DETERMINISTIC AND STOCHASTIC ERROR MODELING OF INERTIAL SENSORS AND MAGNETOMETERS

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND

ELECTRONICS ENGINEERING

AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Görkem Seçer

August 2012

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Billur Barshan(Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Sinan Gezici

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Dr. Volkan Nalbantoğlu

Approved for the Graduate School of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Graduate School

ABSTRACT

DETERMINISTIC AND STOCHASTIC ERROR MODELING OF INERTIAL SENSORS AND MAGNETOMETERS

Görkem Seçer
M.S. in Electrical and Electronics Engineering
Supervisor: Prof. Dr. Billur Barshan
August 2012

This thesis focuses on the deterministic and stochastic modeