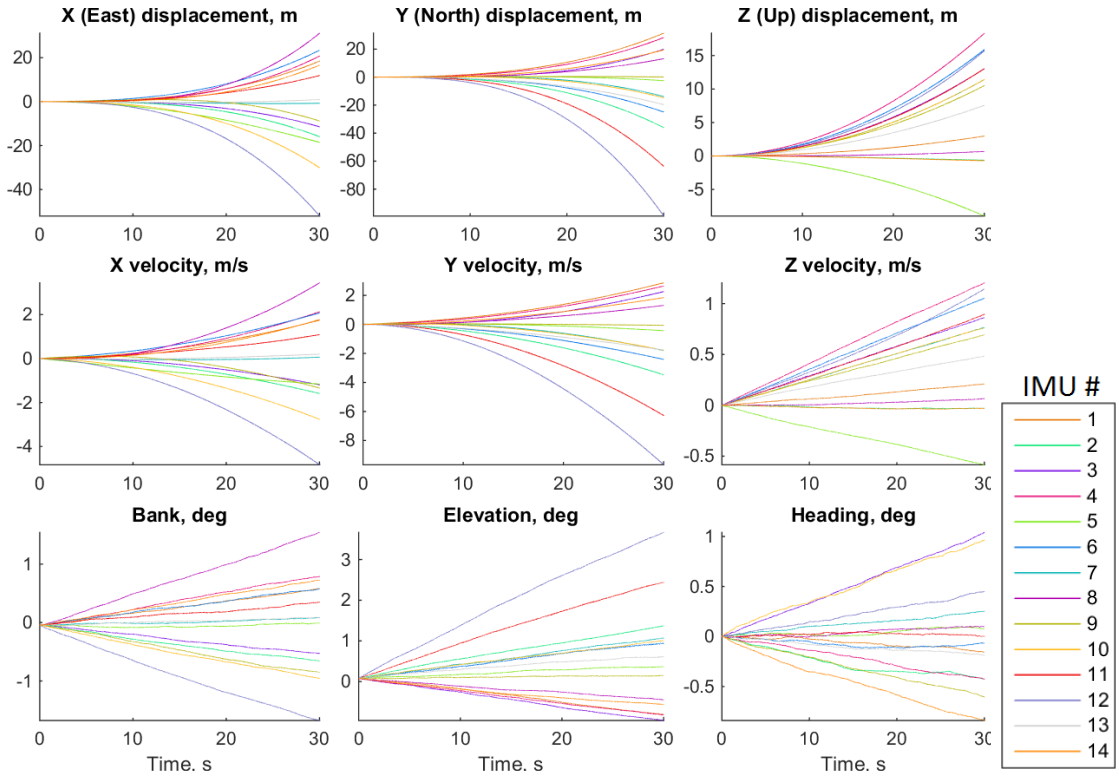


Figure 5.22: Error-compensated IMU measurements integrated over a 30 second static period, using manoeuvre sets C and D.

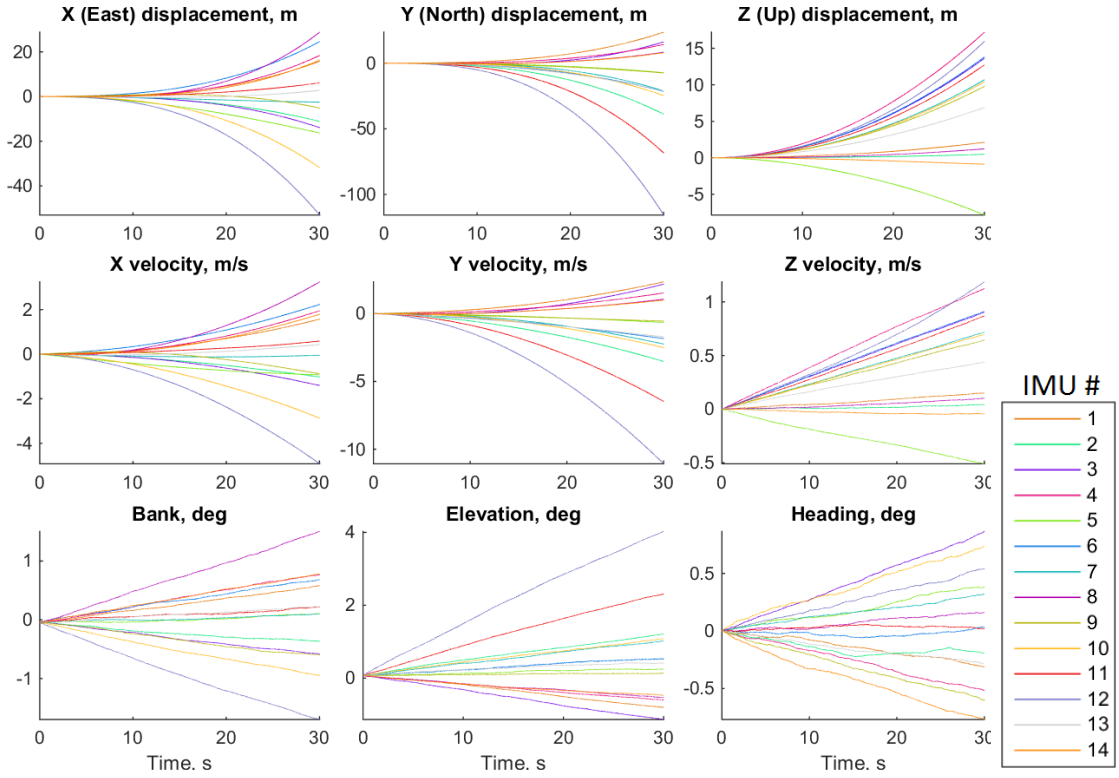
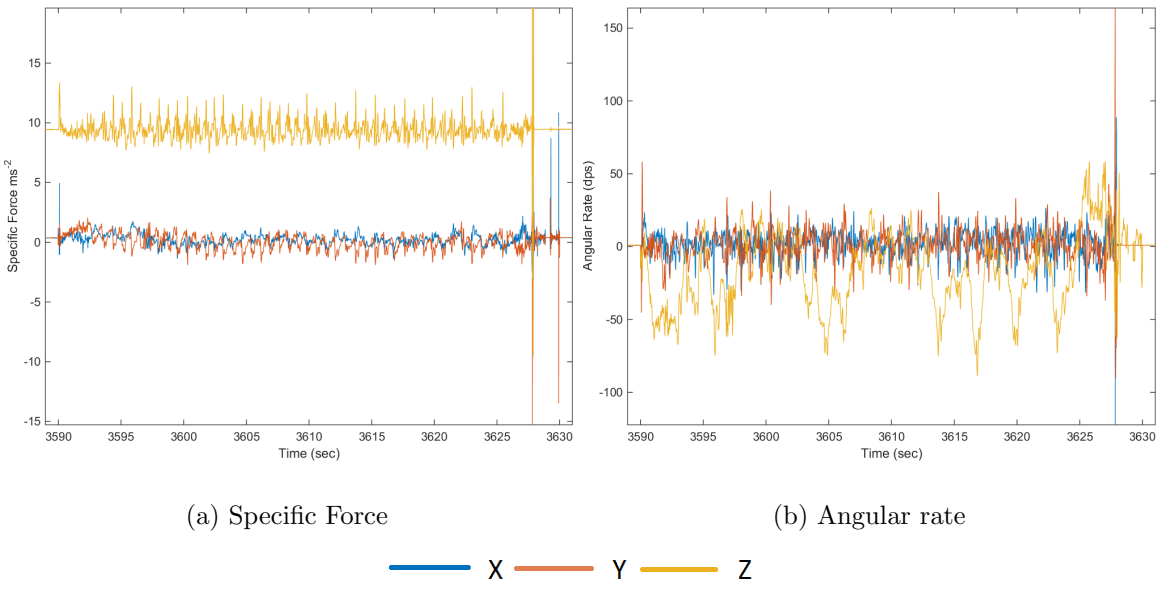


Figure 5.23: Error-compensated IMU measurements integrated over a 30 second static period, using manoeuvre sets A and B.

The PVA solutions calculated from IMU measurements corrected using the advanced (set C and D) and basic (set A and B) calibration techniques are shown in Figures 5.22 and 5.23, respectively. These show much better navigation performance over the 30 seconds observed than Figure 5.21. Position is mostly within a few tens of metres and attitude within a couple of degrees of the static truth. This kind of improvement might be expected, because the uncalibrated gyroscope biases are around 1-2 °/s and the simulation presented in Section 5.3, predicted $\widehat{\boldsymbol{b_g}}$ accuracy of 0.0015 °/s. What is a more interesting observation is that there is very little difference between the IMU measurements corrected by the more laborious 24-position 6-rotation method in Figure 5.22 and the IMU measurements corrected by the 6-position 3-rotation results in Figure 5.23. However, this actually supports the results of the simulation in Table 5.10. This is because the improvement in residual error between for the biases, $\widehat{\mathbf{A}_\mathbf{a}}$ and $\widehat{\mathbf{G}_\mathbf{g}}$ is expected to be modest between the set A and set C estimates, an improvement of less than a factor of two. This is because the optical table is almost level, so the primary benefit of set C over set A is lost. The more significant differences expected (from Table 5.10) in the accuracy of $\widehat{\mathbf{A}_\mathbf{g}}$, between manoeuvre sets B and D, are not excited because this is a static test.

5.6.2 Dynamic test

To provide a more exhaustive test of the IMU navigation performance, inertial data was recorded for a 40 second period of pedestrian motion walking around a laboratory. This, with a single second of static data before and afterwards, was integrated with the inertial navigation equations (see 2.2.3) to assess the performance of the sensor calibration. The data recorded by one of the sensors (SDA02, number 13 in the graphs) for this period is shown in Figure 5.24. A second test was also conducted with very similar results so only one is shown for reasons of space.



(a) Specific Force (b) Angular rate

Figure 5.24: The raw IMU measurements recorded for the pedestrian motion from one of the sensors (number 13).

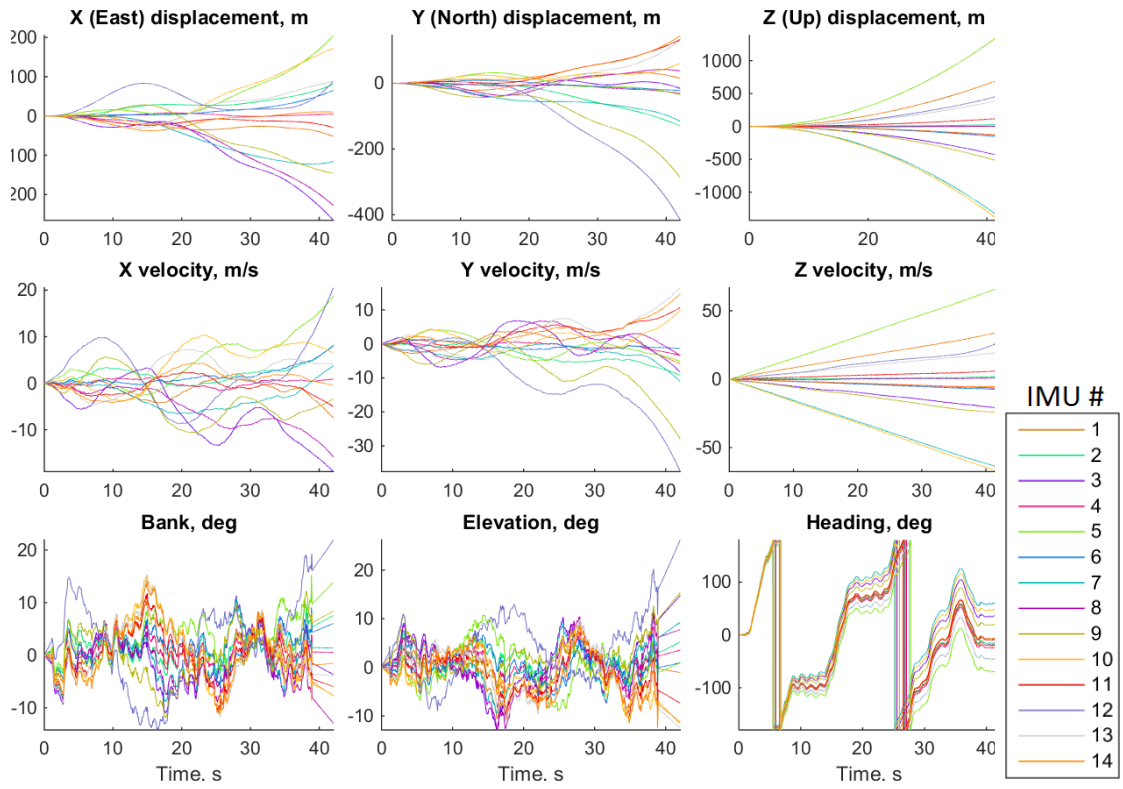


Figure 5.25: Motion profile obtained from a short pedestrian motion test by running inertial navigation with uncorrected IMU measurements.

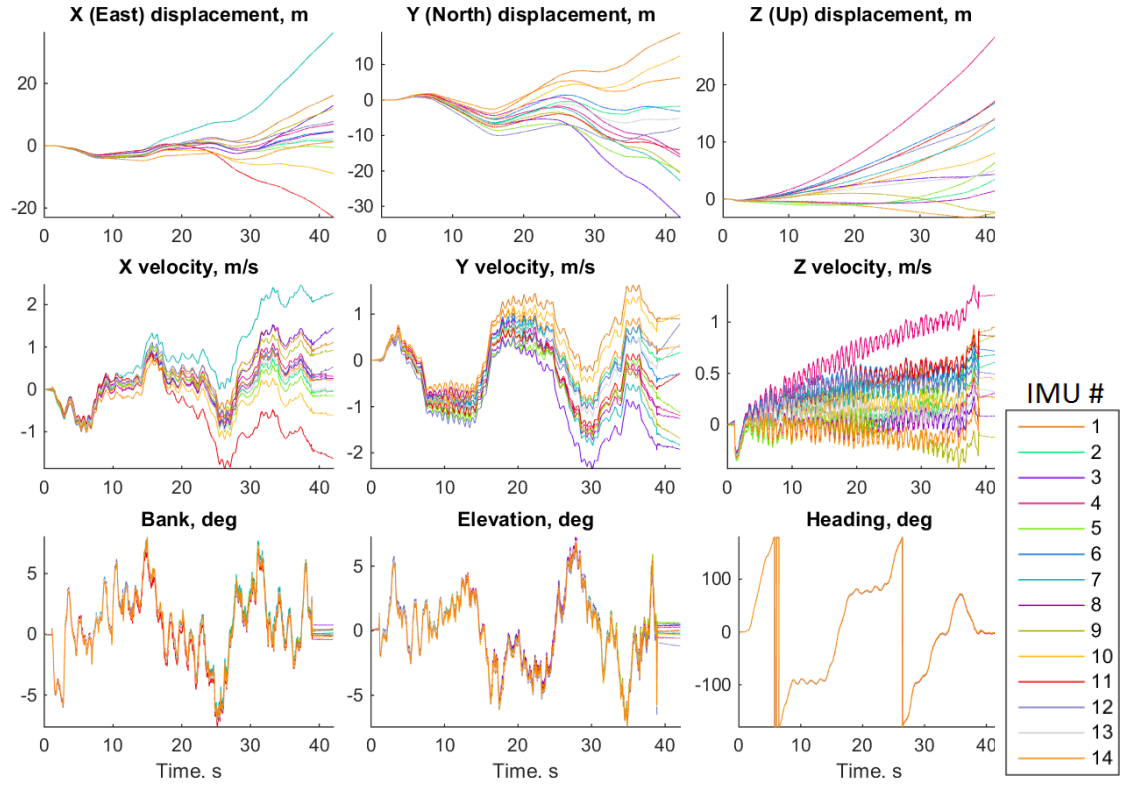


Figure 5.26: Motion profile obtained from a short pedestrian motion test by running inertial navigation with IMU measurements corrected using manoeuvre sets C and D.

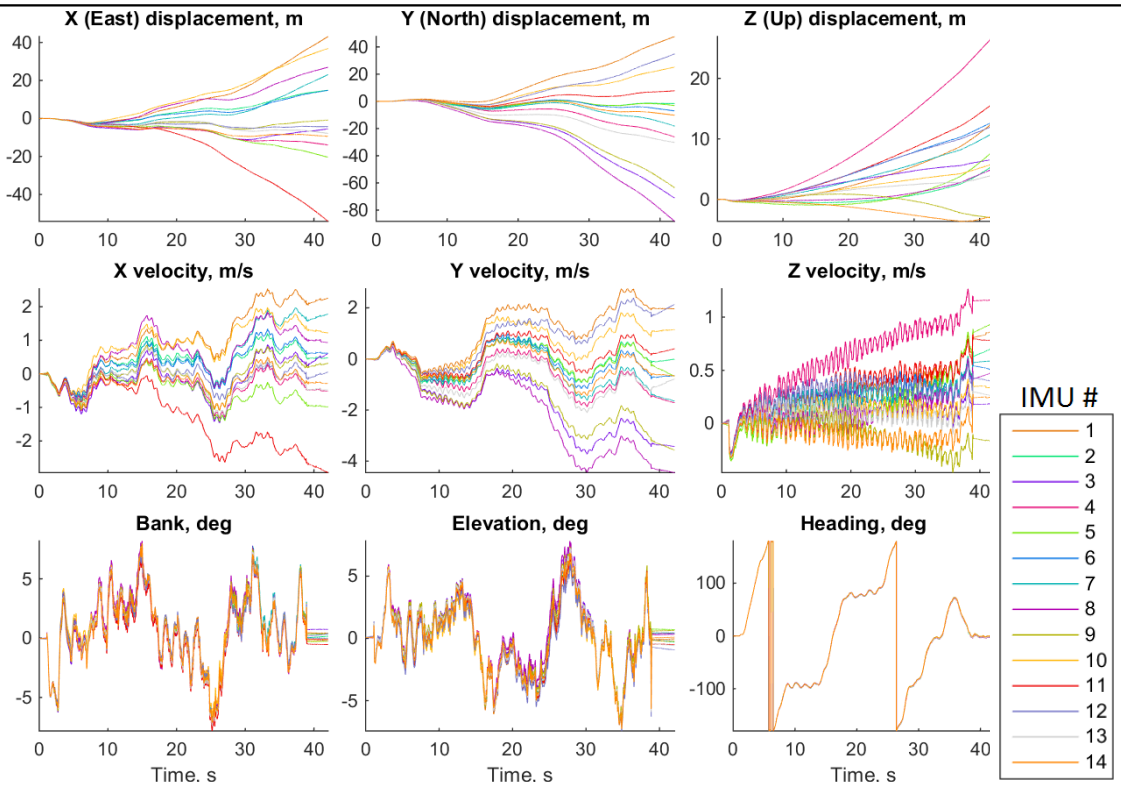


Figure 5.27: Motion profile obtained from a short pedestrian motion test by running inertial navigation with IMU measurements corrected using manoeuvre sets A and B.

While there is no truth model available for these dynamic tests, there are a number of ways of assessing performance. At all times, the equipment remained within 5m of the starting point, and during the first and last second the array was static and at the same position and attitude. Additionally all the sensors are rigidly attached together so their motion should have been identical. Thus, any divergence in their PVA solutions indicates an error.

In a dynamic test such as this, errors in the PVA solution are from a combination of sources, and in addition to the errors mentioned at the start of Section 5.6.1, the attitude solution is also affected by the accuracy of the gyroscope scale factor and alignment/cross-coupling matrix estimate ($\widehat{\mathbf{A}_\mathbf{g}}$) as well as the accuracy of $\widehat{\mathbf{G}_\mathbf{g}}$ and the magnitude of errors which are not estimated by the calibration procedure such as scale-factor non-linearity. The position and velocity solutions are still affected by all the error sources which degrade the attitude solution, including errors in $\widehat{\mathbf{A}_\mathbf{g}}$. However, the variation in specific force, see Figure 5.24a, means that, in addition to all of these errors sources, the accuracy of $\widehat{\mathbf{A}_\mathbf{a}}$ and the magnitude of the unestimated higher order errors such as scale-factor non-linearity also affect the navigation performance.

Figure 5.25 shows the navigation performance obtained from the uncalibrated set of sensors. The PVA solution shown is poor with up to a kilometre of error vertically, several hundred metres horizontally and tens of degrees of attitude error with the heading performance being the worst. The east and north velocity spread out so quickly and in such an apparently unstructured manner that it is very difficult to determine what the true profile might have been from examining the spread of results. The same thing could be said of the bank and elevation profiles.

It is important to note that walking in circles is flattering to the position accuracy because the body-frame is rotated relative to the local navigation frame through the experiment, so the effect of body-fixed errors, such as biases, are being applied in different local navigation frame directions and so their effects partly cancel out. This explains why the performance appears worse for the uncalibrated static experiment (Figure 5.21) than the uncalibrated dynamic experiment (Figure 5.25).

Figures 5.26 and 5.27 show the PVA solution from the calibrated output of the IMUs, from the set C and D and the set A and B manoeuvres, respectively. The navigation performance is much better with this calibration than without calibration, the position errors only being a few tens of metres and the final attitude difference (which is known to be very close to zero) being less than a degree. What can be seen by comparing these two figures is that, in contrast to the analogous pair for the static data, there is significantly less spread with the C/D technique. Comparing the bank and elevation between the two, especially around the peaks on the graph, reveals a greater spread among the 14 sensors in Figure 5.27, even very early into the experiment. Interestingly, examining the final (static) second does not show much greater spread between the two datasets. The velocity solution is also slightly worse in Figure 5.27 than Figure 5.26

The static experiments in Figures 5.22 and 5.23 showed that there is little difference in the bias calibration, between the estimates from the set C and D and the set A and B manoeuvres, therefore the difference must be due to other error sources. The variation in specific force (see Figure 5.24a) over the experiment is not very great compared to a static experiment, as the array remains the same way up, so there is unlikely to significantly more impact from $\widehat{\mathbf{G_g}}$ than in the static case. Thus the difference between Figures 5.22 and 5.23 must be due to $\widehat{\mathbf{A_a}}$ or $\widehat{\mathbf{A_g}}$. Due to the two complete (heading) rotations, any impact of errors in $\widehat{\mathbf{A_a}}$ on the x- and y-velocity solution will cancel out over the course of the whole experiment, but not their influence on the position solution. Similarly, any impact from errors in $\widehat{\mathbf{A_g}}$ on the bank and elevation solutions will cancel out over the circuits, but not their influence on the velocity or position solutions. Comparing the bank solutions in the first 15 seconds (i.e. before the first circuit has been completed) between Figures 5.22 and 5.23, the spread of estimates in the set A/B solutions are greater than the C/D solutions. Also the spread of x- and y-velocity solutions in the final few seconds (more clearly visible for y) is slightly worse for the set A/B solution. Both of these points suggest that the difference in performance between A/B and C/D, is due to errors in $\widehat{\mathbf{A_g}}$ rather than $\widehat{\mathbf{A_a}}$. This is not particularly surprising as the C/D solution is predicted to make more difference to the $\widehat{\mathbf{A_g}}$ than the $\widehat{\mathbf{A_a}}$. This supports the simulation results in Table 5.10.

5.7 Summary

This chapter presented three primary results. It determines the consistency of the calibration procedure by conducting the calibration procedure multiple times and examining the distribution of the output, in Section 5.2. Section 5.3 presented a comparison between these empirical results and the accuracy and consistency expected from a simulation of the type presented in Chapter 4 but with the input modified to better reflect the circumstances of the

empirical calibration procedures. The main results of these experiments and simulations are summarised in Table 5.11. The variations in the IMU error estimates were examined at different range settings (Section 5.4) and from experiments with identical settings on different days (Section 5.5). Results were also presented (Section 5.6) showing how much navigation performance is improved by comparing the inertial navigation performance with the simpler (set A and B) and more complex (set C and D) calibration manoeuvres to uncalibrated sensors using static and dynamic datasets.

IMU Error	typical	Sample of 14 IMUs			Expt repeat.	500 run simulation	
Coefficient	spec.	Mean	SD*	RMS	SD*	Acc.(RMS)	Rep.(SD)
$\widehat{\boldsymbol{b_a}}(x)$ (ms^{-2})	±0.59	0.397	0.382	0.5414	0.0064		
$\widehat{\boldsymbol{b_a}}(y)$ (ms^{-2})	±0.59	0.193	0.173	0.2550	0.0080	0.00626	0.000427
$\widehat{\boldsymbol{b_a}}(z)$ (ms^{-2})	±0.78	-0.508	0.593	0.7644	0.0106		
$\widehat{\boldsymbol{b_g}}(x)$ (°/s)		0.0704	1.295	1.250	0.0187		
$\widehat{\boldsymbol{b_g}}(y)$ (°/s)	±5	0.625	0.8615	1.0393	0.00145	0.00145	0.0014
$\widehat{\boldsymbol{b_g}}(z)$ (°/s)		0.5700	0.7006	0.8836	0.0151		
$\widehat{\mathbf{A_a}} - \mathbf{I_3}(x,x)$		0.00205	0.000883	0.00222	0.0001107		
$\widehat{\mathbf{A_a}} - \mathbf{I_3}(y,y)$	±3%	0.00201	0.000468	0.00206	0.0000886	0.00112	7.59e-5
$\widehat{\mathbf{A_a}} - \mathbf{I_3}(z,z)$		0.00784	0.00298	0.00836	0.0002551		
mean $\widehat{\mathbf{A_a}}$ off-diag.	±2%+$\mathbf{C_b^s}$	-0.000154	0.0048	0.0048	0.0001289		
$\widehat{\mathbf{A_g}} - \mathbf{I_3}(x,x)$		-0.00641	0.00164	0.00660	0.00391		
$\widehat{\mathbf{A_g}} - \mathbf{I_3}(y,y)$	±3%	0.00106	0.00279	0.00289	0.00127	0.00164	0.00105
$\widehat{\mathbf{A_g}} - \mathbf{I_3}(z,z)$		-0.00235	0.00249	0.00336	0.00587		
mean $\widehat{\mathbf{A_g}}$ off-diag.	±2%+$\mathbf{C_b^s}$	-0.000321	0.0051	0.0052	0.0000892		
$\widehat{\mathbf{G_g}}$ (°/s/g) mean all 9 entries	none	0.00077	0.0055	0.0057	0.003839	0.00317	0.00252
$\partial \boldsymbol{b_a}/\partial T\,(x)$ (ms^{-2}/°C)		-0.0201	0.0160	0.0260			
$\partial \boldsymbol{b_a}/\partial T\,(y)$ (ms^{-2}/°C)	±0.0147	-0.00727	0.00690	0.00985	n/a	n/a	n/a
$\partial \boldsymbol{b_a}/\partial T\,(z)$ (ms^{-2}/°C)		0.00618	0.0251	0.0249			
$\partial \boldsymbol{b_g}/\partial T\,(x)$ (°/s/°C)	±30 °/s	0.000516	0.0472	0.0455			
$\partial \boldsymbol{b_g}/\partial T\,(y)$ (°/s/°C)	(-40 to 85	-0.0266	0.0148	0.0302	n/a	n/a	n/a
$\partial \boldsymbol{b_g}/\partial T\,(z)$ (°/s/°C)	°C)	-0.000366	0.0150	0.0144			

Table 5.11: Summary statistics of experimental estimates and simulation results for $\widehat{\boldsymbol{b_a}}$, $\widehat{\boldsymbol{b_g}}$, scale factor errors and bias variation with temperature. Set C (technique 1) and D (sum method) results quoted. Specifications are from [23] and information is from Tables 5.7, 5.10, 6.1 and 6.2. 'SD*' denotes unbiased estimate of population SD, i.e. normalised by $n-1$.

The simulated calibration procedure produced results that were a good match for the empirical results for the $\widehat{\mathbf{A_a}}$ and $\widehat{\mathbf{G_g}}$, and to slightly underestimate the variation found in the accelerometer and gyroscope biases, which could be explained by the biases changing slightly over time and/or with temperature during the physical experiment. In the case of $\widehat{\mathbf{A_g}}$ the situation was more nuanced. For the six-rotation manoeuvre set D, the simulated results correspond well with the experiments, and in both cases it is shown to be effective. However, the simulation failed to take into account all the complexity introduced by the free-hand rotations of manoeuvre set B, so the difference in performance between the sum-of-angular-increments and integrated algorithms was not adequately captured. However, the basic recommendation from the simulation is to avoid the set B manoeuvres, particularly the sum-of-angular-increments algorithm, and the fact that it is even worse than simulated

is good grounds to avoid it.

This chapter also presents results characterising the sensors error performance at different range settings. The range setting was shown not to make a significant difference to the systematic error distribution (Section 5.4) or the noise performance (Table 5.6).

Section 5.5 presents results on how useful a user-conducted calibration procedure is in the longer term, that is whether an old calibration is still an improvement on no calibration weeks later. This was based on the average estimates of calibration procedures conducted on three separate days. The IMU biases were found to vary slightly from day-to-day, as were the scale factors Any effect on $\widehat{\mathbf{G_g}}$ over time was not large enough to be seen given the proportionately high uncertainty in the estimate.

The estimates of off diagonal elements of $\mathbf{A_g}$ and $\mathbf{A_a}$ were shown to remain valid at all the range settings (Figures 5.15b and 5.16b), and despite being several weeks old (Figures 5.19b and 5.20b), they will also be shown to be remain valid when temperature changes (Figures 5.19b and 5.20b). This implies that a one-time calibration would be sufficient for the off-diagonal elements of the $\mathbf{A_a}$ and $\mathbf{A_g}$, meaning that it might be possible to only estimate scale factors as KF states.

It is also worth noting that the run-to-run variation in $\widehat{\boldsymbol{b_a}}$ and $\widehat{\boldsymbol{b_g}}$ is sufficiently small to be well within the stable region for KF integration, as found in Chapter 3. The simulated RMS accuracy and the experimental repeatability of $\widehat{\mathbf{A_a}}$ are well below the medium level (1% scale factor, 0.5% cross coupling, see Table 3.3) required for stable KF integration, even including the run-to-run variation. In fact, they are nearly within this level uncalibrated. The required accuracy of $\widehat{\mathbf{G_g}}$ is reached even without calibration for these sensors. The accuracy needed for $\widehat{\mathbf{A_g}}$ is the low level (0.3% scale factor, 0.2% cross coupling, see Table 3.3), the simulation predicts that the estimate should be within that accuracy for the manoeuvre set D (the set B manoeuvres are not accurate enough), the experimental repeatability within this requirement for the off-diagonals and is right on this level (0.13–0.6% see Table 5.7) for the diagonal elements. However, the variation from day-to-day of the gyroscope scale factors is above this level so the calibration would need to be repeated each time the IMUs are used.

In this chapter and Chapter 4 the performance improvement possible with user-conducted calibration has been shown. This can make a dramatic improvement in use relative to uncalibrated sensors, see Section 5.6. However, there are limitations to this approach, for example the noise cannot be reduced by calibration, and so will become the dominant error source when the systematic errors are well calibrated, and the calibrated sensor biases will frequently vary with temperature, and given equipment limitations of user calibration techniques this cannot not necessarily be calibrated. In Chapter 6 some techniques which use arrays of sensors, including techniques which can reduce the effects of both of these error sources, will be presented.

Chapter 6

Use of sensor arrays

This chapter presents techniques that can be used to improve the performance of an inertial navigation system (INS) by using an array of IMUs.

This will begin with a discussion of the basic effects of all array techniques and the performance improvement expected by adding more sensors (Section 6.1). Following this groundwork, theoretical methods for improving performance more than a simple array-average approach are presented. This begins with discussion of techniques using arrays of similar sensors, for example sensors of the same model (Section 6.2). This includes orienting the sensors so that their sensitive axes point in opposite directions to reduce temperature dependent bias drift. This is followed by discussion of arrays of sensors with different performance characteristics , particularly when the This includes where the performance and performance requirements of sensors within the array are asymmetric and techniques for weighting the output of sensors with different performance characteristics (Section 6.3). This chapter also examines the idea of arrays of IMUs with different measurement ranges (Section 6.4) This chapter also discusses the best way to calibrate an array of IMUs using user calibration procedures (Section 6.5).

Experiments which provide supporting evidence for the array techniques investigated in this chapter are presented in Section 6.6. This includes examining the distribution of errors of the 14 sensors in the test hardware described in Appendix A, and conducting dynamic experiments. These experiments include the effect of changing the ambient temperature.

This sections main findings are summarised in Section 6.7.

6.1 General error characteristics of sensor arrays

The type of array considered here is a set of multiple inertial sensor triads, fixed relative to one another within the body frame of a navigation device. This is assumed in subsequent discussions. An example would be multiple MEMS IMUs on the same printed circuit board (PCB). As there are multiple triads each of which is measuring the inertial forces in/about all three principle axes, all the inertial forces are measured by more than one IMU.

The simplest thing that can be done with these multiple measurements of specific force and/or angular rate is to transform them all into an array resolving axes, and then average their readings into a single virtual IMU (VIMU) output for the navigation system. The transformation is fairly straightforward (see Section 4.3) as in the case of a single PCB

sensor positions and orientations should be known to < 0.5mm and $< 5°$, respectively (according to [186] placement accuracy of between 10–50 μm SD for robotically constructed mass production PCBs), as otherwise their mis-placement would cause short-circuits. This mis-placement should be effectively fixed after construction, see Section 5.5, so a one-time calibration should be sufficient to determine these offsets. These offsets are required to compensate for the lever-arm effect. This effect is important wherever the sensors are not located at the same point, i.e. always. The IMU or IMU array is meant to be describing the motion of a particular point fixed in the body-frame, but many of the sensors are not all located precisely on at this point. If the body frame rotates about this point in inertial space then any sensor not located at the will measure a specific force due to this rotation proportional to its distance from that point. This can be compensated for if the position of each accelerometer in the body frame is known.

This combined VIMU output will have smaller errors than those of the individual sensors, or rather the errors will tend towards the array-average. In the case of the sensor white noise, since by definition that error is zero-mean, the averaged signal of an array of n sensors will be $\sqrt{n}$ less noisy. This behaviour is noted frequently in the literature, e.g. [164, 187], but what might not be so obvious is that systematic errors will also be reduced by this averaging. Take any error distributed over the population with a mean value a and SD of b, the overall averaged n-sensor signal will be distributed with mean value a and SD of $b/\sqrt{n}$. This assumes that the IMU error coefficients of all sensors in the same array are independently chosen from this distribution, which may not be the case for those describing array alignment. Strictly the distribution of the array mean tends towards being distributed normally, even if the underlying sensor distributions are not normal, by the law of large numbers, but arrays of sensors will likely contain tens rather than hundreds of sensor triads so this will not have a significant effect.

Assuming that the x- y- and z-sensors of each IMU behave similarly and that each of the sensors' individual errors are drawn from zero-mean distribution (i.e. $a = 0$), the actual physical orientation of the sensors in the body frame does not make a difference. There is no difference if the sensors are arranged so that all the sensor axes are parallel or if they are at an arbitrary (but known) orientation to one another.

This virtual IMU could also incorporate some integrity checks which eliminate gross errors from the virtual IMU output, there is significant research into this topic, see Section 2.5.2, but it is not considered in this thesis.

The main benefit of the VIMU approach, over treating the sensors individually, is to separate the combination of the sensors from the integration algorithm to reduce computational costs. If all the sensors were treated separately there would be $6n$ bias states for the n IMUs rather than just 6 for the VIMU. Additional computational resources may be better deployed using a better integration algorithm (e.g. UKF [130]) rather than adding many states to a standard KF.

Note that, in order for techniques involving averages of the sensors, such as this one, to make any difference values of the sensors' errors must be different across the set being averaged. This is trivial for most error sources, but some calibration procedure errors may effect all

the sensors in an array in a virtually identical way. Ideally the mean of the error source would be zero, which would make the effect of error tend to zero as the The errors need not be distributed exactly as specified, or rather as implied by the specification, but there must be a random distribution from which the coefficients of each within the array are drawn. This should generally be a reasonable assumption to make for uncalibrated errors. However, if the sensors have been subject to a calibration procedure, this may not apply because some of the residual error coefficients are primarily functions of the imperfection in the procedure. For example, in the simulations of Chapter 4, many of the accelerometer errors were shown to be more strongly related to the box's deviation-from-orthogonal-sided than any other factors. This is partly because the default procedure was not simulating a highly accurately made box. The implications of this are that array averaging will not make as much difference residual sensor errors that are primarily a function of the cube's accuracy such as $\widehat{\mathbf{A}_\mathbf{a}}$, as these errors are common to all sensors in the array. However, it might still have an effect on those error sources that are not significantly affected by cube accuracy such as $\widehat{\boldsymbol{b}_\boldsymbol{g}}$, the accuracy of which mostly related to sensor noise (see Chapter 4).

However, in the event that a certain (even-order) error characteristic is not zero-mean, the sensor orientations can be specifically chosen to make the array averaged error zero-mean, as will be explained in the next section.

6.2 Anti-parallel arrays of similar sensors

If one constructs an array consisting of multiple units of a single type of IMU then certainly in a mass-production device, these are likely to be IMU from the same batch. One reason for this is that component placing machines are generally supplied with small surface-mount (SMD) components on a reel, so sensors produced together will be packaged together and stay together until circuit assembly.

One might expect the output of sensors from a single batch to be more correlated than the entire population of sensors, because they may all be constructed from the same silicon wafers at the same time. This correlation can be exploited to improve the array's performance.

This section focusses on and extends an idea proposed by Yuksel *et al.* [55]. This idea is to create an array of gyroscopes consisting of six sensors, where two are arranged for each orthogonal axis with their sensitive axes pointing *in opposite directions.* A 2D illustration of this concept is shown in Figure 6.1. The angular rate reading for each axis is then half the difference of the two sensors. Yuksel *et al.* observe earlier in the paper that two sensors of the same model (ADXRS150 [188]) had very similar temperature response curves. They then suggest if the temperature and g-dependent response coefficients were the same for both sensors then this formulation of "half the difference" would eliminate the temperature dependent response and one of the three components of the g-dependent bias (for each axis). They note that "with proper individual sensor placements, common bias like factors can be eliminated in multi-IMU configurations. (It should be noted that. more than 2 sensors must be used for each axis in order to eliminate all of the g-dependence effects.)" [55]. They also

demonstrated that the idea with a static test where the temperature varies with time. They plot the individual outputs of two sensors and their combination to illustrate the reduction in bias variation with temperature ($\partial \boldsymbol{b_g}/\partial T$). They do not prove this was viable with more than one pair of sensors. In [8], my co-authors and I demonstrated a similar effect for a pair of L3G4200D 3-axis gyroscopes [24], although not for all axes. It should be noted that even without an (apparent) change in ambient temperature, as in Section 6.6, the temperatures of IMUs within an array can still be highly correlated (e.g. see Figure 5.12)

In fact this opposing sensitive axis idea is more powerful than Yuksel *et al.* presented. First, for this idea the coefficients of the error (e.g. $\boldsymbol{b_g}$ or $\partial \boldsymbol{b_g}/\partial T$) need not be equal; they need only have a non-zero mean. Importantly, if the array is arranged anti-parallel this technique means that array-average even-power errors, when distributed with mean a and SD b will converge to $b/\sqrt{n}$ rather than $a + b/\sqrt{n}$. This means that if the error coefficients are zero-mean (i.e. $a = 0$) it behaves in the same way as a parallel array, but it helps when $a \neq 0$. This means that there is no downside to using this technique compared to aligned arrays.

It should be observed that this anti-parallel array technique works for errors which do not depend on the sensors' output (0^{th}-order) such as bias, but it will not work for those which depend on an odd power of the output, such as scale-factor or alignment (1^{st}-order). It would also theoretically work for errors depending on an even power of sensor output, henceforth even-power errors, such as quadratic sensor non-linearity (2^{nd}-order), which for some MEMS models is the most significant part of the non-linearity, as discussed in Section 2.2.6. Thus it works for temperature dependent bias drift, but not temperature dependent scale-factor drift. These two temperature-related errors are likely to be the more significant than other temperature-related errors on the basis that they are the two most commonly specified on datasheets, such as [23, 25]. Note that anti-parallel arrays will also reduce the non-temperature related part of the bias, if that is asymmetrically distributed.

Yuksel *et al.* [55] were working with single-axis-per-chip inertial sensors which meant that they could create a fully anti-parallel array with only 6 sensors. However, any realistic low-cost array implementation in future will be based on 3-axis sensors (one SMD component which contains 3 orthogonal gyroscope and/or 3 orthogonal accelerometers) so a single pair cannot be fully opposing. For example in Figure 6.1, the z-axes would be in the same direction, out of the paper.

In order to create a fully anti-parallel set one can either assume that the x- and y-axes are sufficiently similar to have +x of one IMU chip oppose -y of the other or use four sensors arranged like those in Figure 6.1 on both sides of a PCB. This 4-IMU approach has the advantage of eliminating all the components of $\mathbf{G_g}$ from effecting the result. One might expect the x- and y-sensors of an IMU to be the same design (see Section 6.3.3) and thus have similar error distributions. However, as will be seen in Section 6.6, our research does not support the idea that the x- and y-axis sensors always have the same error coefficient distributions.

This would equally apply to bias drift with temperature which is just as, if not more, likely to have a highly non-zero distribution such as $0.015 \pm 0.005\,°/\text{s}/°\text{C}$. This is particularly significant because bias drift with temperature is difficult for an end-user to calibrate. This

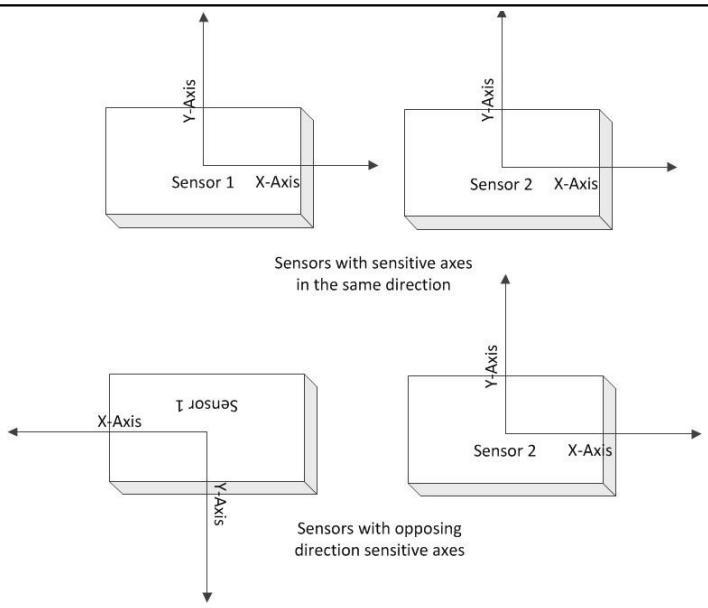


Figure 6.1: An illustration of sensors with aligned and opposing x and y sensitive axes.

is because any user calibration procedure would have to be repeated at a wide range of different operating temperatures for an accurate calibration.

The critical question for this idea is: how likely are non-zero-mean IMU error coefficient distributions?

In the simulations presented in Chapters 3 and 4, all the IMU errors were *assumed* to be zero-mean and Gaussian. This assumption is a natural one given the specifications on the datasheets which are symmetric about zero, e.g. $\pm 2\%$. However, as discussed in Section 2.2.7, this assumption is not necessarily true.

The prominence given to the specified performance in flyers and leaflets distributed by inertial sensor manufacturers, implies that it is valuable for marketing. This combined with the difficulty (at least for a small-scale user) of verifying the manufacturers' claimed performance, might be justifiable reason for scepticism among readers. Additionally any non-zero mean error if presented as such on the datasheet would look like the manufacturers were not doing as good a job as they might on the out-of-the-box calibration. Essentially they would be flagging up that they had made a mistake.

These reasons why the manufacturers would imply that their sensors have zero-mean errors do not mean that the errors are *actually* zero-mean. The fact that manufacturers do not always accurately state sensor characteristics that are easy to check should raise further scepticism, particularly with regard to characteristics that are hard to check such as non-linearity or the sensor's response to temperature. For example in [8], my co-authors and I presented an example of a 12-bit sensor where the LSB of the ADC was always a zero. However, few published studies into the accuracy of these datasheets exist.

There are a few published examples independent from sensor manufacturers which have shown non-zero mean distributions of errors, although most are from small samples of sensors. An example is in [155], where incidentally to the main point of the paper, one can

see their sample of 14 accelerometers is certainly not zero-mean for all the errors they have calibrated (e.g. x-axis bias). It is difficult to draw conclusions from reports which publish the calibration parameters of only 2 or 3 examples of a sensor model, such as [55], so these are disregarded.

There are also studies from sensor manufacturers. Although these are partly marketing, it is still worth considering their results. An example of this type of study is [48], where the manufacturer ST presents data on the performance of their gyroscope model L3G4200D. While some of the error coefficients are close to zero-mean others are clearly not. For example the bias (over 1000 samples) appears, from a histogram, to have a significant negative mean on two of the three axes. This implies that the anti-parallel array would help the bias performance of uncalibrated sensors. In this report, the error type with the strongest non-zero-mean distribution is the "Sensitivity change with temperature", this is specified as $\pm 2\%$ "typical" over $-40°$C to $+85°$C, which would imply a zero-mean distribution. However, the distribution reported over 22 sensors in [48], has mean $-0.0270/-0.0301/-0.0248$ with SD of 0.0091/0.0042/0.0038 in units of %/°C for pitch/roll/yaw sensors respectively. This translates to -3.375% (SD 1.1375%), i.e. $-3 \pm 1\%$, over the whole temperature range. This error is not affected by the anti-parallel array idea. However, data on the bias change with temperature, collected from 33 sensors, implies that the anti-parallel technique suggested here would reduce bias drift with temperature on the yaw-gyroscope but not on the pitch and roll ones.

6.3 Arrays of sensors with asymmetric performance characteristics

While the previous section relies on exploiting correlations between sensors of the same model, this section and Section 6.4 exploit differences.

6.3.1 Weighting IMUs with different performance characteristics

If the same inertial signal is measured by sensors with significantly different performance then an equal weighting of their outputs does not make sense, so some weighting scheme needs to be developed.

If the combination is of sensors where one is simply better than the other, then the weighting should be relatively straightforward. This becomes complex when a sensor is superior by one metric but inferior by another. There is also the question of how an array consisting of several sensor types should be considered in the integration algorithm.

One idea is to combine the whole array as a VIMU was discussed above (Section 6.1). Consider, where the array consists of a mixture of IMUs type A and B. In the simplest case, where one of the sensor-types consistently under-performs the other, a simple weighting corresponding to their relative errors can easily be established. However, what about, for example, when A has better noise performance, and B better scale factor?

One option is to construct a complicated VIMU combination algorithm. For instance, if A has better noise and B better bias, take a low- and high-pass filter of the output of both with the same characteristic frequency, construct differently weighted combination of the two low-frequency signals and the two high-frequency signals, before recombining the high- and low-frequency parts to create the VIMU output. Another possible example is, if A has better noise and B better scale-factor and/or non-linearity then have the weighting of the combination change with the magnitude of the signal. A final example is: A and B have identical performance at room temperature, but A has bias that varies more than that of B with temperature, then, assuming that they had been calibrated at room temperature, there could be a 50/50 combination at room temperature, but the percentage weighting could change as the temperature becomes higher or lower.

The alternative would be to create two VIMUs one of all the A sensors and another of all the B sensors. These could be fed as separate measurements into the overall integration algorithm. This has the advantage that the precise performance differences need not be determined for creating an optimal VIMU from the two, and it should theoretically perform better than a pre-determined combination. However, it requires more processing power, due to at least estimating the error states separately, but not as much as estimating every sensor in the array separately.

6.3.2 Asymmetric INS requirements

Any navigation system is built to a performance specification, either explicitly or implicitly, by specifying its use for a particular application. The first task when considering an array of sensors with different performance characteristics is to consider whether there is any asymmetry in the performance requirements.

There are few applications where the required navigation system performance is truly the same in all directions. The most common one (for low-cost applications) will be a hand-held navigation system, e.g. in a smartphone. This is because it will be used in a wide variety of contexts [10], e.g. for both pedestrian navigation and vehicle navigation, and its orientation relative to the motion/vehicle/person will not necessarily be fixed due to its hand-held nature.

In most other contexts in which an INS would be used, particularly where its orientation relative to the vehicle/motion is fixed, the required sensor performance is not likely to be the same on all axes, even if the integrated navigation system must achieve an overall target performance that is the same in all 3 dimensions.

If a sensor that only measures a subset of the navigation solution is integrated with the INS, then clearly that has asymmetric implications for the requirement performance of the INS. For example, integrating a magnetometer only measures (local navigation frame) heading (see Section 2.3.2.1), this would make it easier to observe the bias of whichever gyroscope whose axis is most frequently co-incident with the local gravity vector (e.g. the yaw gyroscope for a land vehicle). Another example is that, a barometer provides a measure of height (see Section 2.3.2.2). This is particularly helpful for the accelerometer that most frequently

coincides with the local gravity vector, but also in-directly helps the attitude solution in pitch and roll, due to the compensation for gravity in the specific force to acceleration conversion. This helps the estimation of the pitch and roll gyroscopes.

Even though INS/GNSS provides a full 3D position, its use to estimate INS errors has complex effects. The differing accuracy in the vertical and horizontal directions (see Section 2.3.1) mean that it is directly more helpful to the x- and y- accelerometers than the z. While its indirect correction of the attitude solution makes it more helpful to the x- an y-gyroscopes which can be corrected based on the correction for gravity as above while the heading can only be corrected by motion.

Any vehicle motion constraints (see Section 2.3.3) have by definition more effect in one (body-frame) direction than another, making e.g. the performance of the accelerometers in the direction of travel and the yaw gyroscope much more important than the pitch or roll gyroscope for road vehicles. The directional effect of map-matching techniques (see Section 2.3.4) is more subtle, but they may be asymmetric in some conditions, e.g. in a large flat field, map-matching would not provide useful information horizontally, but vertical position uncertainty would be much smaller.

6.3.3 Asymmetric INS performance

If one were to construct an array of low-cost MEMS IMUs asymmetric performance (different performance for each sensor in the triad) is almost inevitable. MEMS inertial sensor are often fabricated using technology developed for silicon micro-processors. This is based on building up layers of thin silicon wafers and thus there is not full freedom to create 3D structures [20]. For 3-axis IMUs, usually all three sensors of each type are constructed as a unit, so they must share the same construction plane. This is desirable because it means that the triad is well aligned (i.e. orthogonal). As a result, while the two sensors sensitive to motion along or about the in-plane axes (typically x and y) may be of an identical design rotated by 90 degrees, the out-of-plane sensor (typically z) must be of a different design.

In some cases the manufacturer specifies that the performance of the out-of-plane sensor is different e.g. for the ADXL345 [49]. In other cases the specifications are the same for in- and out-of-plane sensors, such as the BMA180 [52]. However, in both these cases the performance characteristics frequently differ, see [8]. This is also the case for other sensor designs. The ST report on L3G4200D performance [48] also gives significant data supporting there being different performances for the different axes, despite all the specifications on the datasheet [24] being identical for all axes.

If the axes' properties are significantly different, the navigation performance of the system will become non-isotropic, perhaps with larger drift in one direction than the others. This in and of itself is not an issue; it is better to acknowledge the anisotropy in the resulting system and tune the filter accordingly than to overstate the errors on the better performing axes and in order to treat it as an isotropic system. Given the cost and space limitations are related mostly to the number of micro-chips being placed, single-axis sensors are not likely to be economic for array techniques. This means that the choice is really, use all

three sensors in the triad or ignore the output of one of them, the latter is unlikely to be worthwhile.

However, if the array is constructed across several PCBs then a combination of in-plane and out-of-plane sensors can face in the same direction, if this is implemented then an 'IMU' made of all the out-of-plane sensors could be considered a type A and another of all the in-plane sensors a type B, and then these treated in one of the ways discussed above.

If the designer wishes to create an array with equal error characteristics in all directions, this is possible even with IMUs with markedly different properties between in- and out-of-plane sensors. One such example would be mounting the sensors on the faces of a platonic solid (such as the tetrahedron used in [163]). If the orientation of the IMUs on their faces are chosen carefully then even if the x- and y-sensors are different there can be equal spread of sensor sensitive axes in all directions. Note that for there need not be one IMU per face, several IMUs could be mounted on each face providing that the number was the same for all the faces, there could even be small anti-parallel arrays as suggested in Section 6.2 on each face.

6.4 Sensor arrays with mixed measurement ranges

It is also possible to construct arrays to exploit the differences between IMU models. An example for combining sensors with dissimilar qualities would be constructing an array of sensors with different measurement ranges.

The measurement range or full scale of a MEMS sensor is the maximum specific force, or angular rate, that it can measure; for example $\pm$ 4g or $\pm$ 500 °/s, respectively. One might assume that if two sensors are of the same quality but different measurement range the lower-range sensor will measure small inertial forces more accurately. The validity of this assumption is examined later in this section.

For any given application, the maximum specific force and angular rate that the INS needs be sensitive to can be specified. This must be based on the peak specific force/angular rate that the INS will be exposed to during regular use. However, for many applications these peak dynamics will only be experienced a very small fraction of the time. So while the system is experiencing low-dynamics the lower-range (assumed to be higher accuracy) sensor is used, but when the forces exceed the range of this sensor the high-range sensor is used, thus avoiding the signal being clipped.

This necessitates that there be some kind of switching method to combine the sensors' output depending on the amplitude of the signal. The simplest implementation of such a system would be a switch between two (or more) sensors, where at some threshold (close to the maximum measurement range of the low-range sensor) the array's output switches from 100% low-range sensor to 100% high range. However for a number of reasons this is not optimal. First, the high-range sensor could still potentially provide useful information in the low-dynamics domain, which might reduce the effect of the noise (or other errors). An example, of this is in [172] where arrays are constructed one of 10g, 100g and 1000g and

another of 2g, 10g and 100g accelerometers, with a very simple weighting regime, where each is weighted in inverse proportion to its range, and zero when out of range. Second, as a sensor gets close to the extremes of its measurement range, one might assume that its reliability and accuracy become lower as it gets close to its maximum output, as for instance the effect of non-linearity increases [47], as it has terms proportional to output squared and cubed. This second point makes an abrupt switch undesirable. However a weighted combination of the two outputs could be designed so that is less affected by these issues. The weighting system at very low dynamics would weight the low-range sensor more highly, and its relative contribution would decrease as the dynamics increase reaching 0% at the end of its range. However, the precise weighting factor used should depend on the relative improvement in performance given by the decrease in measurement range, and its optimal composition depends on the specific relationship between these characteristics.

A key point for this idea, is that the cost, size and power consumption depend on the overall number of sensors. So, the choice is between having an array entirely consisting of high-range sensors, and thus improving performance by the expected $\sqrt{n}$ throughout the range, or whether to switch *part* of the array to lower-range sensors, potentially improving the low-dynamics performance more, but at the expense of (some of) the high-dynamics improvements. So the question is whether the lower-range performance is better by a sufficient margin. But here it is important to consider the fraction of the time the sensors are going to be used outside the low-range. That is, a 5% lower errors for 99% of the time might be worth 50% higher errors during the remaining 1%

This idea assumes that the lower-range sensor performs better. One error that certainly will scale with measurement range is quantisation error. This is created when a continuous quantity is converted to a discrete number of levels by an analogue-to-digital converter (ADC). If two sensors both have the same word-length ADCs, the number of discrete levels they can measure will be the same. So the smallest increment that can be measured will scale with the measurement range, and thus the quantisation error will increase for higher measurement range sensors. Additionally, one might hypothesise that the magnitude of other errors such as noise might scale with dynamic range, e.g. it might be equally difficult to make a ±16g accelerometer accurate to ±160 milli-g as to make a ±4g accelerometer accurate to ±40 milli-g.

Sensors that have very high ranges such as the 200g ADXL375 [50] do have poorer accuracy than their lower-range equivalents. This sensor has a min/max scale factor tolerance of 10%, bias typical ± 400 and min/max ± 6000 milli-g and noise 5 mg/$\sqrt{\text{Hz}}$. This is considerably worse than even the several years older 2g-to-16g ADXL345 [49], e.g. its noise is 0.42mg/$\sqrt{\text{Hz}}$. This means that an array of 100 ADXL375s would still have worse noise performance than a single ADXL345. Thus, it would certainly be worth constructing a mixed-range array if 100+ g or 10000 °/s ranges are required.

If the INS application does not require these very high measurement ranges, the evidence is not so clear cut.

When comparing IMU models in the sub-20g and sub-5000 °/s, to ascertain the improvement in performance that a lower-range sensor will have over a high-range sensor it is difficult

to determine when sensors are of equivalent quality. This is because the performance of low-cost MEMS sensors has improved significantly, so slightly older designs or designs from different companies are not necessarily of equivalent quality despite being at the same price-point. One way to get around this problem is to use a sensor type that has a programmable range, then one can compare the same sensor running at different ranges.

Many digital inertial sensors frequently have configuration options which include different range settings. For example, for the Invensense MPU-9250 IMU, used in this thesis' hardware (see Appendix A), the gyroscopes can be set at ± 250, 500, 1000 or 2000 °/s and the accelerometers can be set to ± 2, 4, 8 or 16g [23]. The remaining specifications of the IMU may or may not be defined with reference to the range at which the sensor is set. In the case of the MPU-9250 the "accelerometer sensitivity change versus temperature" is defined at one specific range (± 2g), and not at others and all other specifications make no reference to range, and so are implicitly defined to be range-setting-independent. On the other hand the Bosch BMA-180 accelerometer has possible range settings or ± 1, 1.5, 2, 3, 4, 8 or 16g and has a separately defined sensitivity (scale factor) error at every one of between 1.5% and 3% [52]. The non-linearity also varies between ±0.15 and ±0.75 % FS depending on the range. The Bosch BMI055 IMU also has programmable measurement ranges of ± 2, 4, 8 or 16g and ± 125, 250, 500, 1000 or 2000 °/s, but the all the sensors errors are only specified for at one particular range (2g and 2000 °/s), the sensors' performance at other range settings is unspecified [45].

This is of course only the *specified* performance, the actual performance of the sensors may well differ. It is unlikely that the sensor performance with unspecified range settings is better, for simple marketing reasons.

The report on STMicrotronics L3G4200D gyroscope performance [48] shows the performance of all the errors tested at all 3 possible range settings (250, 500 and 2000 °/s). It does not show any significant differences for sensitivity change with temperature or zero-rate-level (bias) change with temperature at the different ranges. It does show, as specified a worse distribution of biases at higher range, and a worse non-linearity. Scale factor is only shown at one range. Surprisingly, the noise is almost identical and in fact slightly worse for the lower-range settings than the higher range settings.

For this type of programmable-range sensor, it is very unlikely, for reasons of cost, that there are multiple sensitive elements for the different range settings, so a single proof-mass is used for all ranges with some electronic changes. In the examples where designs or CAD drawings are published, e.g. [48], this appears to be the case. Thus, a 2g-to-16g accelerometer is mechanically a 16g accelerometer, at least in the open-loop case, where the change could be as simple as amplifying the signal before digital conversion. This would explain why the noise characteristics do not change, One would still expect the sensor to be more linear at its low setting, assuming that non-linearity is dominated by quadratic or higher order terms (see Section 2.2.6), and one might expect different biases, given the specifications are frequently different. The sensor will certainly be more precise, in that the physical quantity represented by 1 LSB is smaller, but may not be more accurate. It might be simply because the sensor is more accurate at measuring small ($< 1g$) forces, irrespective of the range setting. If the only improvement from switching to a lower-range setting is to

the bias, which was shown in Chapter 4 to be the easiest error to calibrate, this mixed-range idea is unlikely to make a worthwhile improvement.

6.5 Individual Calibration or VIMU calibration

When using an IMU array, It is also worth considering whether the user calibration procedure outlined in Chapter 4 should be conducted on the individual sensors in the array, or the array-averaged VIMU. For static calibration, in general it should not make a difference whether the sensors are calibrated individually or as a complete VIMU. For the dynamic calibration presented in this thesis, where a set of 360° rotations are used to calibrate the sensors using no other information than that they are 360° about a particular axis, the final performance should not be different. However, for some other dynamic calibration procedures that use magnetometers and/or the accelerometer signals to calculate a partial attitude solution, such as [154, 156, 161], then using an array-averaged accelerometer or magnetometer signal should produce better results for the gyroscope calibration. This is because the algorithm assumes the accelerometer or magnetometer signal is true to calculate $\widehat{\mathbf{A}_\mathbf{g}}$, so a less noisy signal accelerometer or magnetometer signal which also has equal or smaller systematic errors should produce a better $\widehat{\mathbf{A}_\mathbf{g}}$.

However, calibrating the sensors individually has some advantages. First, if there are individual sensor outputs then the results can be checked against the specifications as a way to detect gross errors, which may indicate a faulty sensor, if it only affects one sensor, or a mis-conducted calibration procedure, and prompt the user to repeat the procedure. Second, if any sort of fault detection and exclusion algorithm or dynamically changing weighting algorithm is used then if a sensor is down-weighted or ignored, the complete VIMU calibration becomes incorrect, so individual calibrations are required.

Applying a single VIMU correction is less computationally intensive than applying the corrections to every sensor individually before averaging. However, a single VIMU correction could be calculated from the individual correction factors once, as well as by putting the VIMU output into the calibration algorithm, and so the in-run computational cost could be the same in either case.

6.6 Experimental support for these techniques

6.6.1 IMU error coefficient distribution

The array techniques presented in this chapter rely on certain characteristics of the distribution of measurement errors. The general array effects (Section 6.1) rely on independence across the sensors. The anti-parallel array ideas in Section 6.2 only make a difference when the distribution of errors is asymmetrical about zero, i.e. not zero-mean. Incidentally they also remove any skewness of the distribution. The techniques in Section 6.3 rely only on there being differently performing sensors, which is a given. However, a combination of differently performing IMUs being better than an equal sized array of identical ones, needs

Error	typical spec.	Mean	Standard Deviation	RMS
$\widehat{\boldsymbol{b_a}}(x)$ (ms^{-2})	± 0.59	0.397	0.382	0.5414
$\widehat{\boldsymbol{b_a}}(y)$ (ms^{-2})	± 0.59	0.193	0.173	0.2550
$\widehat{\boldsymbol{b_a}}(z)$ (ms^{-2})	± 0.78	-0.508	0.593	0.7644
$\widehat{\boldsymbol{b_g}}(x)$ ($^\circ$/s)	± 5	0.0704	1.295	1.250
$\widehat{\boldsymbol{b_g}}(y)$ ($^\circ$/s)	± 5	0.625	0.8615	1.0393
$\widehat{\boldsymbol{b_g}}(z)$ ($^\circ$/s)	± 5	0.5700	0.7006	0.8836
$\widehat{\mathbf{A_a}} - \mathbf{I_3}(x,x)$	$\pm 3\%$	0.00205	0.000883	0.00222
$\widehat{\mathbf{A_a}} - \mathbf{I_3}(y,y)$	$\pm 3\%$	0.00201	0.000468	0.00206
$\widehat{\mathbf{A_a}} - \mathbf{I_3}(z,z)$	$\pm 3\%$	0.00784	0.00298	0.00836
$\widehat{\mathbf{A_g}} - \mathbf{I_3}(x,x)$	$\pm 3\%$	-0.00641	0.00164	0.00660
$\widehat{\mathbf{A_g}} - \mathbf{I_3}(y,y)$	$\pm 3\%$	0.00106	0.00279	0.00289
$\widehat{\mathbf{A_g}} - \mathbf{I_3}(z,z)$	$\pm 3\%$	-0.00235	0.00249	0.00336

Table 6.1: Summary statistics of experimental estimates for $\widehat{\boldsymbol{b_a}}$, $\widehat{\boldsymbol{b_g}}$ and scale factor errors

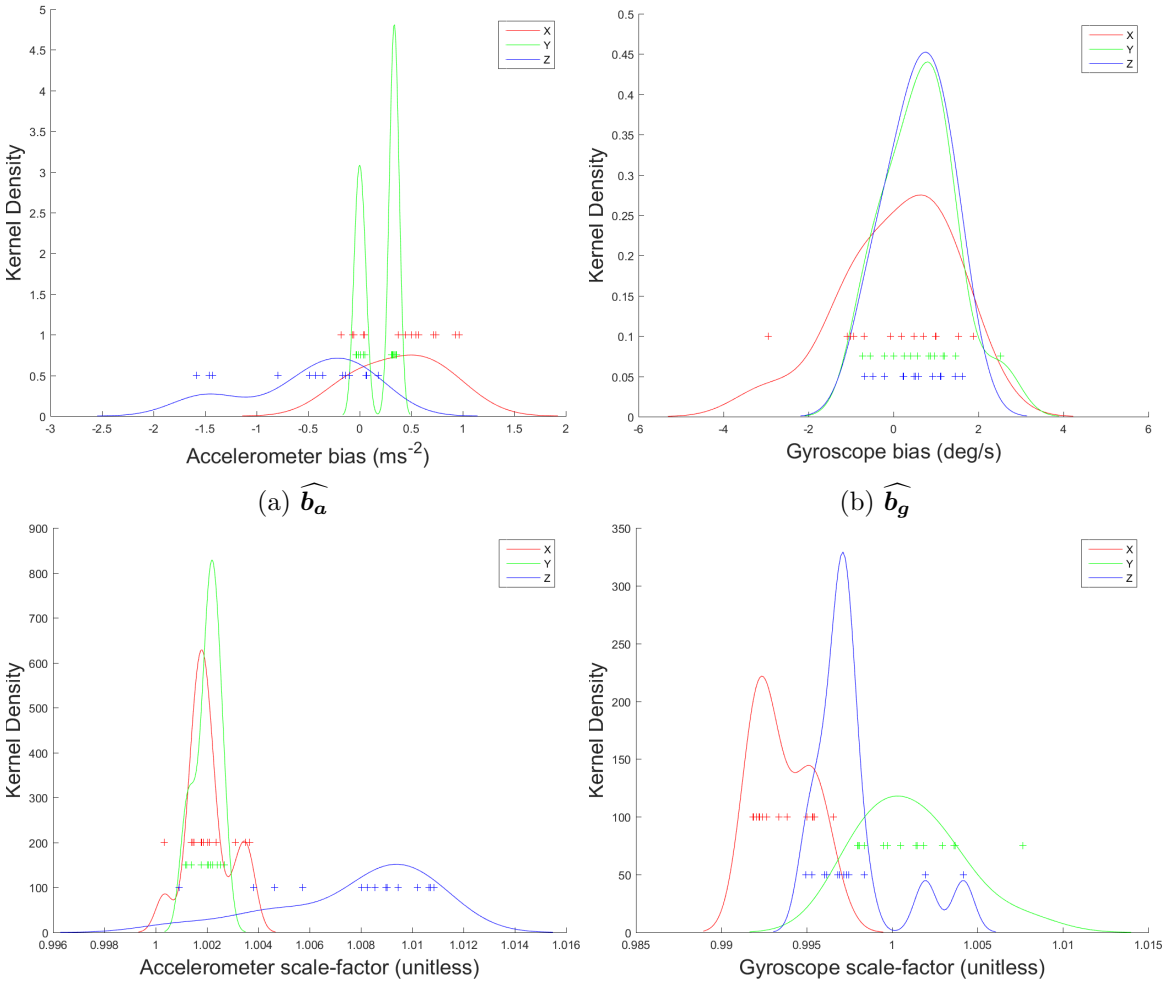
an IMU A to be better in one way than IMU B and worse in another. In order for the mixed measurement arrays described in Section 6.4 to work then low-range sensors must be substantially better performing than the high-range ones.

In Chapter 5 of this thesis some experimental results for an array of MPU-9250 sensors [23] were presented. This shows that, for the 14 working sensors in the array there is effectively no difference in performance at the different range settings in terms of noise (Section 5.1) or significant improvement for the other errors calibrated (Section 5.4). One might reasonably hypothesise that the only effect of changing the range setting is to amplify the signal. The quantisation level is sufficiently small in comparison with the noise that it does not make a significant difference to the results. This makes an array of differently set MPU-9250s not a good candidate for the techniques proposed in Section 6.4.

There are significant differences between the sensors' axes. However, the experiments show that the z-axis accelerometers are significantly worse in uncalibrated scale-factor and bias than the x- or y-accelerometers (4× and 1.5×, respectively), as well as being noisier (1.5× see Table 5.4). By contrast, the z-gyroscope is slightly less noisy than the x- or y-gyroscopes (0.8×).

Using the calibration procedure presented in Chapter 5, the estimates of the IMU errors bias, alignment and scale factor matrix, and gyroscope g-dependence for each of the 14 sensors can be used as a small sample to examine the of the starting distribution of errors. The resulting estimates for gyroscope g-dependent error are so small that the measurement uncertainty and is of almost equal magnitude (see Section 5.2, particularly Figure 5.5, and Table 5.11) so their distribution cannot be determined. So the distribution of $\mathbf{G_g}$ will not be discussed here.

Summary statistics for the other error sources calibrated for are presented in Table 6.1. As can be seen the mean is significant in proportion to the standard deviation, for every error source but x-gyroscope bias ($\widehat{\boldsymbol{b_g}}(x)$). This is also reflected in the fact that the RMS is significantly greater than the SD. This means that making the distribution zero-mean by



(a) $\widehat{\boldsymbol{b_a}}$ (b) $\widehat{\boldsymbol{b_g}}$

(c) $\widehat{\mathbf{A_a}}$ diagonal elements (d) $\widehat{\mathbf{A_g}}$ diagonal elements

Figure 6.2: Kernel density estimates from calibration of 14 sensors. X-, Y- and Z-axes in red, green and blue, respectively. Individual samples are also marked.

using the opposing axes would make an improvement. For example, the x-axis accelerometer bias would have a mean of zero and standard deviation of 0.19 ms^{-2} in an anti-parallel array of 4 sensors, rather than mean of 0.40 ms^{-2} and SD of 0.19 ms^{-2}, which would need to be modelled in a zero-mean assumption KF with an uncertainty of nearly 0.6 ms^{-2}, about a factor of 3 improvement from changing the alignment.

The 14 sensors' systematic errors are also visualised in Figure 6.2 in the form of Kernel density estimation (KDE) [189, 190]. This is used here as a visualisation technique and is used rather than a more standard histogram because there are only a small number of points which will not be adequately represented by binning values. Additionally, the continuous nature of the estimates can be represented by KDE. The KDE is an estimate of the probability density of a distribution from a finite number of points taken from it, which allows smoothing. Each point is replaced by a kernel with the property that the area under the curve integrates to 1, in this case a fixed-width Gaussian, with a degree of freedom known as the bandwidth parameter, which controls the level of smoothing. In this case the bandwidth parameter is the variance of the Gaussian, i.e. its width. The kernels representing each points are combined to form an overall estimate of the probability

density. The bandwidth setting presented here is that which is optimal when the distribution is normal. The bandwidth, u, is calculated as $u = \widehat{\sigma}\left(\frac{4}{3n}\right)^{0.2}$, where n is the number of data points and $\widehat{\sigma}$ is calculated as follows, $\widehat{\sigma} = \frac{1}{0.6745}\,\text{median}_{1\leq i\leq n}\left(\,|y_i - \text{median}_{1\leq j\leq n}\,(y_j)\,|\,\right)$, where y_i are the data points. The sample points, about which the kernels are centred, are also shown in Figure 6.2, by crosses (+) of the appropriate colour.

Figure 6.2 shows that nearly all the distributions are not only non-zero-mean (as could be seen in Table 6.1) but also appear skewed, particularly $\widehat{\boldsymbol{b_g}}$. However, there are not sufficient samples to make this statement reliably.

The experimental results show the 14 sensors on the testbed perform well within their stated specification ($\boldsymbol{b_a}$ 0.78–1.47ms^{-2}, $\boldsymbol{b_g}$ 5 °/s, both scale factors 3%) . However, it is notable that the scale factor errors are much better than specified: the accelerometer has an RMS over the sample of between 0.2 and 0.8% as compared to the 3% specified.

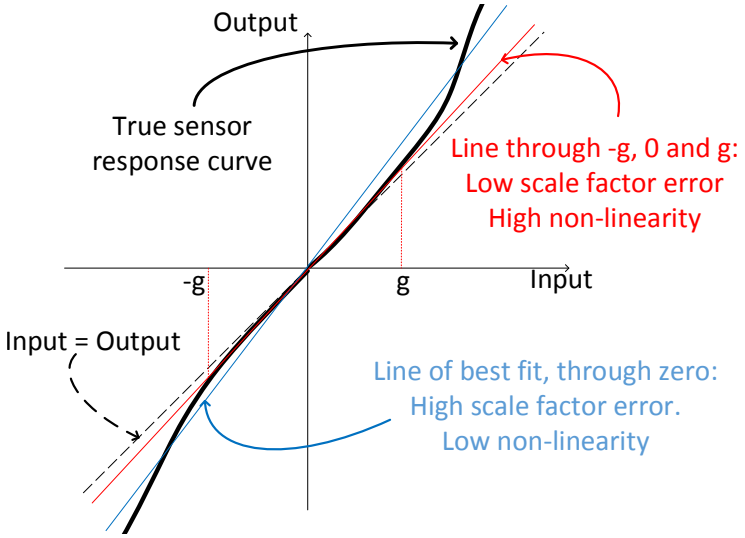


Figure 6.3: An illustration of different possible definitions of scale factor error and non-linearity.

This highlights a problem with getting a consistent comparison between the specification and the calibration outputs.

The MPU9250 datasheet [23] describes the non-linearity parameter as "Best fit straight line; 25°C" for the gyroscope and as "Best Fit Straight Line" for the accelerometer. It also define the quantity referred in this thesis as $\boldsymbol{b_a}$ as "Zero-G Initial Calibration Tolerance". The the implication of these two points is that they are taking the input-output curve, removing $\boldsymbol{b_a}$ so that this input-output curve passes through the origin and then fitting a straight-line to this curve that passes through the origin. This is illustrated in Figure 6.3 by the blue line. This is not the same as fitting a straight line to the curve and then using its y-intercept as $\boldsymbol{b_a}$, although they may also be doing that. However, the calibration procedure presented in Chapter 4 of this thesis and used for the experiments in Chapter 5, does not do this. The procedure effectively takes the output when the input is −g, 0 and g and fits a straight line between these 3 points. In fact, the Technique 2 bias estimation method only fits the straight line between −g and g. This is illustrated in Figure 6.3 by the blue line. If one assumes that the non-linearity error increases as the sensor's input increases, which is there

is evidence to suggest is a reasonable assumption in general, see Section 2.2.6. As can be seen the former approach (blue line) results in a greater scale factor error but a smaller non-linearity than the latter approach.

Note that, this effect is common to any 6-position calibration method, see Section 2.5.1.3, and any of the static calibration methods that use more than 6 positions are at best fitting a straight line through $-$g, 0, g and a small number of points in between, e.g. in [161] they are also effectively fitting it through $\frac{g}{\sqrt{2}}$. As can be seen in Figure 6.3 this would have virtually the same effect.

A similar but potentially less extreme effect is produced by the user-calibration procedure in estimating gyroscope scale factor. The manufacturers' method is to fit a straight-line through the input-output curve with an (implicitly) equal weight to every point on the line. On the other hand the user-calibration procedure effectively weights the points on the input-output curve by their frequency of occurrence in the recorded data, and because the user must be careful not to saturate the gyroscope by rotating too fast, there will be a much greater frequency of slower angular rates, leading to a similar problem to that outlined for the accelerometer above where the scale-factor is determined predominantly from the lower magnitude angular rates. Additionally if calculating $\widehat{\mathbf{A_g}}$ from the set B manoeuvres then the straight line will be fitted to, for example, only positive angular rates, as the rotation only takes place in one direction. This is remedied in set D because while the rotation that appears to be in the same direction to the user (table frame) it is in the opposite direction in the sensor frame for the rotations on opposite faces. This is a fundamental limitation, which can be summarised as: it is only possible to fit the line to the data available.

6.6.2 Bias variation with temperature

An experiment to determine the approximate response to temperature of the accelerometers was carried out, to see if there was a significant chance of the anti-parallel approach working for the MPU-9250 sensors in the custom hardware test-bed described in Section A.2.

First, some calibration procedures were run which are discussed in Section 6.6.3, but are not relevant to this section. Then the test-bed was left static for around a minute. Then the sensors were heated using a hair-dryer from the side at a distance of 85cm. For the first 2.5 minutes the hair-dryer was on 'cool', which caused a drop in internal temperature, then for next for 2.5 minutes on 'medium' then 2.5 minutes on 'hot'. Finally it was heated from 40cm straight above on 'hot' for 2.5 minutes, then left to cool for approximately 15 minutes. The outputs of the IMUs internal temperature sensors during this time are shown in Figure 6.4.

In Figures 6.5 to 6.10, the smoothed raw output of each type of inertial sensor is presented against the individual sensor's smoothed internal temperature sensor output. The data from after 3600 sec when the sensors are no longer being heated, and so are cooling down, are shown by the dotted lines. The different colours are the different sensors, each sensor is given the same colour in Figures 6.4 to 6.10, enabling one to see that the bright green sensor which drifts the most in Figures 6.5 and 6.6 is from the same IMU.

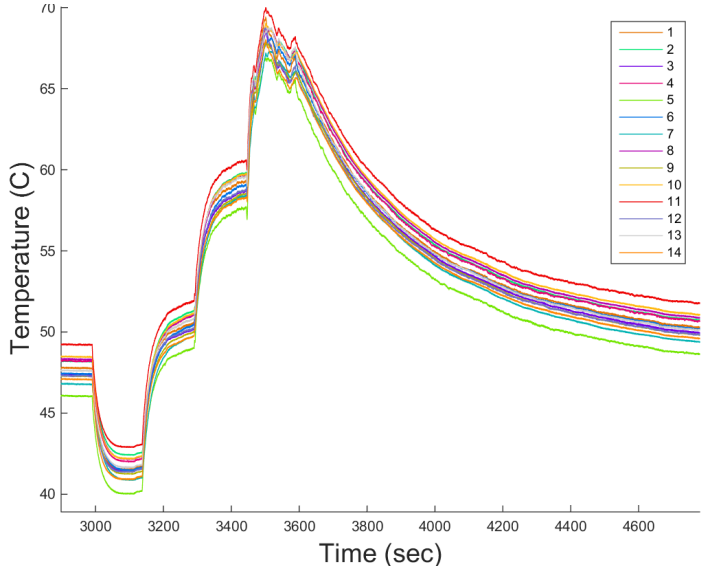


Figure 6.4: The temperature profile during the heating experiment. Each colour is a different sensor's unfiltered temperature sensor output.

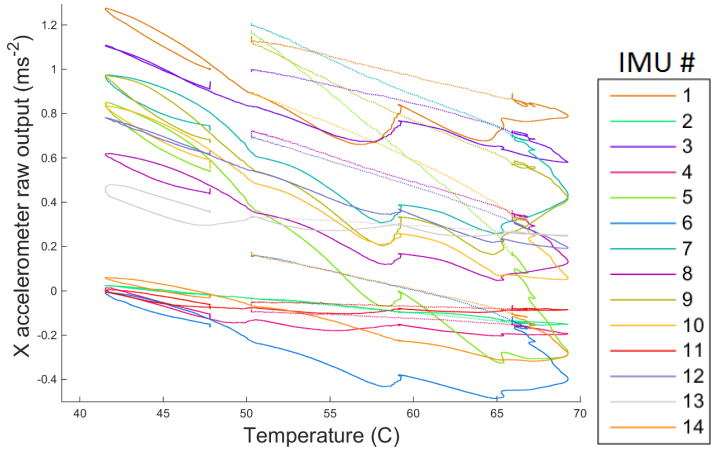


Figure 6.5: The response of the X accelerometer to heat. Raw output of specific force and temperature filtered over 5 seconds (1250 samples). The dotted part of the line is the second part of the data when the sensor was cooling down.

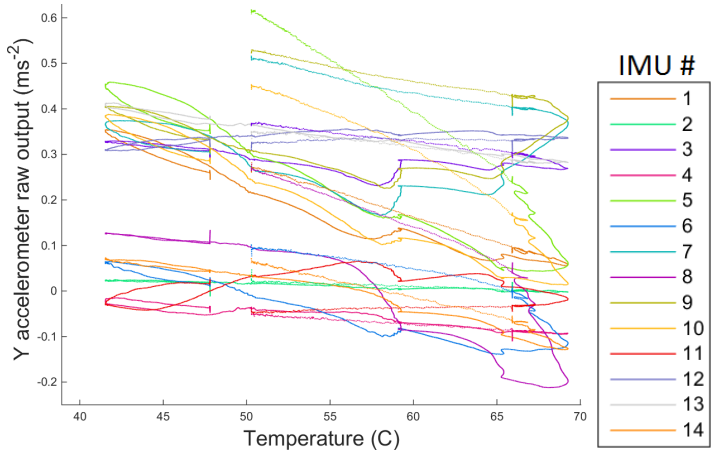


Figure 6.6: The response of the Y accelerometer to heat. Raw output of specific force and temperature filtered over 5 seconds (1250 samples). The dotted part of the line is the second part of the data when the sensor was cooling down.

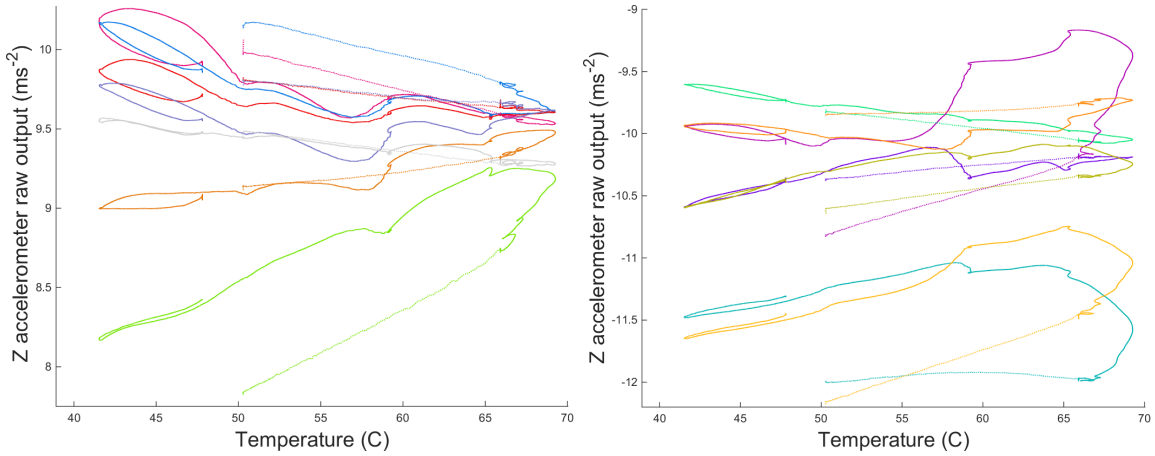


Figure 6.7: The response of the Z accelerometer to heat. Raw output of specific force and temperature filtered over 5 seconds (1250 samples). The dotted part of the line is the second part of the data when the sensor was cooling down. Two graphs are necessary as half the sensors read $\approx g$ and the other half $\approx -g$.

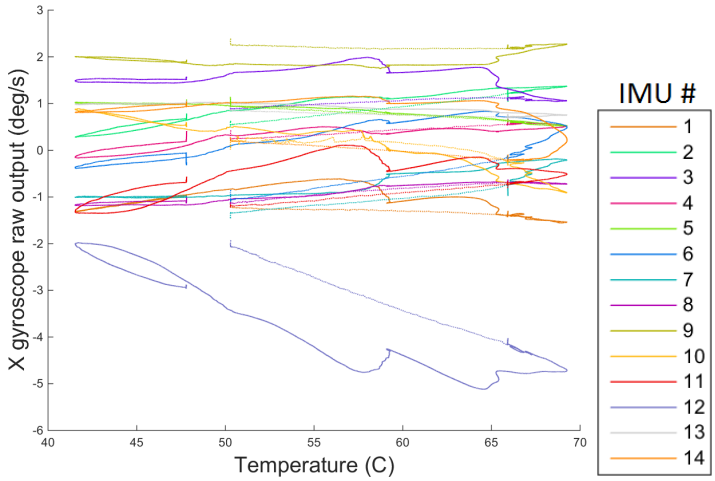


Figure 6.8: The response of the X gyroscope to heat. Raw output of angular rate and temperature filtered over 5 seconds (1250 samples). The dotted part of the line is the second part of the data when the sensor was cooling down.

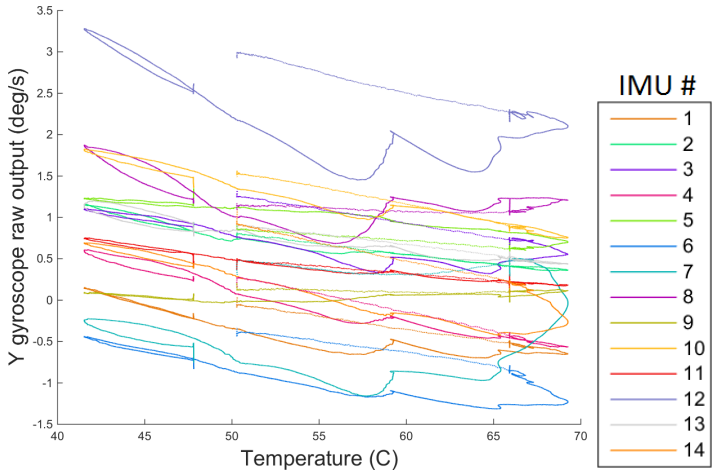


Figure 6.9: The response of the Y gyroscope to heat. Raw output of angular rate and temperature filtered over 5 seconds (1250 samples). The dotted part of the line is the second part of the data when the sensor was cooling down.

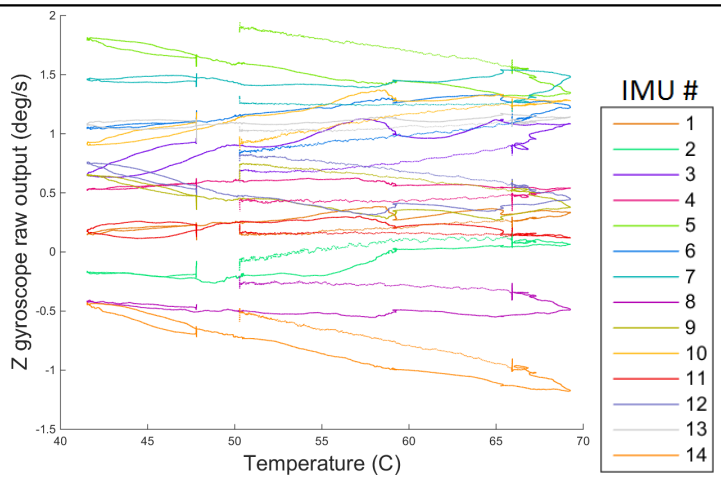


Figure 6.10: The response of the Z gyroscope to heat. Raw output of angular rate and temperature filtered over 5 seconds (1250 samples). The dotted part of the line is the second part of the data when the sensor was cooling down.

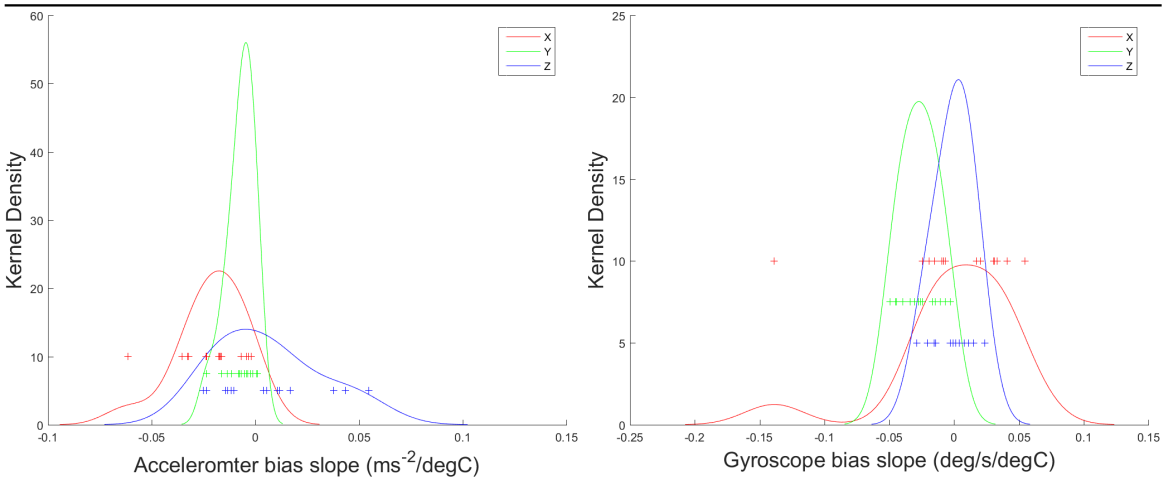
There is clearly some hysteresis in the temperature response and/or some lag between the temperature sensor detecting the change and sensors readings being affected. This is not a precise temperature calibration, rather an indication of the trend of temperature response.

In Figures 6.5 and 6.6 it can be seen that the general trend in temperature response for the x- and y-accelerometers are negative with temperature. In Figure 6.7 the trend for the z-accelerometer is more complex. The trend is not consistently positive or negative and the magnitude of the change is much greater. However, the scale factor change with temperature is also a factor for this measurement, as illustrated by the fact that it must be split into $\approx 10\,\mathrm{ms}^{-2}$ and $\approx -10\,\mathrm{ms}^{-2}$ graphs, for the sensor top and bottom of the PCB, respectively. The approximately 30°C change in temperature, and the specified ±0.026 % / °C typical scale factor change with temperature, means a typical change should be around 0.75% or 0.075 ms^{-2}. Thus if the specification is believed then this is can only account for a small amount of the >0.5 ms^{-2} change in output.

The x-gyroscopes (Figure 6.8), for all but 3 of the 14, show a slight positive trend of bias with temperature, although the 3 that show a negative trend have greater magnitudes. The y-gyroscopes (Figure 6.9) generally show negative trends. This difference which would imply that a 2-sensor opposed array, which opposes x of one sensor with -y of the other would not work well for this sensor model. The z-gyroscopes (Figure 6.10) generally show smaller variations, but are neither consistently positive or negative. Performing a simple linear fit to

Error	typical specification	Mean	Standard Deviation	RMS
$\partial \boldsymbol{b_a}/\partial T\,(x)$ (ms^{-2}/°C)	±0.0147	-0.0201	0.0160	0.0260
$\partial \boldsymbol{b_a}/\partial T\,(y)$ (ms^{-2}/°C)	±0.0147	-0.00727	0.00690	0.00985
$\partial \boldsymbol{b_a}/\partial T\,(z)$ (ms^{-2}/°C)	±0.0147	0.00618	0.0251	0.0249
$\partial \boldsymbol{b_g}/\partial T\,(x)$ (°/s/°C))	±30 °/s (-40 to 85 °C)	0.000516	0.0472	0.0455
$\partial \boldsymbol{b_g}/\partial T\,(y)$ (°/s/°C))	±30 °/s (-40 to 85 °C)	-0.0266	0.0148	0.0302
$\partial \boldsymbol{b_g}/\partial T\,(z)$ (°/s/°C))	±30 °/s (-40 to 85 °C)	-0.000366	0.0150	0.0144

Table 6.2: Summary statistics for estimated bias-temperature slopes



(a) $\partial \boldsymbol{b_a}/\partial T$ (b) $\partial \boldsymbol{b_g}/\partial T$

Figure 6.11: Kernel density estimates of bias-temperature slope estimates of 14 sensors. X-, Y- and Z-axes in red, green and blue, respectively. Individual samples are also marked.

the post-3600 seconds (cooling) part of the data (the dotted lines) can generate approximate temperature change coefficients. Summary statistics for these 14 sets of estimates are given in Table 6.2. These are also visualised as Kernel densities in Figure 6.11. These summary statistics support the observations made above from Figures 6.5 to 6.10. The x- and y-axis accelerometers show more potential for improvement than the z-axis one. Additionally the much lower drift of the y-axis accelerometers is noteworthy. For the gyroscopes the anti-parallel technique looks most promising for the y-axis and no use at all for the z-axis (which is effectively zero-mean). In discussing Figure 6.8, the distribution of the x-gyroscopes slopes was noted mostly positive, but with a few highly-negative sensors. This can be seen clearly in the red (X) line of Figure 6.11b.

6.6.3 Dynamic tests of the array techniques

In order to demonstrate that these array techniques work in practice a set of further dynamic tests was carried out, whose results can be compared with the experiments in Section 5.6.2.

First a full array average was made by taking the mean of the 14 available sensor measurements before running the measurements through the inertial navigation equations, using the same data as used in Section 5.6.2. Figure 6.12 presents the profile calculated from running the uncalibrated data (green line - as in Figure 5.25) and the calibrated data (orange line - as in Figure 5.26). This figure shows a dramatic improvement relative to either of the figures from Section 5.6.2. It is worth noting, that the test-bed is nearly an anti-parallel array, see Figure A.4, only the two broken sensors prevent this from being the case. Thus this improved performance is not only a demonstration of the basic array averaging of Section 6.1 but also the anti-parallel techniques in Section 6.2. Note that the difference between the uncalibrated and calibrated results is rather smaller than might have been expected from comparing the spreads of results from the sensors individually in Figures 5.25 and 5.26, particularly the velocity solutions. This is an indication that the anti-parallel array cancelling of the uncalibrated biases is also having an effect.

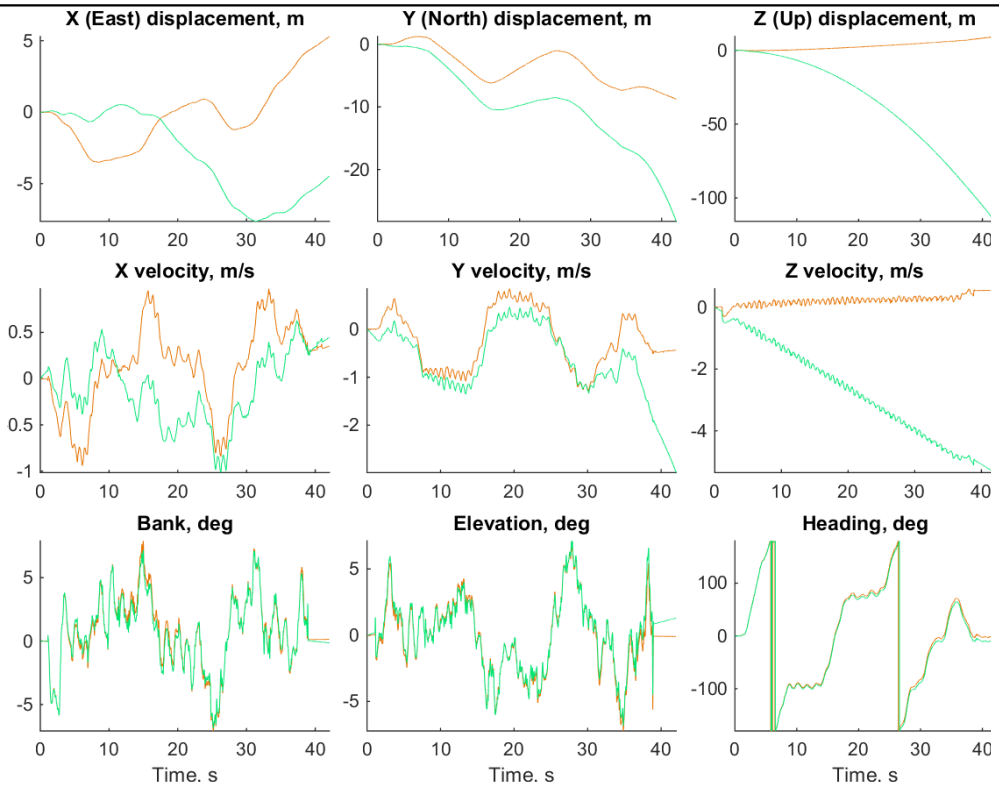


Figure 6.12: Motion profile obtained from a short pedestrian motion test by running inertial navigation on the mean of the whole array's uncorrected (green) and corrected (orange) inertial data.

A second experiment is also run to examine the effect of the anti-parallel arrays on the bias drifts with temperature. The testbed was allowed to warm up for 15 minutes then 4 complete calibration procedures were carried out. Then the heating described in Section 6.6.2 was conducted, which will not be considered in this section. After this the test-bed was briefly re-heated with the hair-dryer, and then left static for 1 minute. Then the a short period of pedestrian motion (about 40 seconds) was recorded. Then two further calibration procedures were recorded. The temperature outputs during this entire experiment are presented in Figure 6.13.

In order to compare aligned (i.e. basic) and anti-parallel arrays subsets of the array were combined. In order to compare the performance the sensors output was calibrated using the calibration procedure estimates from the 4^{th} procedure, i.e. from before the heating began. In the original design, see Figure A.4, there were 8 sensors on each side, 2 at each 90° rotated orientation, but two sensors are broken. Thus, there six possible aligned pairs whose output can be used as the null-hypothesis and is presented in Figure 6.14. This can be compared to sets of xy-opposed pairs where the z-axes are in opposite directions, and the '$+x$' direction of one sensor is the '$-y$' direction of the other. This is presented in Figure 6.15. All 12 possible pairs are presented. There are also 12 possible pairs where the z-axis directions are in the same direction, but in the horizontal plane they are opposed, i.e. the '$+x$' direction of one sensor is the '$-x$' direction of the other (this is the orientation depicted in Figure 6.1). This is presented in Figure 6.16.

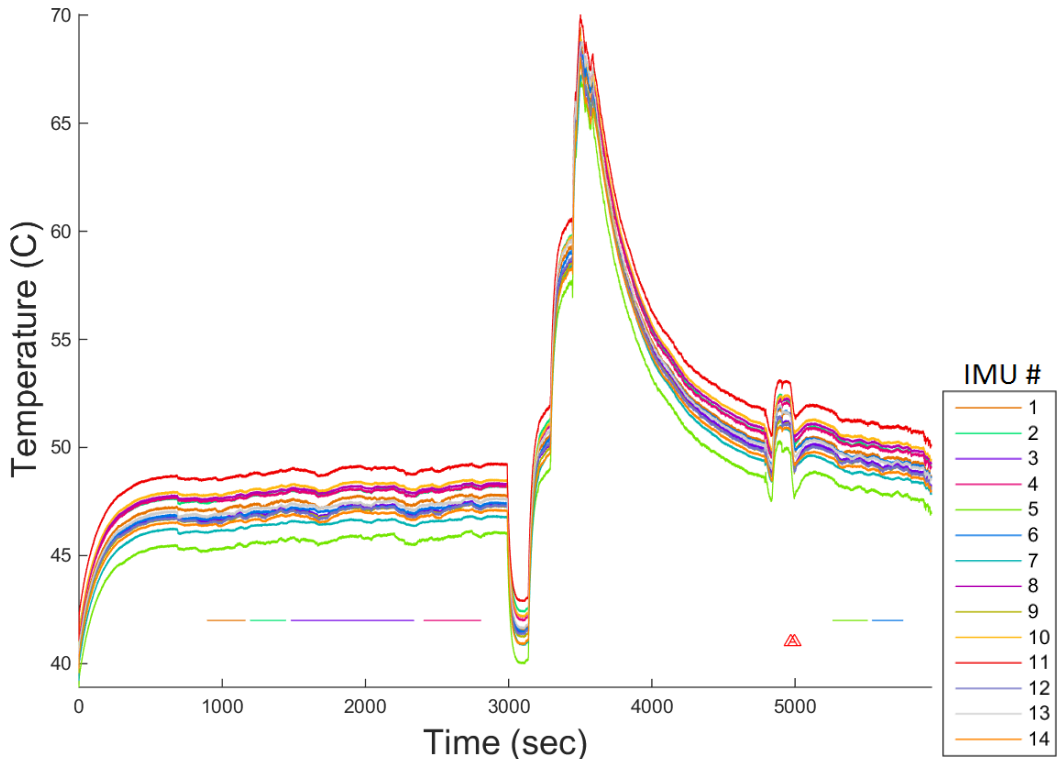


Figure 6.13: The output of the internal temperature sensors during the experiment. The duration of each of the 6 calibration procedures are marked by coloured horizontal lines, the walking test by the red triangles. Compare to Figure 5.12

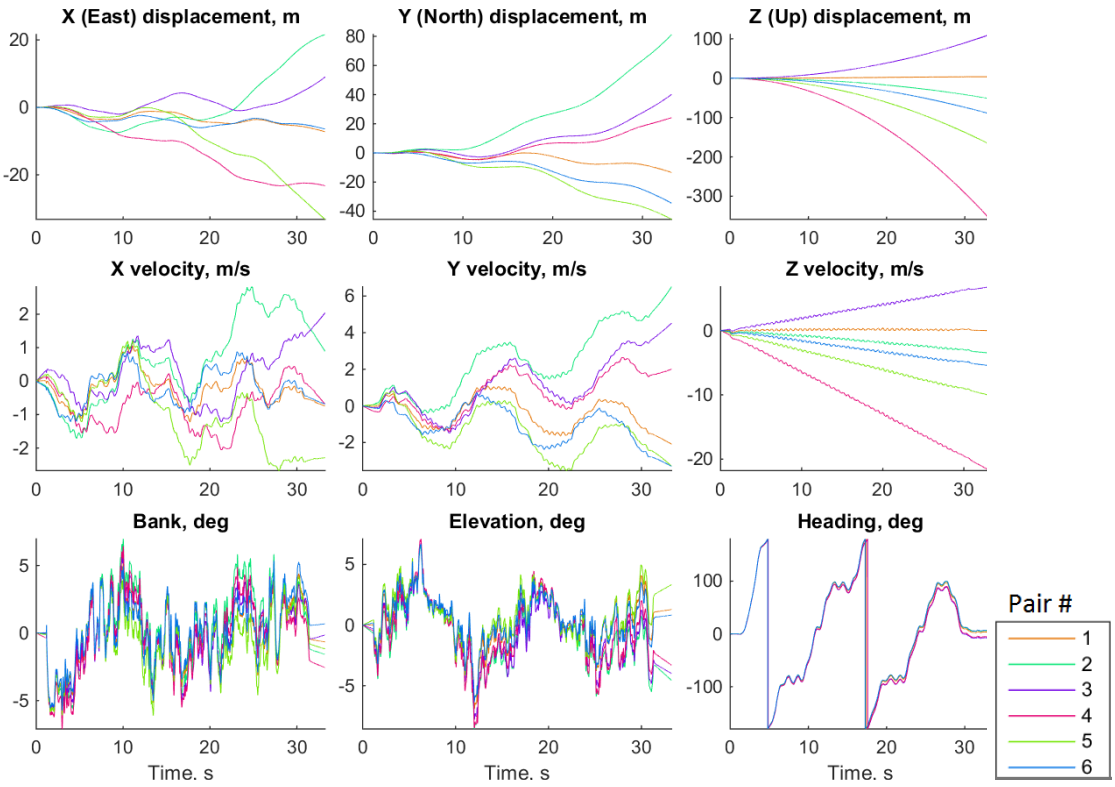


Figure 6.14: Motion profile obtained from a short pedestrian motion test by running inertial navigation on the mean of each aligned pair's corrected inertial data.

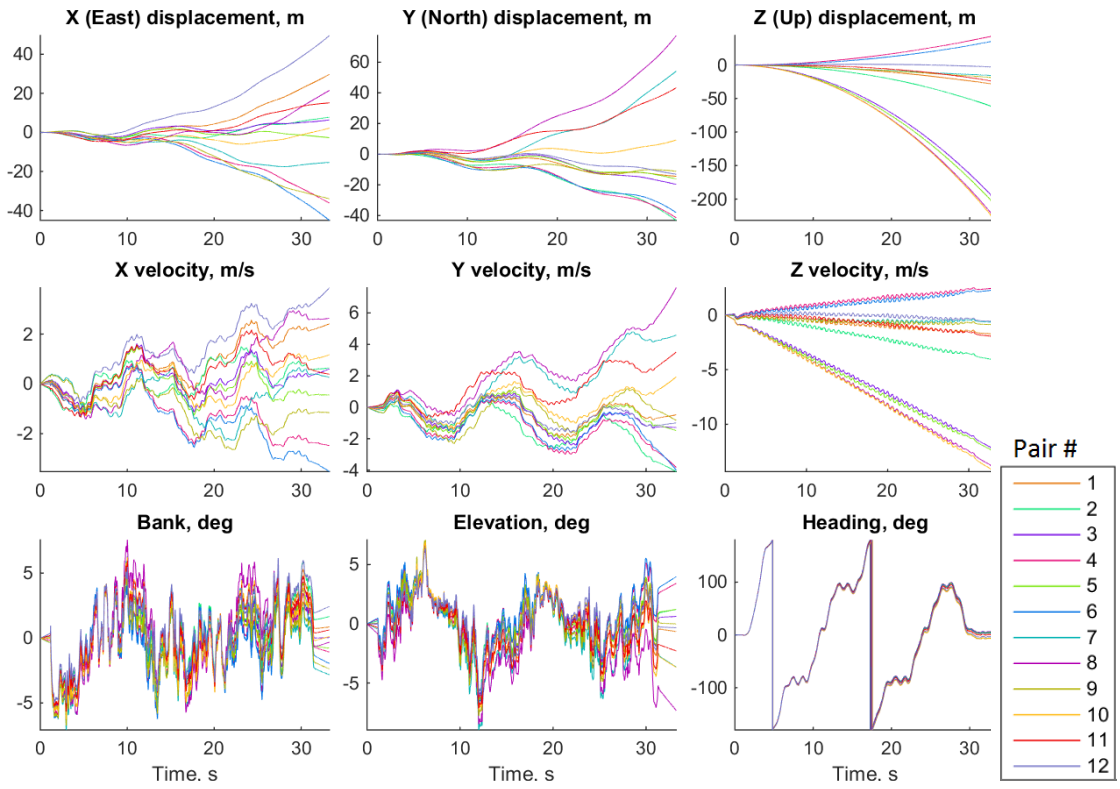


Figure 6.15: Motion profile obtained from a short pedestrian motion test by running inertial navigation on the mean of each xy-opposed pair's corrected inertial data.

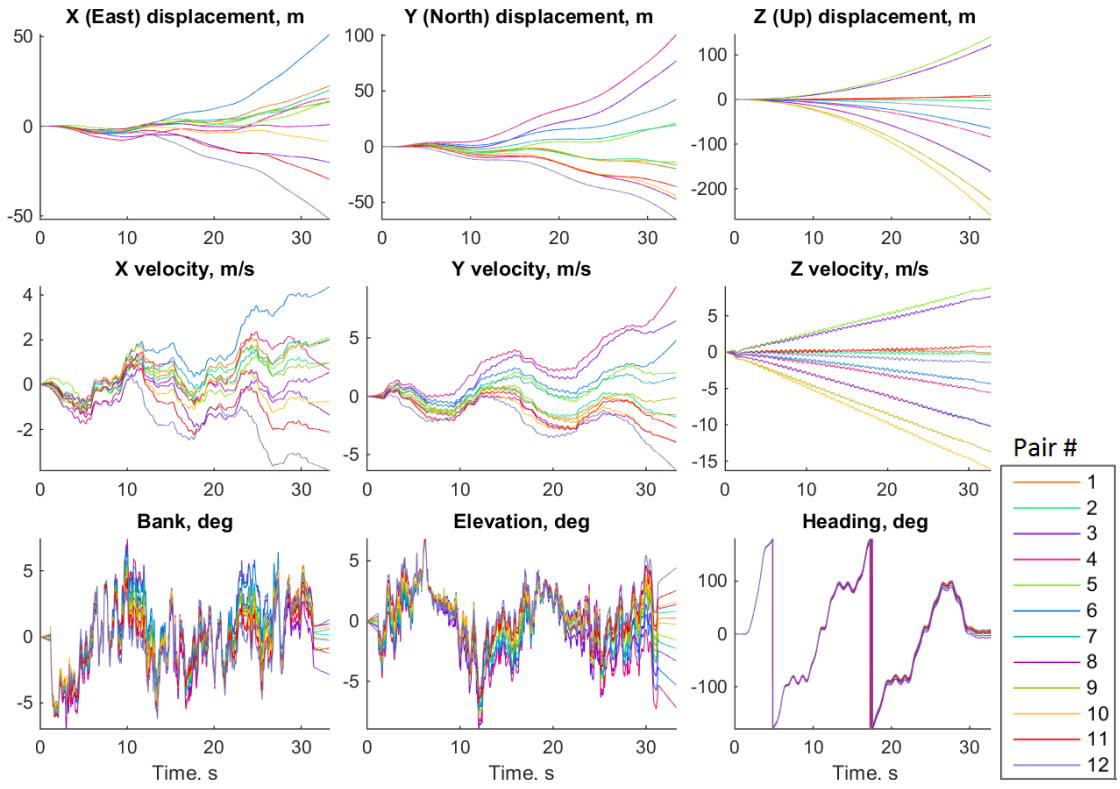


Figure 6.16: Motion profile obtained from a short pedestrian motion test by running inertial navigation on the mean of each horizontally-opposed pair's corrected inertial data.

It is also possible to construct fully opposed sets of 4 IMUs, by combining a horizontally-opposed pair of IMUs on each side of the PCB (see Section 6.2). These would be expected to perform better than the 2 IMU pairs, due to $\sqrt{4}$ rather than $\sqrt{2}$ averaging of both noise and error coefficient distribution SDs.

There are not 4 aligned sensors on the PCB to which these can be compared, so this is presented only for completeness. Their input is presented as Figure 6.17, all 20 possible sets are presented. The improvement is most obvious, when comparing the position solutions between this figure and Figures 6.14 to 6.16.

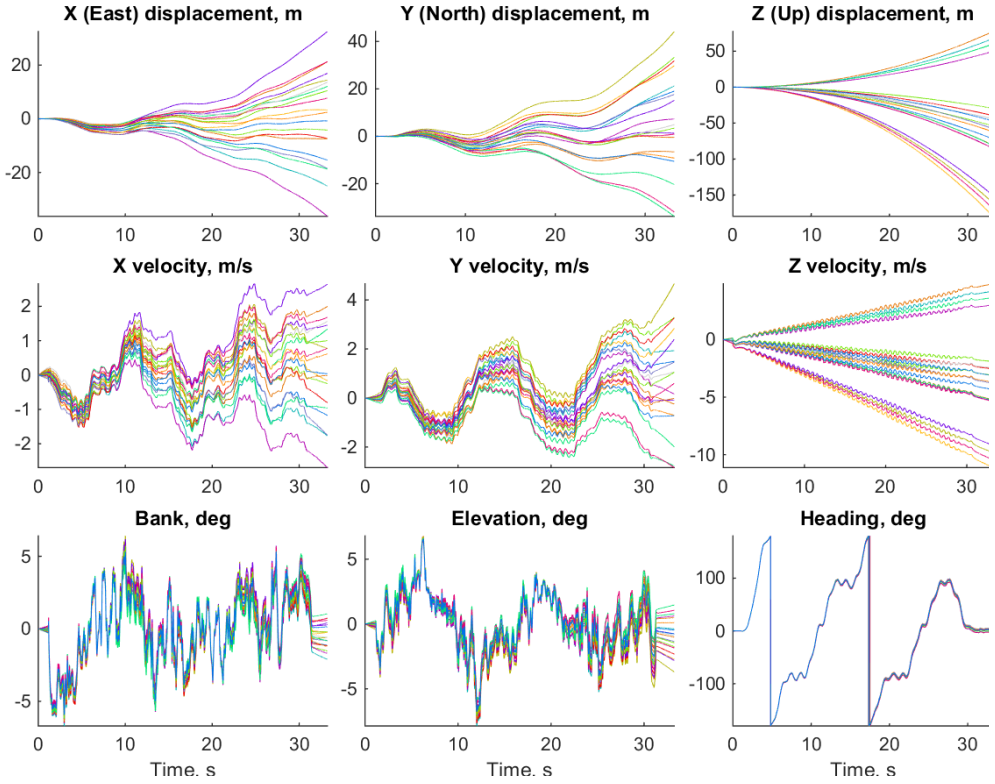
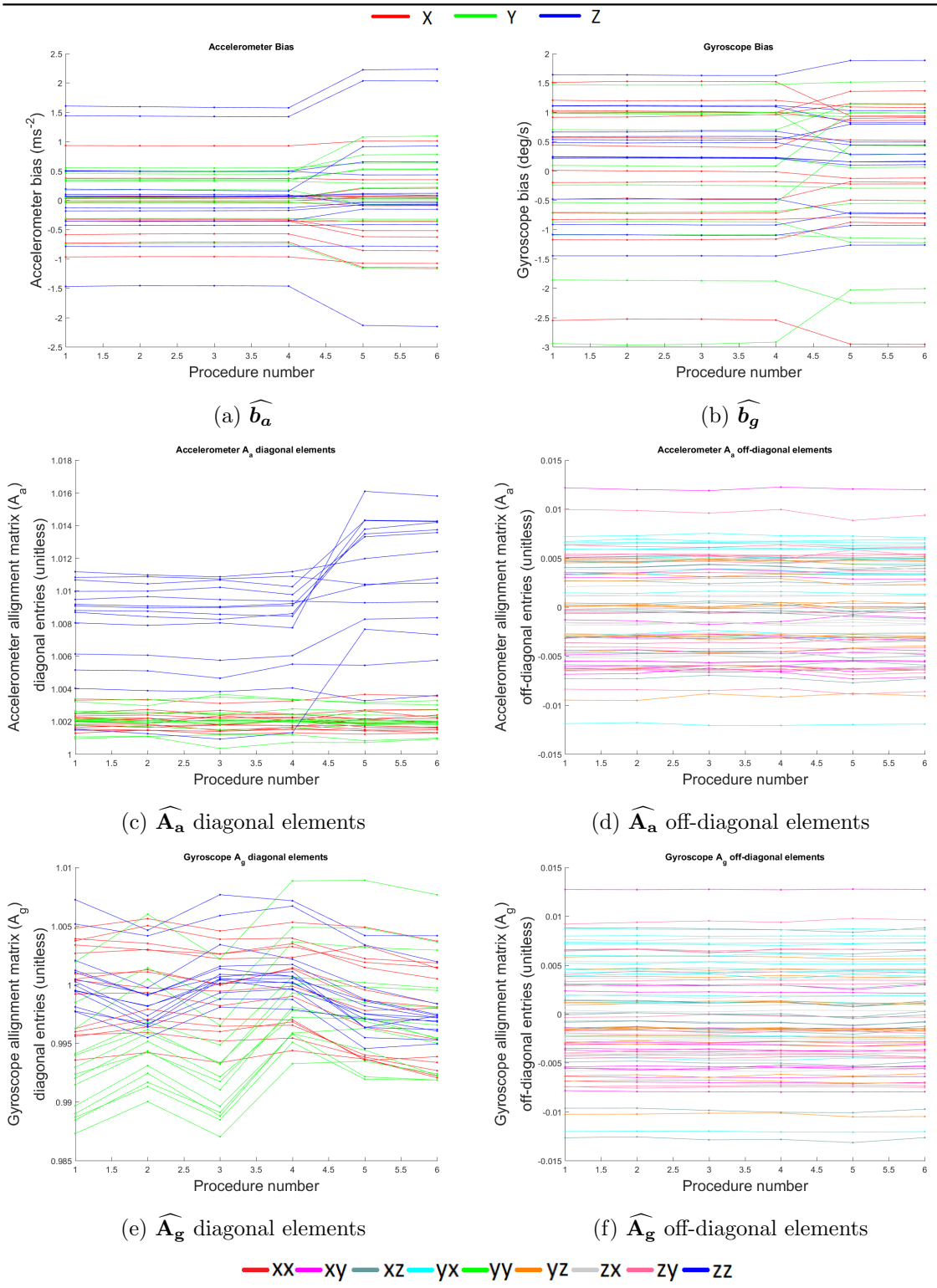


Figure 6.17: Motion profile obtained from a short pedestrian motion test by running inertial navigation on the mean each 4 IMU anti-symmetric set's corrected inertial data.

If the anti-parallel arrangement did not work then one would expect the aligned pairs (Figure 6.14) to perform equally well as both the xy-opposed (Figure 6.15) and horizontally-opposed (Figure 6.16) pairs. However, the aligned pairs appear to perform considerably worse. For example, the z-position spreads approximately 450m for the aligned pairs and about 300m for the xy-opposed pairs and 350m for the horizontally-opposed pairs.

Note that, by including all the possible pairs in Figures 6.14 to 6.17, it is clear this thesis is not cherry-picking the best performing pairs to present.

The improvement from the opposed pairs is due to part of the temperature related drift being cancelled. This can be seen in Figure 6.18 where the 2 post-heating calibration procedures (5 and 6) produce very different outputs in both bias and scale factor from the 4 pre-heating procedures, because, as can be seen in Figure 6.13, the internal temperature of the sensors was about 5°C higher. Note also that while the biases and scale factors (particularly the z-accelerometers) vary considerably between these two sets, the cross-



(a) $\widehat{\boldsymbol{b_a}}$ (b) $\widehat{\boldsymbol{b_g}}$

(c) $\widehat{\mathbf{A_a}}$ diagonal elements (d) $\widehat{\mathbf{A_a}}$ off-diagonal elements

(e) $\widehat{\mathbf{A_g}}$ diagonal elements (f) $\widehat{\mathbf{A_g}}$ off-diagonal elements

Figure 6.18: Comparison of the six procedures calibration estimate outputs: 1–4 pre-heating; 5–6 after heating, approximately 5°C hotter, see Figure 6.13.

coupling components do not significantly change. This implies, in conjunction with re apparent lack of variation day-to-day (Section 5.5), that an in-run estimate procedure could assume that the cross-coupling estimates from an older user calibration procedure at a different ambient temperature remained valid, and thus did not need to be estimated as a state.

6.7 Summary

This chapter presented several techniques for improving INS performance using an array of sensors. First, the baseline effect of averaging a simple array of n IMUs to create a single virtual IMU (VIMU) was described. In brief, this is a reduction in both noise and systematic error coefficient SD of $\sqrt{n}$. This chapter then describes three possible techniques, which could further improve the performance of an array of IMUs.

One technique is to orient the IMUs in an anti-parallel manner. This arrangement, illustrated in Figure 6.1, arranges the IMUs in the array such that all of their x- y- and z-axes are parallel, but half of the IMUs have their positive direction in each of the two possible parallel directions. To accomplish this requires at least 4 IMUs. For this idea to improve performance compared to the baseline of aligned IMUs the error coefficients much be distributed with a non-zero mean. The is evidence that this improves performance both in the literature (see Section 6.2) and from the experiments in Section 6.6.

Another aspect examined in this chapter is the difference between x- y- and z-axes in both performance of most models of low-cost IMUs and required performance for most of their applications. Optimisation of the IMU array to its requirements is examined, as well as ways to make an array of asymmetrically performing IMUs symmetrical. Evidence for the occurrence of these asymmetries is presented. The considerable differences between the x- y- and z-axes sensors in the MPU9250 IMUs in this thesis' hardware test-bed are also demonstrated.

The third technique examined in this chapter is IMU arrays with mixed measurement ranges. This technique assumes that lower-range sensors are more accurate. The idea is that part of an array of high-range sensors could be replaced with lower-range sensors, which could be used to measure the inertial forces more accurately during periods of lower-dynamics and then when the dynamics are higher, causing the low-range sensors to saturate, the higher-range sensors can be used alone. The trade-off is between an array of all high-range sensors, which has the same performance all of the time, and a mixed-range array which has better low-dynamics performance but slightly worse high-dynamics performance, due to the reduced number of high-range sensors. In many applications the system will be measuring low dynamics the majority of the time, so this technique will be a benefit. Experiments with the MPU9250 IMUs show no particular performance benefit to using the sensors at their lowest-range settings (250 °/s and 2g) as compared to their highest range setting (2000 °/s and 16g) thus this technique is not recommended for these IMUs.

Chapter 7

Conclusions and future work

In this thesis, calibration of sensors for inertial navigation has been discussed and new research presented in the following areas:

- A new method for assessing KF stability, which has been used to determine how well-calibrated an IMU must be for a basic 15-state INS/GNSS integration to remain stable.
- Several alternative methods that can be used to calibrate an IMU without requiring any equipment other than a flat surface with a 'corner'. These alternative methods have been analysed through statistical tests to determine which produces the most accurate calibration under specific circumstances.
- Three techniques for improving INS performance using IMU arrays, which produce better performance than simple averaging, have been presented and discussed. The circumstances under which they will improve performance have also been examined.
- An electronic hardware platform has been developed and experiments to support the ideas presented above have been carried out.

This chapter begins with the main conclusions of the thesis (Section 7.1). Then, the overall implications of and recommendations arising from this research are discussed in Section 7.2. Finally, recommendations for future research to build on the work presented in this thesis are made in Section 7.3.

7.1 Conclusions

The conclusions are divided into those related to the research on filter stability (Section 7.1.1), end-user calibration (Section 7.1.2) and on IMU arrays (Section 7.1.3).

7.1.1 Kalman filter stability

In Chapter 3, a new simulation approach to determine the limit of in-run sensor calibration using Kalman or Kalman-derivative filter has been presented. Filter stability criteria are defined based on the consistency of the state estimation errors and their uncertainties. This is tested with a basic 15-state Kalman filter INS/GNSS integration.

Monte Carlo simulations using a basic INS/GNSS Kalman filter to calibrate the sensor biases have been used to determine which IMU specifications are good enough for in-run

calibration. It is important to note that this technique is not specific to GNSS integration, another aiding technology or zero-velocity updates could be used instead.

Both car and UAV motion have been simulated. These tests have shown that within the ranges of errors encountered by real MEMS inertial sensors the most important error is the gyroscope bias. The maximum gyroscope bias standard deviation that is acceptable varies between 0.75 and 2.6 °/s (1 to 3.4 °/s for quadcopter) depending on the accelerometer noise level, gyroscope noise level and accelerometer bias SD, the last being the most important.

A major limitation on filter stability is the size of the higher-order errors, such as scale factor and cross coupling errors and gyroscope g-dependent errors, that are present in typical IMUs.

The basic KF integration will remain stable for a wide range of accelerometer and gyroscope biases when the accelerometer higher order errors are up to a medium level of 1% scale-factor and 0.5% cross coupling errors, and when the gyroscope errors are up to a low level of 0.3% scale factor, 0.2% cross coupling and 5 °/hour/g g-dependent error. Whereas the filter will only be stable for the greater levels of bias tested when there is a high level of 2% scale factor and 0.75% cross coupling for the accelerometer or a medium level of 1% scale factor, 1% cross coupling and 10 °/hour/g g-dependent error.

To maintain KF stability, these must be estimated as Kalman filter states, pre-calibrated and/or averaged out using an array technique.

It should be noted that, now that the stability limits of a standard KF are known, using a higher-performance KF-variant (such as a UKF [130]), or KF with more states at the start, and switching to the lower-computational load basic KF and once the state estimates are sufficiently accurate is a viable option.

7.1.2 User-conducted calibration

Another approach to improve the performance of the IMU is for the end-user to calibrate the IMU themselves. Chapter 4 presents possible series of manoeuvres which can be used to estimate the IMUs error parameters through a combination of IMU measurements recorded during static periods and dynamic manoeuvres. Simpler and more complex alternative versions of both static (either 6-position or 24-position) and dynamic (either 3-rotation or 6-rotation) manoeuvre sets were included. Importantly, these manoeuvres require no equipment other than an approximately level surface with a corner or straight edge down one side to calibrate the IMU in its box, which is assumed to be cubic or cuboid. Several alternative methods to calculate the calibration outputs from the IMU output recorded during these manoeuvres were also presented. The accuracy of these methods was assessed both through Monte Carlo simulations (Chapter 4) and through experiments with a bespoke IMU array (Chapter 5).

The Monte Carlo simulation determined how much the residual IMU errors depend on procedural error sources such as: how orthogonal sided the IMU's box is, how accurately the user replaces the IMU between manoeuvres, or how level the table is; or IMU error sources such as: how great the gyroscope noise PSD or the starting error in accelerometer bias is.

The conclusions from the KF stability part of this thesis gave minimum standards of IMU specification that needed to be met for a stable INS/GNSS integration. The simulations presented in Chapter 4 show that, the calibration procedure easily reaches and exceeds this standard for gyroscope bias calibration for a wide range of input parameters. They also show that the required standard for accelerometer alignment and scale factor matrix calibration can be reached when the IMU's box has a deviation from orthogonal sided of 1° SD or less. However, the standard that needs to be reached for the gyroscope scale factor and alignment matrix is more challenging. This can be reached with the 6-rotation manoeuvre set provided the cubes deviation from orthogonal sided is less than 0.25° SD. The required accuracy is not realistically possible with the 3-rotation manoeuvre set, although it can improve upon uncalibrated performance.

The simulations showed that the estimates calculated from the 24-position static manoeuvre set consistently outperformed those from the 6-position set. This is partly due to reducing the effects of sensor noise by averaging, but the more significant effect of this method is to virtually eliminate any effect of the table not being level from the results. Aside from this table-not-level error, which only significantly affected the 6-position set, the most significant procedural error was the box's deviation-from-orthogonal-sided. The effect of the user placing the cube at a slightly different heading than intended was less significant than might be expected. The IMU specifications with the most significant effect on the residual errors were the gyroscope noise PSD and the accelerometer non-linearity.

Two possible bias calculation techniques were tested, one was shown to be better for gyroscope bias and accelerometer bias when the box is very accurately made ($\leq 0.25°$ deviation-from-orthogonal SD), the other better for accelerometer bias with less accurate boxes. Two calculation techniques were also tested for the estimation of the gyroscope alignment and scale factor matrix, one of which was found to be better when the rotation axis was accurate ($\leq 2°$ error) but unstable for large axis errors, the other performed better with larger axis errors and with real data.

Experiments, conducted with an array of 14 Invensense MPU9250 IMUs, demonstrated that the calibration procedure produced repeatable estimates. By comparing simulation-derived and experimental repeatability, the simulation was shown to be a good representation of reality. Additionally, by integrating short periods of IMU measurements recorded while static or during pedestrian motion, the calibration was shown to work and the relative improvement that the more complex (24-position, 6-rotation) manoeuvre set makes over the simpler (6-position, 3-rotation) one was verified.

Experiments in calibrating the 14 IMUs on separate days several weeks apart, showed that the variation from day-to-day in the calibrated systematic IMU errors was significantly smaller than the initial distribution of errors between the sensors. This was particularly marked for both the accelerometer and gyroscope cross-coupling which were shown to remain virtually unchanged weeks later. In addition, these cross-coupling errors appeared unaffected by either temperature variation or range setting. This implies that a one-time calibration would be sufficient for accelerometer and gyroscope cross coupling, meaning that it might be possible to only estimate scale factors as KF states.

7.1.3 Sensor array techniques

The third approach to improving low-cost inertial navigation performance studied in this thesis is to use an array of multiple IMU sensors. This is a viable option because the cost of adding additional IMU chips to the navigation device's circuit board is low in comparison with that of factory calibration. Additionally the size and power consumption of MEMS IMUs are small in comparison with the other parts of a usable navigation device such as the CPU and the screen, so this is not a significant barrier. In this thesis, arrays of n IMUs are considered where each IMU contains an accelerometer triad and a gyroscope triad.

It is possible to arrange the IMUs in an array in such a manner that each sensitive axis of one IMU has the equivalent axis of another IMU parallel but pointing in the opposite direction. For example, the '+X' direction of one IMU is the '−X' direction of another. This can reduce the effect of biases, bias variation with temperature, and second-order errors on the navigation performance of the array, provided that these are distributed with a non-zero mean. Experiments characterising the performance of the MPU9250 IMUs in the array lead to the conclusion that this arrangement will improve performance. This is demonstrated with a test where an IMU is calibrated then heated up and around 30 seconds of pedestrian motion were recorded. The position solution was shown to drift by around 350m for parallel arrays and 250m for anti-parallel arrays.

Another possibility explored is that of having an array of sensors with mixture of different measurement ranges. A weighted combination is made of lower- and higher-range IMUs, assuming that the lower-range sensors will be more accurate. The idea is that when the specific force or angular rate is low, assumed to be the majority of the time, the lower-range sensors provide a more accurate signal, but when the dynamics are higher the higher-range sensors are used alone. Experiments characterising the performance of the 2-to-16g and 250-to-2000 °/s MPU9250 IMUs show that the range setting does not significantly change for either systematic or stochastic error performance. Thus, this idea will not work for the MPU9250.

The final topic explored in Chapter 6 is asymmetry in IMU performance. In many applications, the INS performance is not required to be the same in all directions, and the IMUs often have different error performance on their different axes. This asymmetric IMU performance can be optimised for asymmetric requirements. Experiments characterising the MPU9250 IMUs show considerable performance differences between the three axes, not only between the out-of-plane z-axis and the in-plane x and y axes, but also between x and y.

7.2 Implications of the research

The results of the work on KF stability will enable navigation system designers to determine for which IMUs specification levels it is possible to use a basic 15-state KF for sensor integration. Thus, only when outside of these levels is a more complex and/or non-linear Kalman filter variant and/or sensor pre-calibration required. The research into end-user-calibration will enable the optimal estimation algorithm to be applied given the IMUs specifications

and the construction accuracy of the INS's housing. Better navigation performance can be obtained by averaging the output of an array of IMUs, the work on array techniques in this thesis will allow users to optimise the performance of an array of a given size, given the performance characteristics of the particular model of sensor or the complementary properties of different sensor models.

Better low-cost INS performance will reduce the cost of systems that currently require factory-calibrated IMUs for their operation. It will also make integrated INS/GNSS available in applications where it is currently not economic such as smartphones. This makes the much better short-term navigation performance of INS/GNSS compared to GNSS alone much more widely available. Also if the INS can be calibrated when GNSS is available, the ability of the navigation system to bridge signal outages will be markedly improved. These improvements will be particularly useful for low-cost UAVs, emergency response teams and military applications in urban areas.

This thesis makes the following specific recommendations:

- Gyroscope bias has been shown to be more important for filter stability than accelerometer bias. Thus, the user should concentrate on reducing the gyroscope bias before use. This could be by, for example, using a zero angular rate update.
- The main factor that makes low-cost sensors unsuitable for basic KF INS/GNSS integration is their levels of first-order errors, such as scale factor and cross coupling errors, which are too high to be neglected by the integration filter. Thus, the recommendation is that these errors be pre-calibrated, and/or they or a subset of them be modelled as KF states.
- The 24-position static calibration (C) and the 6-rotation (D) manoeuvre sets are recommended for user-calibration, for their considerably better calibration performance.
- The optimal algorithm for calculating the relative magnitudes of the various factors affecting calibration accuracy. The recommendation is that a sample of mass-produced devices have their housings measured and their noise characterised to pick the optimal algorithm.
- Some of the user-calibration procedure algorithms can produce poor estimates in the case of poorly conducted or mis-identified manoeuvres. Thus, before use in production devices a more robust error-checking algorithm is needed.
- Orientating the sensors in an IMU array in an anti-parallel manner is recommended as it has been shown to improve the performance of the MPU9250 IMUs tested.
- The MPU9250 sensors performance is very similar at different measurement range settings. Thus, to avoid clipping, it is recommended that these IMUs be used at their maximum (16g and 2000 °/s) range settings.

7.3 Future work

As the research presented in this thesis splits into three topics the future work for each of these will be presented separately.

7.3.1 Integration Filter stability

The Kalman filter stability simulation method developed in this thesis (Chapter 3) could be extended for further research on more sophisticated Kalman filters or Kalman filter variants. For example, the basic KF could be replaced with a more advanced algorithm such as a Unscented Kalman filter [130]. This kind of filter might reasonably be expected to be more stable, at least for higher levels of starting bias, whether it would also be stable for higher levels of unestimated first-order errors is not clear.

This thesis shows that any important limitation for stable 15-state Kalman filter integration is the error sources not estimated, i.e. scale factor, cross coupling and gyroscope g-dependant errors, exist at a magnitude in low-cost MEMS IMUs that significantly de-stabilises the basic KF integration tested in this thesis. One solution to this is to model these states or a subset of them in the KF integration. This would require a more complex IMU model, such as that used for the research presented in Chapter 4. The gyroscope scale factor and cross coupling matrix is the error source most difficult to calibrate with sufficient accuracy using the user calibration techniques of Chapter 4, it should be added to the 15-state KF first. This could be tested as 3-states (scale factors modelled only), the full 9 states, or a 6-state diagonalised representation of the gyroscope scale factor and cross coupling matrix. The 3 additional state scale-factor-only option is a sensible one to test because this thesis has shown that the scale factor varies more, with temperature and day-to-day, than the cross-coupling. The other un-estimated states of gyroscope g-dependent error and accelerometer scale factor and cross coupling could also be modelled.

Another possible area for future research are different filter stability criteria. More and less stringent versions of the criteria presented here could be tested, as well as alternative methods of assessment, most of which would only require modification of one or two functions of the MATLAB code.

GNSS-aiding was only the example used to test the technique for examining KF stability developed in this thesis. This technique to examine filter stability could be quickly adapted to another aiding technique. There is potential to examine the integration stability with other positioning technologies which have different update rates and different position and velocity measurement accuracies, or to examine the effect of including additional sensors with the IMU, potentially magnetometers would have a large impact because the heading solution is the hardest to observe with GNSS. KF-variant filters could also be used to calibrate the IMU with stationarity detection for ZVUs and ZARUs, and either a defined (as in Chapter 4) or an arbitrary set of gestures. Different motion profiles could also be simulated potentially including pedestrian navigation or more niche highly dynamic ones.

7.3.2 User calibration

There are several ways in which the research presented on user calibration (Chapter 4 and 5) should be extended in the future.

The main topic that should be investigated is determining a more reliable automatic detection and compensation for incorrect (or mis-identified) manoeuvres and/or bad error

estimates caused by unstable iterative processes. In the current code, bad parameter estimates are identified by inspection of automatically generated plots, and the correction done manually. In a production system, this could be done by comparison with previous estimates, or comparison with specified performance. However, as bad data frequently results in highly incorrect estimates e.g. scale factors of ≤ 0.5 or > 2, then very simple thresholds may be sufficient.

Additional static estimation manoeuvre sets should be tested. As discussed in Section 4.5.3, the six-position method is not actually the minimum number of positions. The static testing simulation could be extended to compare the minimal positions. These are 1-position (gyroscope bias), 2-position (accelerometer and gyroscope bias) and 4-position (accelerometer alignment and scale factor and gyroscope g-dependent error). Additionally methods which might be expected to have performance between that of the 6-position and 24-position methods could be tested. Particularly, a 12-position method, using two positions on each face at 180° heading from one another.

There are also alternative algorithms and manoeuvres that could be used for the dynamic estimation. This thesis has shown there are stability issues with the iterative process used for estimating the gyroscope scale factor and cross coupling particularly for the sum method when the axis-errors are larger. Aside from the comparison with specifications or thresholds suggested above, there are other possible approaches. This includes integrating the gyroscope output of other parts of the data, e.g. moving between static faces in manoeuvre set A or C and seeing if the apparent error gets smaller or larger with each correction in the iterative process, quitting the iteration when it grows. One could also test different algorithms for comparison with the sum and integrated methods. One option is separating the estimation of the diagonal and off-diagonal elements, using the ratios of the different angular rates for the off-diagonal elements and then the magnitude of the integrated change in position for correcting the diagonal elements.

There are also other possible dynamic manoeuvres that could be used to estimate gyroscope scale factor and alignment which should be tested. One is to use the four 90° rotations on each face during set C as an alternative to the one 360° rotation of set D. This has the advantage that the rotations are being done anyway, and while the percentage effect of each heading replacement errors is larger, the fact that there are 8 heading replacements per side rather than 2 alleviates this somewhat through averaging. Also for a real system, a single incident of sensor clipping would only make one rotation invalid leaving 270° worth, rather than invalidating the reading for that entire face.

A thorough assessment of the accuracy of other user calibration techniques suggested in the literature should be made through Monte Carlo simulation, so that their results can be compared with those of the procedure presented in this thesis under equivalent circumstances. This might include the technique, presented in [154, 161], which ignores the fact that the box's faces are orthogonal and uses a least squares estimation to calibrate the accelerometers. Another which should be assessed is that used in [159], to determine its sensitivity to procedural and IMU errors.

Many of the ideas of averaging IMU error sources across an array of sensors in Chapter

6 make the assumption that the error on each sensor is independently distributed. While this is a reasonable assumption for uncalibrated sensors, the extent to which it is for post-calibration residual errors, when all the sensors in the array are calibrated simultaneously using the same set of manoeuvres in the same box, should be tested.

7.3.3 Sensor arrays

The research presented in Chapter 6 should be extended in the future.

The limited resources of this project has resulted in only characterising the performance of one model of consumer-grade IMU, the MPU9250, where some of the ideas have been shown to be viable. Assessing the performance of different IMU models may reveal non-zero mean error distributions for other IMU models which can be exploited by the array techniques suggested in this thesis. To properly test these ideas, new bespoke test-beds would need to be designed and manufactured, but the operating principle could be the same.

The multiple measurement-range-array techniques would be more promising in instances where much higher dynamics are expected, so future research in this idea should concentrate on these areas. However, even for applications where one might not expect high inertial forces, such as pedestrian navigation, it is possible to saturate the sensors, particularly the gyroscopes. This means that including even a single IMU with much higher measurement range (e.g. 100g and 10,000 °/s) would make clipping very unlikely and provide a low-accuracy signal to bridge the short periods of time when clipping would otherwise occur. In order to assess their viability, sensors with both higher and lower maximum measurement ranges than the 16g and 2000 °/s of the MPU9250s tested in this thesis should be tested.

Additionally, it would be useful to assess the linearity of the sensors across a range of outputs. This would require a centrifuge and/or rate-table. This would be make it possible to assess whether the anti-parallel arrays would reduce the error due to 2^{nd}-order non-linearity terms, by seeing if the quadratic terms were distributed with a non-zero mean across the sensors.

It would also be desirable to perform heat-chamber testing on the consumer-grade IMUs to more accurately assess their errors response to variation in temperature. The results presented in this thesis are only indicative of the inertial sensors bias behaviour with temperature. A proper laboratory thermal calibration would enable, state of the art temperature compensation to be compared with the approximate removal of temperature effects made by the array techniques, in addition to comparing it with no compensation.

References

[1] P. D. Groves, *Principles of GNSS, inertial, and multi-sensor integrated navigation systems*. Artech House, 2 ed., 2013.

[2] U. I. Bhatti, W. Y. Ochieng, and S. Feng, "Integrity of an integrated GPS/INS system in the presence of slowly growing errors. part I: A critical review," *GPS Solutions*, vol. 11, no. 3, pp. 173–181, 2006.

[3] M. J. Veth, "Navigation using images, a survey of techniques," *Navigation*, vol. 58, no. 2, pp. 127–139, 2011.

[4] D. Titterton and J. Weston, *Strapdown Inertial Navigation Technology*. IEE Radar Series, Institution of Electrical Engineers, 2004.

[5] IEEE, *Standard Specification Format Guide and Test Procedure for Single Axis Interferometric Fiber Optic Gyros*, 1998. IEEE Std 952-1997 Annex C.

[6] H. Martin, P. Groves, and M. Newman, "The limits of in-run calibration of MEMS inertial sensors and sensor arrays," *Navigation*, vol. 63, no. 2, pp. 127–143, 2016.

[7] H. F. S. Martin, P. D. Groves, and M. Newman, "The limits of in-run calibration of MEMS and the effect of new techniques," in *Proc. ION GNSS*, September 2014.

[8] H. F. S. Martin, P. D. Groves, M. Newman, and R. Faragher, "A new approach to better low-cost MEMS IMU performance using sensor arrays," in *Proc. ION GNSS*, September 2013.

[9] P. D. Groves, L. Wang, D. Walter, H. F. S. Martin, K. Voutsis, and Z. Jiang, "The four key challenges of advanced multisensor navigation and positioning," in *IEEE/ION PLANS*, (Monterey, CA), May 2014.

[10] P. D. Groves, H. Martin, K. Voutsis, D. Walter, and L. Wang, "Context detection, categorization and connectivity for advanced adaptive integrated navigation," in *Proc. ION GNSS+*, (Nashville, Tennessee), 2013.

[11] P. D. Groves, L. Wang, D. Walter, H. Martin, and K. Voutsis, "Toward a unified PNT, part 1: Complexity and context: Key challenges of multisensor positioning," *GPS World*, vol. 25, no. 10, 2014.

[12] J. Farrell, *The Global Positioning System & Inertial Navigation*. McGraw-Hill Education, 1999.

[13] Y. Bar-Shalom, X. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software*. Wiley, 2004.

[14] H. Goldstein, C. Poole, and J. Safko, *Classical Mechanics*. Addison Wesley, 2002.

[15] I. C. A. Organization, *International Standards and Recommended Practices, Annex 10, Radio and Navigation Aids*. Montreal, 2006. Vol. 1.

[16] S. Feng, A. Jokinen, W. Ochieng, J. Liu, and Q. Zeng, *China Satellite Navigation Conference (CSNC) 2014 Proceedings: Volume II*, ch. Receiver Autonomous Integrity Monitoring for Fixed Ambiguity Precise Point Positioning, pp. 159–169. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014.

[17] R. L. Morris and R. J. Palmer, "A short range, high precision navigation system for automated farming," in *Proceedings of the 50th Annual Meeting of The Institute of Navigation (1994)*, (Colorado Springs, CO), pp. 495–500, June 1994.

[18] J. Farrell, *Aided Navigation: GPS with High Rate Sensors*. McGraw-Hill professional engineering: Electronic engineering, McGraw-Hill Education, 2008.

[19] P. Aggarwal, Z. Syed, A. Noureldin, and N. El-Sheimy, *MEMS-based Integrated Navigation*. Artech House, 2010.

[20] V. Kempe, *Inertial MEMS: Principles and Practice*. Cambridge University Press, 2011.

[21] N. El-Sheimy, "The potential of partial IMUs for land vehicle navigation," in *Inside GNSS*, pp. 16–25, Gibbons Media & Research LLC, spring 2008.

[22] D. Sun, M. G. Petovello, and M. E. Cannon, "Use of a reduced IMU to aid a GPS receiver with adaptive tracking loops for land vehicle navigation," *GPS Solutions*, vol. 14, no. 4, pp. 319–329, 2010.

[23] InvenSense Inc., 1745 Technology Drive, San Jose, CA 95110 U.S.A., *MPU-9250 Product Specification*, January 2014. Document Number: PS-MPU-9250A-01, Revision: 1.0.

[24] STMicroelectronics, *L3G4200D MEMS motion sensor: ultra-stable three-axis digital output gyroscope datasheet*, December 2010. Doc ID 17116 Rev. 3.

[25] InvenSense Inc., 1197 Borregas Ave, Sunnyvale, CA 94089, *MPU-9150 Product Specification*, May 2012. PS-MPU-9150A-00, Revision 4.0.

[26] "Product page: VN-100 Rugged." online. `https://www.vectornav.com/purchase/product/vn-100-rugged` accessed 09/03/16.

[27] M. Lemkin and B. E. Boser, "A three-axis micromachined accelerometer with a CMOS position-sense interface and digital offset-trim electronics," *IEEE Journal of Solid-State Circuits*, vol. 34, pp. 456–468, Apr 1999.

[28] R. Amarasinghe, D. V. Dao, T. Toriyama, and S. Sugiyama, "Simulation, fabrication and characterization of a three-axis piezoresistive accelerometer," *Smart Materials and Structures*, vol. 15, no. 6, p. 1691, 2006.

[29] P. G. Savage, "What do accelerometers measure?." online, May 2005. http://www.strapdownassociates.com/Accels%20Measure.pdf, accessed 07/05/15.

[30] M. Liu, B. Chi, Y. Liu, and J. Dong, "A closed-loop MEMS accelerometer with capacitive sensing interface ASIC," *International Journal of Electronics*, vol. 100, no. 1, pp. 21–35, 2013.

[31] J. Wilson, *Sensor Technology Handbook*. Electronics & Electrical, Elsevier Science, 2004.

[32] J.-M. Stauffer, "Microsensors take off," in *Aerospace Testing Int. Showcase 2011*, pp. 14–16, UKIP Media & Events Ltd, 2011.

[33] J. Li, J. Fang, M. Du, and H. Dong, "Analysis and fabrication of a novel MEMS pendulum angular accelerometer with electrostatic actuator feedback," *Microsystem Technologies*, vol. 19, no. 1, pp. 9–16, 2013.

[34] ST Microtronics, *LIS1R02 (L6671) Angular Accelerometer Product Preview*, April 2002.

[35] D. Tuite, "Motion-sensing mems gyros and accelerometers are everywhere." online, July 2009. http://electronicdesign.com/analog/motion-sensing-mems-gyros-and-accelerometers-are-everywhere accessed 12/5/16.

[36] W. M. Macek and D. T. M. Davis, Jr., "Rotation Rate Sensing with Traveling-Wave Ring Lasers," *Applied Physics Letters*, vol. 2, pp. 67–68, Feb. 1963.

[37] V. Vali and R. W. Shorthill, "Fiber ring interferometer," *Appl. Opt.*, vol. 15, pp. 1099–1100, May 1976.

[38] E. J. Loper and D. D. Lynch, "Hemispherical resonator gyro: Status report and test results," in *Proceedings of the 1984 National Technical Meeting of The Institute of Navigation*, (San Diego, CA), pp. 105–107, January 1984.

[39] C. T. Morrow, "Zero signals in the sperry tuning fork gyrotron," *The Journal of the Acoustical Society of America*, vol. 27, no. 3, pp. 581–585, 1955.

[40] J. Bernstein, S. Cho, A. King, A. Kourepenis, P. Maciel, and M. Weinberg, "A micromachined comb-drive tuning fork rate gyroscope," in *Micro Electro Mechanical Systems, 1993, MEMS'93, Proceedings An Investigation of Micro Structures, Sensors, Actuators, Machines and Systems. IEEE.*, pp. 143–148, IEEE, 1993.

[41] M. Moore, C. Rizos, J. Wang, G. Boyd, K. Matthews, W. Williams, and R. Smith, "A GPS based attitude determination system for a uav aided by low grade angular rate gyros," in *Proceedings of the 16th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS/GNSS 2003)*, (Portland, OR), pp. 2417–2424, September 2003.

[42] J. Richeson, *Gravity Gradiometer Aided Inertial Navigation Within Non-GNSS Environments*. University of Maryland, College Park, 2008.

[43] S. Guerrier, R. Molinari, and J. Skaloud, "Automatic identification and calibration of stochastic parameters in inertial sensors," *Navigation*, vol. 62, no. 4, pp. 265–272, 2015.

[44] M. Pittelkau, "Attitude determination kalman filter with a 1/f flicker noise gyro model," in *Proceedings of the 26th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2013)*, (Nashville, TN), pp. 2143–2160, September 2013.

[45] Bosch Sensortec, *BMI055 Data sheet*, May 2013. BST-BMI055-DS000-06.

[46] STMicroelectronics, *LSM9DS0 iNEMO inertial module: 3D accelerometer, 3D gyroscope, 3D magnetometer Datasheet*, August 2013. DocID: 024763 Rev 2.

[47] O. Le Traon *et al.*, "The VIA vibrating beam accelerometer: Concept and performances," in *Proc. IEEE/ION PLANS*, (Palm Springs, CA), pp. 25–37, 1998.

[48] STMicrotronics, *Everything about STMicroelectronics' 3-axis digital MEMS gyroscopes*, July 2011. Doc ID 022032 Rev 1.

[49] Analog Devices, One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A., *ADXL345 3-Axis Digital Accelerometer Datasheet*, 2009-2011. Rev C.

[50] Analog Devices, One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A., *ADXL375 3-Axis Digital Accelerometer Datasheet*, 20132014. Rev B.

[51] Bosch Sensortec, *SMI130 Product Information sheet*, August 2014.

[52] Bosch Sensortec, *BMA180 Digital, triaxial acceleration sensor*, December 2010. BST-BMA180-DS000-07.

[53] InvenSense Inc., Sunnyvale, CA, *ITG-3200 Product Specification*, March 2010. Revision 1.4, Document Number: PS-ITG-3200A-00-01.4.

[54] Analog Devices, One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A., *ADIS16367 Six Degrees of Freedom Inertial Sensor Datasheet*, 20102012. Rev B.

[55] Y. Yuksel, N. El-Sheimy, and A. Noureldin, "Error modeling and characterization of environmental effects for low cost inertial MEMS units," in *Position Location and Navigation Symposium (PLANS), 2010 IEEE/ION*, pp. 598 –612, May 2010.

[56] M. Becker, U. Bestmann, A. Schwithal, P. Hecker, R. Bott, P. Kohl, S. Knedlik, H. Scheyer, and E. v. Hinber, "Evaluation of a double equipped MEMS IMU based on real flight trial scenarios," in *Position Location and Navigation Symposium (PLANS), 2010 IEEE/ION*, pp. 202–213, May 2010.

[57] Y. Dong, P. Zwahlen, A. M. Nguyen, F. Rudolf, and J. M. Stauffer, "High performance inertial navigation grade sigma-delta MEMS accelerometer," in *Position Location and Navigation Symposium (PLANS), 2010 IEEE/ION*, pp. 32–36, May 2010.

[58] VectorNav Technologies, *VN-300 User Manual: Inertial Navigation Modules*, 2014. UM005 v0.14.

[59] J. Bird and D. Arden, "Indoor navigation with foot-mounted strapdown inertial navigation and magnetic sensors [emerging opportunities for localization and tracking]," *IEEE Wireless Communications*, vol. 18, pp. 28–35, April 2011.

[60] P. D. Groves *et al.*, "Inertial navigation versus pedestrian dead reckoning: Optimizing the integration," in *Proc. ION GNSS*, 2007.

[61] International Loran Association, *Enhanced Loran (eLoran) Definition Document*, January 2007. Report Version: 0.1.

[62] United States Coast Guard, *The LORAN-C User's Handbook, COMDTPUB P16562.6*, 1990.

[63] L. Wang, P. D. Groves, and M. K. Ziebart, "Multi-constellation GNSS performance evaluation for urban canyons using large virtual reality city models," *Journal of Navigation*, vol. 65, no. 03, pp. 459–476, 2012.

[64] P. Misra and P. Enge, *Global positioning system: signals, measurements, and performance*. Ganga-Jamuna Press, 2006.

[65] A. G. Proctor, C. W. Curry, J. Tong, R. Watson, M. Greaves, and P. Cruddace, "Protecting the UK infrastructure - a system to detect GNSS jamming and interference," *inside GNSS, vol. September/October*, pp. 49–57, 2011.

[66] H. Kuusniemi, E. Airos, M. Z. H. Bhuiyan, and T. Kröger, "GNSS jammers: how vulnerable are consumer grade satellite navigation receivers?," *European Journal of Navigation*, vol. 10, no. 2, pp. 14–21, 2012.

[67] T. E. Humphreys, B. M. Ledvina, M. L. Psiaki, B. W. OHanlon, and P. M. Kintner Jr, "Assessing the spoofing threat: Development of a portable GPS civilian spoofer," *Proceedings of the ION GNSS international technical meeting of the satellite division*, vol. 55, p. 56, 2008.

[68] S. Feng and A. Jokinen, "Integer ambiguity validation in high accuracy GNSS positioning," *GPS Solutions*, pp. 1–9, 2015.

[69] A. Leick and S. Lambert, *GPS Satellite Surveying*. Wiley-Interscience publication, Wiley, 1990.

[70] G. Wübbena, M. Schmitz, and A. Bagge, "PPP-RTK: precise point positioning using state-space representation in RTK networks," in *Proceedings of ION GNSS*, vol. 5, pp. 13–16, 2005.

[71] A. Jokinen, S. Feng, W. Schuster, W. Ochieng, C. Hide, T. Moore, and C. Hill, "GLONASS aided GPS ambiguity fixed precise point positioning," *Journal of Navigation*, vol. 66, pp. 399–416, 2013.

[72] S. Hong, J. Y. Choi, C.-S. Kim, M. H. Lee, and J. Speyer, "Estimation of errors in INS with multiple GPS antennas," in *Industrial Electronics Society, 2001. IECON '01. The 27th Annual Conference of the IEEE*, vol. 1, pp. 410–415 vol.1, 2001.

[73] P. Y. Montgomery, T. E. Humphreys, and B. M. Ledvina, "Receiver-autonomous spoofing detection: Experimental results of a multi-antenna receiver defense against a portable civil GPS spoofer," in *Proceedings of the 2009 International Technical Meeting of The Institute of Navigation*, (Anaheim, CA), pp. 124–130, January 2009.

[74] P. D. Groves, C. J. Mather, and A. A. Macaulay, "Demonstration of non-coherent deep INS/GPS integration for optimized signal to noise performance," in *Proc. ION GNSS*, 2007.

[75] X. Wang, X. Ji, and S. Feng, "A scheme for weak GPS signal acquisition aided by SINS information," *GPS Solutions*, vol. 18, no. 2, pp. 243–252, 2013.

[76] M. Caruso, "Applications of magnetic sensors for low cost compass systems," in *Position Location and Navigation Symposium, IEEE 2000*, pp. 177 –184, 2000.

[77] Honeywell, 12001 Highway 55, Plymouth, MN 55441, *3-Axis Digital Compass IC HMC5883 datasheet*, March 2010. Form 900405 Rev A.

[78] STMicroelectronics, *LSM303DLHC Ultra compact high performance e-compass 3D accelerometer and 3D magnetometer module*, April 2011. DocID: 018771 Rev 1.

[79] F. Zampella, M. Khider, P. Robertson, and A. Jimenez, "Unscented Kalman filter and magnetic angular rate update (MARU) for an improved pedestrian dead-reckoning," in *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION*, pp. 129–139, April.

[80] R. Faragher, C. Sarno, and M. Newman, "Opportunistic radio slam for indoor navigation using smartphone sensors," in *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION*, pp. 120 –128, April 2012.

[81] "Suunto ambit 3 vertical product page." online. `http://www.suunto.com/en-GB/Products/sports-watches/Suunto-Ambit3-Vertical/Suunto-Ambit3-Vertical-Black/`, accessed 08/05/16.

[82] M. B. Mathews, P. F. Macdoran, and K. L. Gold, "SCP enabled navigation using signals of opportunity in GPS obstructed environments," *Navigation*, vol. 58, pp. 91–110, Summer 2011.

[83] Z. Shen, M. Korenberg, and A. Noureldin, "A real-time approach to nonlinear modelling of inertial errors with application to three dimension vehicle navigation," in *Proceedings of the 2011 International Technical Meeting of The Institute of Navigation*, (San Diego, CA), pp. 811–818, January 2011.

[84] D. Gustafson, J. Dowdle, and K. Flueckiger, "A deeply integrated adaptive GPS-based navigator with extended range code tracking," in *Position Location and Navigation Symposium, IEEE 2000*, pp. 118 –124, 2000.

[85] Y.-K. Kim, S.-H. Choi, H.-W. Kim, and J.-M. Lee, "Performance improvement and height estimation of pedestrian dead-reckoning system using a low cost MEMS sensor," in *Control, Automation and Systems (ICCAS), 2012 12th International Conference on*, pp. 1655–1660, Oct.

[86] M. Chowdhary, M. Sharma, A. Kumar, K. Paul, M. Jain, C. Agarwal, and G. Narula, "Context detection for improving positioning performance and enhancing user experience," in *Proceedings of the 22nd International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2009)*, (Savannah, GA), pp. 2072–2076, September 2009.

[87] S. Saeedi, N. El-Sheimy, X. Zhao, and Z. Sayed, "Context aware mobile personal navigation services using multi-level sensor fusion," in *Proceedings of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2011)*, (Portland, OR), pp. 1394–1403, September 2011.

[88] M. J. Veth, R. K. Martin, and M. Pachter, "Anti-temporal-aliasing constraints for image-based feature tracking applications with and without inertial aiding," *IEEE Transactions on Vehicular Technology*, vol. 59, pp. 3744–3756, Oct 2010.

[89] C. Hide, T. Boterill, and M. Andreotti, "An intergrated IMU, GNSS and image recognition sensor for pedestrian navigation," in *Proc. ION GNSS 2009*, (Savannah, GA), pp. 527–537, September 2009.

[90] G. Nützi, S. Weiss, D. Scaramuzza, and R. Siegwart, "Fusion of imu and vision for absolute scale estimation in monocular slam," *Journal of intelligent & robotic systems*, vol. 61, no. 1-4, pp. 287–299, 2011.

[91] O. Kupervasser, R. Lerner, and E. Rivlin, "Error analysis for a navigation algorithm based on optical-flow and a digital terrain map," in *Proceedings of IEEE/ION PLANS 2008*, (Monterey, CA), pp. 1203–1212, May 2008.

[92] B. Hartmann, N. Link, and G. Trommer, "Indoor 3D position estimation using low-cost inertial sensors and marker-based video-tracking," in *Proceedings of IEEE/ION PLANS 2010*, (Indian Wells, CA), pp. 319–326, May 2010.

[93] D. Walter, P. D. Groves, B. Mason, J. Harrison, J. Woodward, and P. Wright, "Road navigation using multiple dissimilar environmental features to bridge GNSS outages," in *Proceedings of the 28th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2015)*, pp. 2098–2114, 2015.

[94] J. Johns, *U.S. National Aviation Handbook for the VOR/DME/TACAN Systems*, September 1982. N0. 9840.1.

[95] T. A. Webb, *Differential Positioning using Signals of Opportunity*. PhD thesis, UCL (University College London), 2013.

[96] R. Faragher, "Spoofing mitigation, robust collision avoidance, and opportunistic receiver localisation using a new signal processing scheme for ADS-B or AIS," in *Proceedings of the 27th International Technical Meeting of The Satellite Division of the Insti-*

tute of Navigation (ION GNSS+ 2014), Tampa, Florida, September 2014, pp. 858–868, 2014.

[97] R. Faragher and R. Harle, "Location fingerprinting with bluetooth low energy beacons," *IEEE Journal on Selected Areas in Communications*, vol. 33, pp. 2418–2428, Nov 2015.

[98] J. B. Bullock, D. Chowdhary, Mahesh ANDRubin, D. Leimer, G. Turetzky, and M. Jarvis, "Continuous indoor positioning using GNSS, Wi-Fi, and MEMS dead reckoning," in *Proceedings of the 25th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2012)*, (Nashville, TN), pp. 2408–2416, September 2012.

[99] L. Ojeda and J. Borenstein, "Personal dead-reckoning system for GPS-denied environments," in *Safety, Security and Rescue Robotics, 2007. SSRR 2007. IEEE International Workshop on*, pp. 1–6, September.

[100] S. Y. Cho and C. G. Park, "MEMS based pedestrian navigation system," *Journal of Navigation*, vol. 59, pp. 135–153, 0 2006.

[101] J. O. Nilsson, I. Skog, and P. Hndel, "Performance characterisation of foot-mounted zupt-aided inss and other related systems," in *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*, pp. 1–7, Sept 2010.

[102] D. Grinberg, "*Nike+iPod reverse engineering (protocol too).*" Online. `http://dmitry.co/index.php?p=./04.Thoughts/05.%20Nike%20plus%20iPod` , accessed 14/11/12.

[103] N. Seidle, "*Nike+iPod Dissection.*" online, January 2007. `http://www.sparkfun.com/tutorials/41` , accessed 14/11/12.

[104] P. Robertson, M. G. Puyol, and M. Angermann, "Collaborative pedestrian mapping of buildings using inertial sensors and FootSLAM," in *Proceedings of the 24th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2011)*, (Portland, OR), pp. 1366–1377, September 2011.

[105] O. Woodman and R. Harle, "Pedestrian localisation for indoor environments," in *Proceedings of the 10th international conference on Ubiquitous computing*, pp. 114–123, ACM, 2008.

[106] Y. Zhao, *Vehicle Location and Navigation Systems.* Artech House intelligent transportation systems library, Artech House, 1997.

[107] M. A. Quddus, W. Y. Ochieng, and R. B. Noland, "Current map-matching algorithms for transport applications: State-of-the art and future research directions," *Transportation Research Part C: Emerging Technologies*, vol. 15, no. 5, pp. 312–328, 2007.

[108] D. Walter, P. Groves, B. Mason, J. Harrison, J. Woodward, and P. Wright, "Novel environmental features for robust multisensor navigation," in *Proceedings of the 26th International Technical Meeting of The Satellite Division of the Institute of Navigation*, (Nashville, TN), pp. 488–504, September 2013.

[109] P. Groves, L. Wang, and M. Ziebart, "Shadow matching: Improved GNSS accuracy in urban canyons," *GPS World*, vol. 23, no. 2, pp. 14 – 18+27, 2012.

[110] P. D. Groves, "Shadow matching: A new GNSS positioning technique for urban canyons," *The Journal of Navigation*, vol. 64, no. 03, pp. 417–430, 2011.

[111] A. Bourdeau, M. Sahmoudi, and J.-Y. Tourneret, "Constructive use of GNSS NLOS-multipath: Augmenting the navigation Kalman filter with a 3D model of the environment," in *Information Fusion (FUSION), 2012 15th International Conference on*, pp. 2271 –2276, July 2012.

[112] J. i. Meguro, T. Murata, J. i. Takiguchi, Y. Amano, and T. Hashizume, "GPS multipath mitigation for urban area using omnidirectional infrared camera," *IEEE Trans Intell Transp Syst*, vol. 10, no. 1, pp. 22–30, 2009.

[113] C. Pinana-Diaz, R. Toledo-Moreo, F. Toledo-Moreo, and A. Skarmeta, "A two-layers based approach of an enhanced-map for urban positioning support," *Sensors*, vol. 12, no. 11, pp. 14508–14524, 2012.

[114] T. Sonmez and H. E. Bingol, "Modeling and simulation of a terrain aided inertial navigation algorithm for land vehicles," in *2008 IEEE/ION Position, Location and Navigation Symposium*, pp. 1046–1052, May 2008.

[115] N. El-Sheimy, C. Valeo, and A. Habib, *Digital Terrain Modeling: Acquisition, Manipulation, and Applications*. Artech House remote sensing library, Artech House, 2005.

[116] P. D. Groves, R. J. Handley, and A. R. Runnalls, "Optimising the integration of terrain referenced navigation with INS and GPS," *Journal of Navigation*, vol. 59, pp. 71–89, 2006.

[117] M. Adjrad and P. D. Groves, "Enhancing conventional GNSS positioning with 3D mapping without accurate prior knowledge," in *Proceedings of ION GNSS+*, 2015.

[118] M. Adjrad, P. D. Groves, and C. Ellul, "Enhancing GNSS positioning with 3D mapping," in *Proceedings of RIN INC*, 2015.

[119] C. Danezis and V. Gikas, "Performance evaluation of a novel terrain aiding algorithm for GNSS tracking in forested environments," in *Proceedings of the 25th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS 2012)*, (Nashville, TN), pp. 2083–2090, September 2012.

[120] Z. Jiang and P. D. Groves, "GNSS NLOS and multipath error mitigation using advanced multi-constellation consistency checking with height aiding," in *ION GNSS*, September 2012.

[121] L. Lucido, J. Opderbecke, V. Rigaud, R. Deriche, and Z. Zhang, "A terrain referenced underwater positioning using sonar bathymetric profiles and multiscale analysis," in *OCEANS '96. MTS/IEEE. Prospects for the 21st Century. Conference Proceedings*, vol. 1, pp. 417–421 vol.1, Sep 1996.

[122] "Openstreetmap website." online. `https://www.openstreetmap.org/`, accessed 11/5/16.

[123] S. J. Julier and J. K. Uhlmann, "Using covariance intersection for {SLAM}," *Robotics and Autonomous Systems*, vol. 55, no. 1, pp. 3 – 20, 2007. Simultaneous Localisation and Map Building.

[124] Dynastream Innovations Inc., *Flexible & Interoperable Data Transfer (FIT) Protocol*, 2014. D00001275 Rev1.8.

[125] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[126] D. B. Cox Jr., "Integration of GPS with inertial navigation systems," *Navigation: JION*, vol. 25, no. 2, pp. 236–245, 1978.

[127] R. Brown and P. Hwang, *Introduction to Random Signals and Applied Kalman Filtering with MATLAB Exercises, 4th Edition*. Wiley, 2012.

[128] P. S. Maybeck, *Stochastic Models, Estimation, and Control*. Mathematics in Science and Engineering, Elsevier Science, 1982.

[129] R. G. Brown, "Integrated navigation systems and kalman filtering: A perspective," *NAVIGATION, Journal of The Institute of Navigation*, vol. 19, pp. 355–362, Winter 1972-1973.

[130] S. J. Julier and J. K. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in *Proc. AeroSense: The 11th Int. Symp. on Aerospace Defence Sensing, Simulation and Controls, SPIE*, pp. 182–193, 1997.

[131] S. J. Julier and J. K. Uhlmann, "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, pp. 401–422, Mar 2004.

[132] R. K. Mehra, "Approaches to adaptive filtering," *IEEE Trans. on Automatic Control*, vol. AC-17, pp. 693–698, 1972.

[133] A. H. Mohammed and K. P. Schwartz, "Adaptive Kalman filtering for INS/GPS," *Journal of Geodesy*, vol. 73, pp. 193–203, 1999.

[134] D. T. Magill, "Optimal adaptive estimation of sampled stochastic processes," *IEEE Trans. on Automatic Control*, vol. AC-10, pp. 434–439, 1965.

[135] C. Hide, T. Moore, and M. Smith, "Adaptive Kalman filtering for low cost INS/GPS," in *Proceedings of the 15th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2002)*, (Portland, OR), pp. 1143–1147, September 2002.

[136] D.-J. Lee and K. Alfriend, "Adaptive sigma point filtering for state and parameter estimation," in *AIAA/AAS Astrodynamics Specialist Conference and Exhibit Guidance, Navigation, and Control and Co-located Conferences*, (Providence, Rhode Island), American Institute of Aeronautics and Astronautics, 2004.

[137] C. Goodall, X. Niu, and N. El-Sheimy, "Intelligent tuning of a Kalman filter for INS/GPS navigation applications," in *Proceedings of the 20th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2007)*, (Fort Worth, TX), pp. 2121–2128, September 2007.

[138] V. Strassen, "Gaussian elimination is not optimal," *Numer. Math.*, vol. 13, pp. 354–356, 1969.

[139] D. Coppersmith and S. Winograd, "Matrix multiplication via arithmetic progressions," *J. Symbolic Comput.*, vol. 9, pp. 251–280, 1990.

[140] F. Gustafsson *et al.*, "Particle filters for positioning, navigation and tracking," in *IEEE Trans. on Signal Processing*, pp. 425–437, 2002.

[141] A. Doucet, N. de Freitas, and N. Gordon, eds., *Sequential Monte Carlo Methods in Practice*. Springer Verlag, 2001.

[142] L. Wang, "Kinematic GNSS shadow matching using particle filters," in *ION GNSS+ 2014*, 2014.

[143] S. Maskell and S. Julier, "Optimised proposals for improved propagation of multimodal distributions in particle filters," in *Information Fusion (FUSION), 2013 16th International Conference on*, pp. 296–303, July 2013.

[144] M. Dumville and M. Tsakiri, "An adaptive filter for land navigation using neural computing," in *Proceedings of the 7th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 1994*, (Salt Lake City, UT), pp. 1349–1356, September 1994.

[145] L. Semeniuk and A. Noureldin, "Modelling of INS position and velocity errors using radial basis function neural networks for INS/GPS integration," in *Proceedings of the 2006 National Technical Meeting of The Institute of Navigation*, (Monterey, CA), pp. 479–489, January 2006.

[146] W. Abdel-Hamid, A. Noureldin, and N. El-Sheimy, "Adaptive fuzzy prediction of low-cost inertial-based positioning errors," *Fuzzy Systems, IEEE Transactions on*, vol. 15, pp. 519–529, June 2007.

[147] D. W. Allan, "Statistics of atomic frequency standards," *Proceedings of the IEEE*, 1966.

[148] N. El-Sheimy, H. Hou, and X. Niu, "Analysis and modeling of inertial sensors using allan variance," *IEEE Transactions on Instrumentation and Measurement*, vol. 57, pp. 140–149, Jan 2008.

[149] R. W. Mitchell, "Laboratory motion testing of MEMS inertial measurement units (IMU)," in *Proceedings of the 63rd Annual Meeting of The Institute of Navigation*, (Cambridge, MA), pp. 735–741, April 2007.

[150] P. Aggarwal, Z. Syed, and N. El-Sheimy, "Thermal calibration of low cost MEMS sensors for land vehicle navigation system," in *Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE*, pp. 2859–2863, May 2008.

[151] Xsens Technologies B.V., *MTi-G User Manual and Technical Documentation*, October 2010. Document MT0137P Rev. H.

[152] Honeywell Aerospace, Phoenix, AZ, *HG1900 Inertial Measurement Unit High Performance 17 cubic inch MEMS IMU.* Rev E713908-2012.

[153] 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*, vol. 18, no. 7, p. 1897, 2007.

[154] C. Ren, Q. Liu, and T. Fu, "A novel self-calibration method for MIMU," *IEEE Sensors Journal*, vol. 15, pp. 5416–5422, Oct 2015.

[155] J.-O. Nilsson, I. Skog, and P. Händel, "Aligning the forces eliminating the misalignments in IMU arrays," *IEEE Transactions on Instrumentation and Measurement*, no. 99, 2014.

[156] W. T. Fong, S. K. Ong, and A. Y. C. Nee, "Methods for in-field user calibration of an inertial measurement unit without external equipment," *Measurement Science and Technology*, vol. 19, no. 8, p. 085202, 2008.

[157] I. Skog and P. Händel, "Calibration of a MEMS inertial measurement unit," *XVII IMEKO World Congress*, pp. 1–6, 2006.

[158] P. Aggarwal, Z. Syed, X. Niu, and N. El-Sheimy, "A standard testing and calibration procedure for low cost MEMS inertial sensors and units," *Journal of Navigation*, vol. 61, pp. 323–336, 4 2008.

[159] M. Hwangbo and T. Kanade, "Factorization-based calibration method for MEMS inertial measurement unit," in *Proceedings International Conference on Robotics and Automation*, May 2008.

[160] E.-H. Shin and N. El-Sheimy, "A new calibration method for strapdown inertial navigation systems," *Zeitschrift für Vermessungswesen.–2002.–Zfv*, vol. 127, no. 1, pp. 41–50, 2002.

[161] C. M. Cheuk, T. K. Lau, K. W. Lin, and Y. Liu, "Automatic calibration for inertial measurement unit," in *Control Automation Robotics Vision (ICARCV), 2012 12th International Conference on*, pp. 1341–1346, Dec 2012.

[162] T. Nieminen, J. Kangas, S. Suuriniemi, and L. Kettunen, "An enhanced multiposition calibration method for consumer-grade inertial measurement units applied and tested," *Measurement Science and Technology*, vol. 21, no. 10, 2010.

[163] A. Waegli, S. Guerrier, and J. Skaloud, "Redundant MEMS-IMU integrated with GPS for performance assessment in sports," in *Position, Location and Navigation Symposium, 2008 IEEE/ION*, pp. 1260–1268, May.

[164] I. Skog, J.-O. Nilsson, and P. Händel, "An open-source multi inertial measurement units (MIMU) platform," in *Proc. Int. Symp. on Inertial Sensors and Systems (ISISS)*, (Laguna Beach, CA, USA), February 2014.

[165] Honeywell Defense & Space, Clearwater, FL, *Miniature Inertial Measurement Unit: proven performer for satellite and deep-space-probe applications*, 2006.

[166] M. Tanenhaus, T. Geis, D. Carhoun, and A. Holland, "Accurate real time inertial navigation device by application and processing of arrays of MEMS inertial sensors," in *Position Location and Navigation Symposium (PLANS), 2010 IEEE/ION*, pp. 20 –26, may 2010.

[167] A. R. Schuler, A. Grammatikos, and K. A. Fegley, "Measuring rotational motion with linear accelerometers," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-3, pp. 465–472, May 1967.

[168] C.-W. Tan and S. Park, "Design of accelerometer-based inertial navigation systems," *Instrumentation and Measurement, IEEE Transactions on*, vol. 54, pp. 2520 – 2530, December 2005.

[169] W.-T. Ang, P. Khosla, and C. Riviere, "Design of all-accelerometer inertial measurement unit for tremor sensing in hand-held microsurgical instrument," in *IEEE International Conference on Robotics and Automation*, pp. 1781–1786, IEEE, September 2003.

[170] P. Sun, H. Qian, and H. Zhao, "SDRE filter for improving the performance of a gyroscope-free IMU," in *Intelligent System Design and Engineering Application (ISDEA), 2012 Second International Conference on*, pp. 1302–1305, Jan.

[171] I. Skog, J. O. Nilsson, P. Handel, and A. Nehorai, "Inertial sensor arrays, maximum likelihood, and cramer-rao bound," *IEEE Transactions on Signal Processing*, vol. PP, no. 99, pp. 1–1, 2016.

[172] T. Hudson, S. Holt, P. Ruffin, J. McKee, M. Whitley, M. Kranz, and E. Tuck, "Multi-sensor inertial array technologies for advanced microsystems," in *Position, Location, And Navigation Symposium, 2006 IEEE/ION*, pp. 574–585, April 2006.

[173] J. F. Wagner, G. Kasties, and M. Klotz, "An alternative approach to integrate satellite navigation and inertial sensors," in *Proc. ION NTM*, (Santa Monica, CA), pp. 141–150, January 1997.

[174] G. Dissanayake *et al.*, "The aiding of a low-cost strapdown inertial measurement unit using vehicle motion constraints for land vehicle applications," *IEEE Trans. on Robotics and Automation*, vol. 17, no. 5, pp. 731–747, 2001.

[175] DJI, "DJI phantom specifications." `http://www.dji.com/product/phantom/spec`, accessed 27/01/15.

[176] K. Voutsis, P. D. Groves, C. Ford, and M. Holbrow, "The importance of human motion for simulation testing of GNSS," in *Proc. ION GNSS*, September 2014.

[177] M. De Agostino, A. M. Manzino, and M. Piras, "Performances comparison of different MEMS-based IMUs," in *Position Location and Navigation Symposium (PLANS), 2010 IEEE/ION*, pp. 187–201, IEEE, 2010.

[178] S. Godha, G. Lachapelle, and M. Cannon, "Integrated GPS/INS system for pedestrian navigation in a signal degraded environment," in *Proceedings of ION GNSS 2006*, (Fort Worth, TX), September 2006.

[179] S. Kwakkel, G. Lachapelle, and M. Cannon, "GNSS aided in situ human lower limb kinematics during running," in *Proceedings of the 21st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2008)*, (Savannah, GA), pp. 1388–1397, September 2008.

[180] I. Skog, J.-O. Nilsson, and P. Händel, "Evaluation of zero-velocity detectors for foot-mounted inertial navigation systems," in *Indoor Positioning and Indoor Navigation (IPIN), 2010 International Conference on*, pp. 1–6, IEEE, 2010.

[181] "MATLAB simulated annealing: Find global minima for bounded nonlinear problems." online. `http://uk.mathworks.com/discovery/simulated-annealing.html`, accessed 09/04/16.

[182] InvenSense Inc., 1745 Technology Drive, San Jose, CA 95110 U.S.A., *MPU-9250 Register Map and Descriptions*, September 2013. Document Number: RM-MPU-9250A-00, Revision: 1.4.

[183] "Product page: Newport RPR Reliance^{TM} industrial and educational grade optical tables." online. `http://www.newport.com/RPR-Reliance-Series-Industrial-and-Educational-Gr/139727/1033/info.aspx#tab_Specifications` accessed 30/04/16.

[184] A. H. Cook, "A new absolute determination of the acceleration due to gravity at the national physical laboratory," *Nature*, vol. 208, p. 279, October 1965.

[185] G. Appleby, V. Smith, M. Wilkinson, M. Ziebart, and S. Williams, *Gravity, Geoid and Earth Observation: IAG Commission 2: Gravity Field, Chania, Crete, Greece, 23-27 June 2008*, ch. Comparison of Height Anomalies Determined from SLR, Absolute Gravimetry and GPS with High Frequency Borehole Data at Herstmonceux, pp. 107–113. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010.

[186] P. Folmar, "The truth about placement accuracy: How to weigh variables that will affect long-term yields." online. Solid State Technology Insights for Electronics Manufacturing, `http://electroiq.com/blog/2000/04/the-truth-about-placement-accuracy/` accessed 20/4/16.

[187] J. B. Bancroft and G. Lachapelle, "Data fusion algorithms for multiple inertial measurement units," *Sensors*, vol. 11, no. 7, pp. 6771–6798, 2011.

[188] Analog Devices, One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A., *ADXRS150 ±150°/s Single Chip Yaw Rate Gyro with Signal Conditioning Datasheet*, 2004. Rev B.

[189] "MATLAB ksdensity: Kernel smoothing function estimate for univariate and bivariate data." online. `http://uk.mathworks.com/help/stats/ksdensity.html` accessed 21/4/16.

[190] A. Bowman and A. Azzalini, *Applied Smoothing Techniques for Data Analysis: The Kernel Approach with S-Plus Illustrations*. Oxford Statistical Science Series, OUP Oxford, 1997.

[191] "Arduino project homepage." `www.arduino.cc`.

[192] Bosch Sensortec, *BMP180 Digital pressure sensor datasheet*, January 2012. Rev. 2.4, BST-BMP180-DS000-08.

[193] Atmel Corp., *AT32UC3A3/A4 Series Summary*, October 2012. Rev. H.

[194] "Atmel AT32UC3A3256S product page." online. `http://www.atmel.com/devices/AT32UC3A3256S.aspx`, accessed 06/04/16.

[195] NXP Semiconductors, *UM10204 I2C-bus specification and user manual*, June 2007. Rev. 03.

Appendix A

Development of electronic hardware

This thesis presents ideas for the use of inertial sensors. To test these ideas some physical experiments needed to be carried out, and for these several pieces of electronic equipment were made. This test bed is required to log the output of multiple inertial sensors simultaneously to a storage device (e.g. SD card) which can be used to implement the calibration algorithms and assess their effectiveness. The development process of these experimental platforms is described in this appendix. As the physical designs do not constitute the novel contribution of this thesis this development process is treated briefly.

Preliminary research into the error behaviour of different models of inertial sensors required a simple and fast turn-around experimental platform, which could flexibly sample many different models, and read different sub-sets of them. The one chosen is based on Arduino-family microcontrollers. This was chosen because it could quickly be built based on a combination of commercially-available circuit boards for which considerable libraries of code were already available. This is described in Section A.1.

However, in order to conduct better tests of the ideas presented in this thesis a higher-performance experimental platform was also built. This is based around array of single sensor model and uses a much higher-performance Atmel AVR32 microcontroller. However, it required a completely bespoke printed circuit board (PCB) design and custom-written firmware. This is described in Section A.2.

In addition, during the development of the calibration techniques presented in Chapter 4, two calibration cubes were made that could contain the test-beds. These cubic or cuboid frames are intended to enclose the circuit boards. There is a larger but less-accurately made one which can contain either test setup, and there is a smaller more accurately orthogonal-sided one designed for the high-performance array. These are described in Section A.3

A.1 Arduino-based test platforms

During the initial exploratory phase of the research presented in this thesis several different experimental platforms were constructed by combining commercially available circuit boards. These included microcontroller boards, secure digital (SD) memory card readers, and various so called break-out boards containing individual inertial sensors. These break-

out boards consist of individual sensors and small peripheral components, such as capacitors and/or resistors, on a small circuit board with 0.1 inch spaced holes[1] on one side connected to the pins of the inertial sensor chip, so that it can easily be used on a breadboard. An example is shown in Figure A.1. The purpose of this test-bed is to simultaneously log an array of inertial sensors consisting of a mixture of different models to allow their comparison, although this was only partially successful for reasons detailed below.

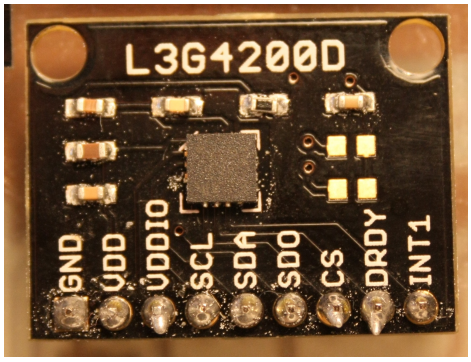


Figure A.1: A break-out PCB containing an L3G4200D gyroscope [24].

The Arduino platform was used because of its ease of use and the large variety pre-existing libraries and peripherals for it (for more information see [191]). Several different preliminary breadboard test platforms were made before the semi-permanent one shown in Figure A.2 was constructed. This set-up is the one used for the experiments in [8]. This experimental platform is better than a breadboard set-up because the breakout boards are soldered down, making the connections more electrically reliable and fixing the position of the sensors better. However, the sensors are still up to 15cm apart and, particularly in some harsher vibration environments, they may not be fixed down adequately.

The basic firmware architecture of Arduino microcontrollers is that there is a setup loop which is run once and a main loop which is run continuously until the test platform is powered down. The setup loop sends the commands to power up the sensors and other necessary peripheral parts of the hardware. Then it programs registers on the sensors for settings, such as the measurement ranges and on-board filters and, finally, it sets them to start generating measurements. The main running loop of the experimental platform reads the sensors and records their output on a micro-SD card.

So-called digital inertial sensors are used for this research meaning that they have an analogue to digital converter (ADC) built into the sensors FLP. This means that the output of the sensors does not depend on the microcontroller or ADC used to read them or, for example, the length of the wiring path between the sensor and the microcontroller. This means that rather than an analogue voltage being sent from the sensors to the microcontroller board, the measurements are passed by a digital communications protocol. The two protocols commonly used on consumer-grade inertial sensors are serial peripheral interface (SPI) and inter integrated circuit (I2C or I^2C), which is also known as two-wire interface (TWI). For some inertial sensors both of these interfaces are available, but SPI is only available for some of those which were to be tested. The main advantage of SPI is that it allows a faster clock speed, if the peripherals are capable of handling it, and the address of the peripheral being communicated with does not need to be included in the message. Both of these mean that there is a faster theoretical reading speed. One disadvantage of SPI is that only a few sensor models support it. Another disadvantage is that it requires two data

[1] 2.54mm the standard electronic breadboard spacing, also known as 100 mil.

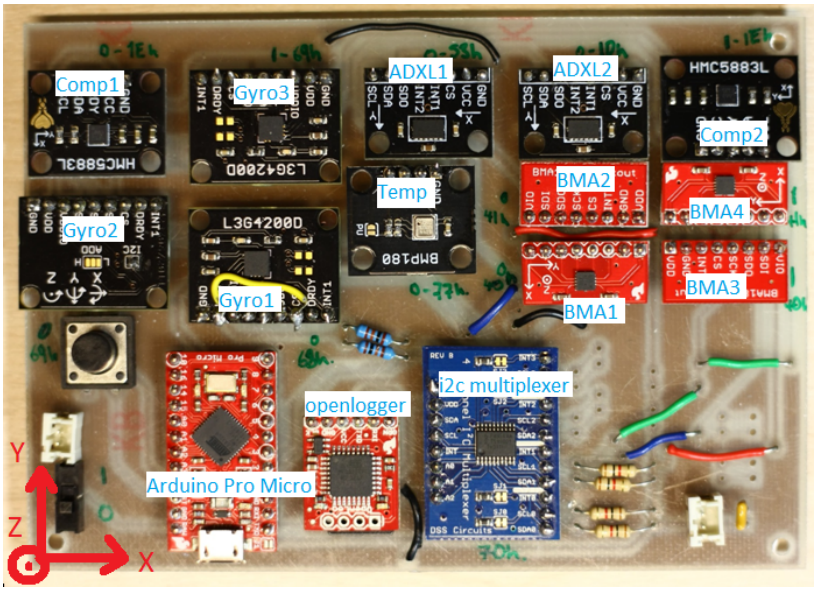


Figure A.2: One of the test beds constructed for inertial sensor testing based around an Arduino microcontroller. It contains ADXL345 ('ADXL' [49]) and BMA180 ('BMA' [52]) accelerometers, L3G4200D gyroscopes ('Gyro' [24]), Honeywell magnetometers ('Comp' [77]) and a temperature and pressure sensor ('Temp' [192])

lines ('master'-to-'slave' and 'slave'-to-'master'[2]), a clock line and one chip select line *per sensor*, to identify which peripheral is being addressed. In contrast, I2C only needs a clock and a single data line, because an address is included at the start of each message which identifies the peripheral that the message is intended for. However, fundamentally both of these interfaces are *serial*, that is, communication takes place one sensor at a time. A method by which parallel communication can be 'hacked' will be discussed in Section A.2.

This serial reading means that the maximum possible logging rate depends on the number of sensors that need to be read. Thus, as the arrays of sensors get larger, this becomes a more serious problem. In addition, the array-based techniques for which these test beds are intended assume that all the sensors are read at the same epoch, which with serial reading is clearly impossible. The time between each sensor being read can be reduced by first reading all the sensors in the array, saving each measurement to the on-board memory and then sending all the measurements together to the micro-SD card at the end. There is a further complication when using I2C to communicate with arrays; the address mentioned above, is a 2-digit hex number (i.e. 0 to 255). While this would in theory not be a problem until the arrays get very large, the reality is that the manufacturers fix this 'slave address' to one of two values allowing only the LSB to be changed. Thus as soon as the array contains more than two of one sensor model you need either a microcontroller with multiple separate I2C buses, of which very few have more than two, or to use a multiplexer to split the bus. This second option is used in the circuit depicted in Figure A.2, and the multiplexer can be seen on the blue breakout board. This does mean extra I2C commands to switch between halves of the multiplexed bus.

[2]The use of this politically incorrect terminology is regrettable. However, it is used in this thesis as these are the standard terms used to describe this type of communication in Electronic Engineering.

As an example of how these various performance-limiting factors added up the test-bed pictured in Figure A.2 could record the output of the 9 inertial sensors pictured and the temperature-and-pressure sensor (but not the two magnetometers) at about 80 Hz. Of course, this means that it could have logged one sensor at about 800 Hz, but many sensors needed to be read to test the array ideas.

Some improvements in logging speed could have been made by using a higher power microcontroller, but the maximum clock rate of I2C at 400 kHz is a major limiting factor. Switching to SPI communication would also have slightly helped for the reasons discussed above but it would have been a small improvement rather than many multiples faster. This is partly because the open-source method of sending data to an SD card, which incidentally is SPI, is slow so that this is also a major limiting factor.

A.2 High-speed simultaneously-logging array of MPU-9250 IMUs

The limited logging rate, caused by reading and logging the sensors one-by-one, makes it potentially difficult to correctly capture dynamic motion, additionally the break-out boards place the sensors further apart than would be possible if chips were placed on a single PCB creating a larger lever-arm between them, which is also potentially less rigid than a single board. Both of these make the Arduino-based test-beds less than ideal for dynamic tests. So, considering these limitations, a higher-performance experimental platform was constructed with all the sensors on a single PCB, and considerably closer together. For this final piece of hardware only a single model of IMU was chosen: the Invensense MPU9250 [23]. 16 of these sensors were connected to a much higher performance Atmel microcontroller the AT32UC3A3256S [193]. This was chosen because it is much faster than the microcontrollers used on an Arduino. It also includes a 4-bit wide parallel bus for SD-card writing functionality which significantly increases the writing speed (also see [194]). This high-performance platform is shown in Figure A.3.

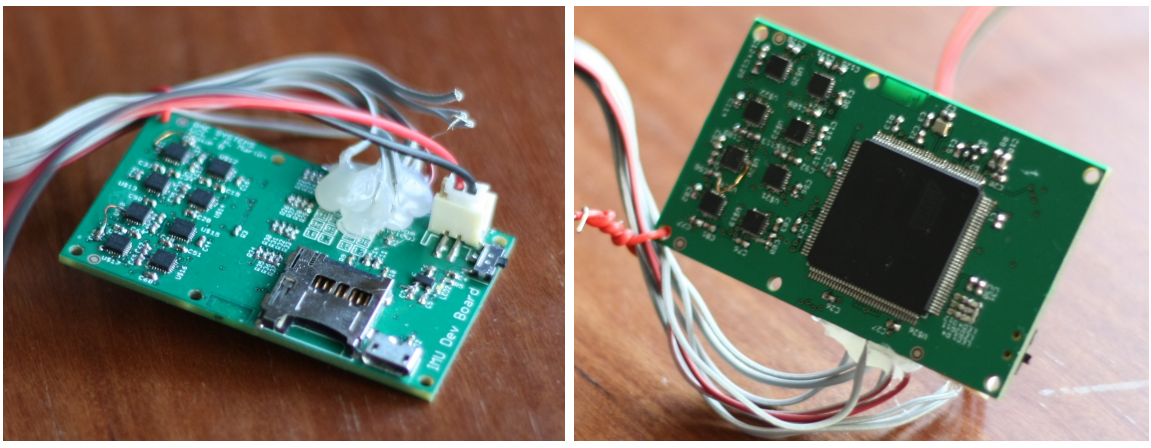


Figure A.3: The high-performance test bed. Both front and rear are shown.

However, the really important functional difference between this high-performance platform

and the previous Arduino-based platforms is that the sensors are read simultaneously. The idea behind this comes from the "OpenShoe MIMU" of Skog et al. [164]. The idea is that the I2C 'slaves', that is the IMU chips, are fooled into all responding simultaneously to a single command sent on the I2C bus. To accomplish this the sensor wiring is changed. Instead of a single clock and data line connected to all the sensors there is a separate data line attaching each sensor to the microcontroller chip. Thus, from the point of view of each IMU, it is the only 'slave' on the bus and so the microcontroller can send 'read measurement' commands simultaneously to all the sensors. However, this circumvented or 'hacked' architecture creates certain other problems. First, as mentioned in Section A.1, it is uncommon for a microcontroller chip to have more than two hardware-defined I2C buses, so when connecting 16 sensors (or 32 in Openshoe's case) the microcontroller's regular general-purpose input-output (GPIO) pins needed to be used. This means that, rather than using the built-in I2C functionality, low-level 'bit-banging' code needed to be written. In fact this is needed anyway to conduct the simultaneous read. Using a single shared clock line for all sensors reduces the number of instructions that the bit-banged code needs to create the correct signal for 'read measurement output registers of 'slave' (sensor address)' on all the data lines simultaneously. Due to the way that GPIO pins on a microcontroller work internally changing the value of a group of pins like this is not actually slower than changing the value of a single pin. Then in accordance to the I2C protocol [195], all the IMUs respond down their own data line with their measurement. At the microcontroller end, rather than just recording the data byte digit-by-digit as it comes off a single pin, the outputs of all the pins for a particular digit of the response are recorded in an array. Once the transmission has finished the array can be transposed to reconstruct each sensor's response. This is done to improve computational speed.

Both the hardware and the firmware/software design for this test-bed were fully bespoke. The results of this fully custom design is that the experimental platform can record the output of the 16 sensors on the board at 1000 Hz. While the operating principle is based on the OpenShoe idea, in order to support the SD card at high speed a different model of microcontroller was used. Also a newer model model of IMU was used and to have the sensors at the correct orientations as required for the ideas in Chapter 6 the layout was also changed, these orientations are shown in Figure A.4. These changes among others also made it necessary to write the firmware from scratch. This process of bespoke design took in total approximately 15 months.

Due to the limited resources available to this research project, it was not possible to build a second iteration of the hardware design so some mistakes made at the design stage of the PCB could not be fully rectified. Some mistakes such as non-ideal choice of on-board oscillator could be rectified in the firmware, but others, most importantly the incorrect wiring to the programming interface meant external wires had to be soldered to the surface of the board, replacing, for example the socket in the design with a soldered and glued on ribbon cable (see Figure A.3). Unfortunately, as is also a common problem with hand-placing surface-mount (SMD) components, particularly ones as small as the 3x3mm leadless package MPU-9250s, two of the sensors were damaged during fabrication, so only 14 of the 16 sensors give reliable measurements.

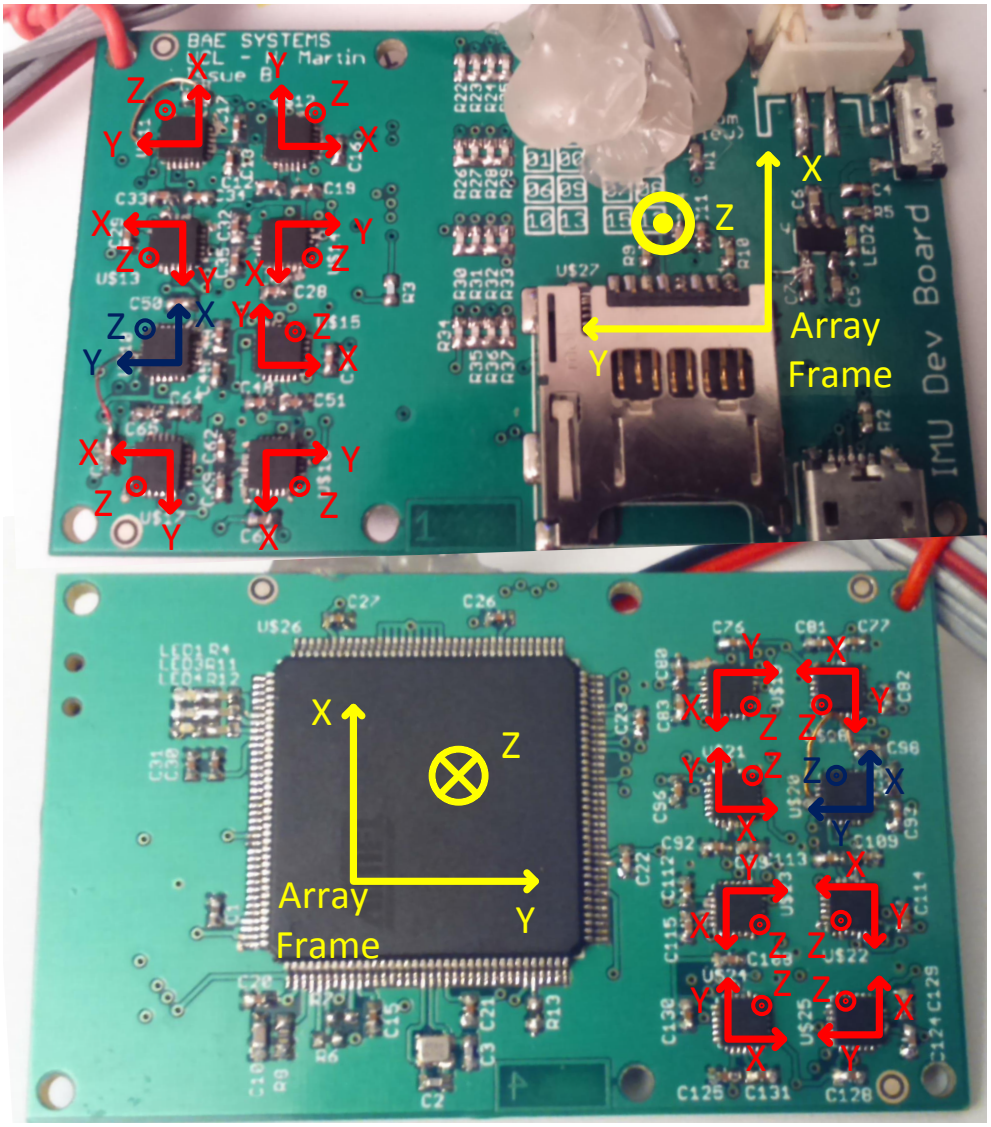
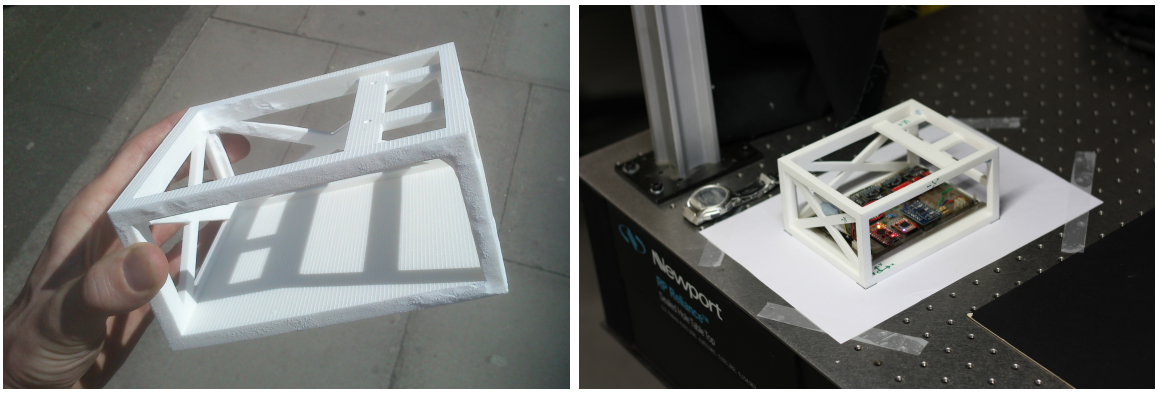


Figure A.4: The orientations of the sensors on the high-performance test bed. Array frame is yellow, sensors are red. The two broken sensors in dark blue.

A.3 Calibration frames

The calibration techniques described in Chapter 4 use a cuboid calibration box or frame in which the sensors are mounted. This is then used to enable static recordings to be made at orthogonal attitudes by resting different faces of the cube on a flat surface.

Initially, a cube was constructed by building a design in SketchUp and having it 3D printed. The printing was carried out on a laser-sintering machine, the plastic powder and the bedding/construction planes are visible in Figure A.5a. However, it is worth considering that 3D printing is a relatively new technique, and as such is not as mature as other construction methods. Perhaps as a result of this the angular accuracy of the cube's walls was rather disappointing, the result being a noticeable bowing of the cube across its long side. This was partially alleviated by having the outside of the bottom solid face machined to render it flatter.While most construction techniques will be more angularly accurate when making a larger items, because of fixed distance tolerances, 3D printing is not. 3D printers frequently



(a) Fresh from 3D printing. (b) During testing with an Arduino test-bed

Figure A.5: The 3D printed calibration frame.

better calibrated in the centre of their printable area with accuracy dropping off towards the edges. Thus for a given printer the angular accuracy is not likely to improve as the box becomes larger. The box is large enough to mount the Arduino test bed of Figure A.2, as is shown in Figure A.5b.

In order to improve on the first cube's drawbacks, a second frame was constructed. This was made by the much more established technique of machining aluminium. It is constructed from four pieces of high-quality aluminium alloy which after being permanently assembled were post-machined to improve its accuracy.

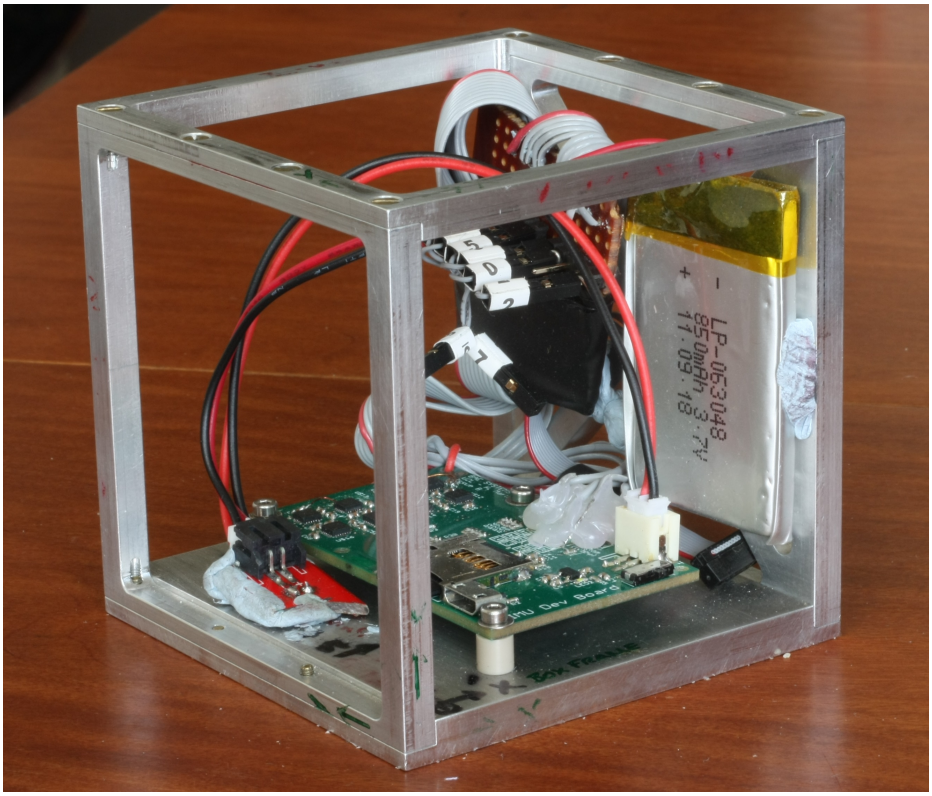


Figure A.6: The aluminium calibration cube.

This cube is shown in Figure A.6, fitted with the MPU9250 array. Note the IMU manufacturer's recommended 3-point mounting and the precision placed holes and machined plastic supports, this was done to minimise the mounting error, $\mathbf{C}_{\mathbf{b}}^{\mathbf{s}}$. This cube is smaller than the 3D printed one with 75mm (external) sides, and thus only the advanced experimental platform fits inside. This cube is the one used for the experiments presented in Chapter 5.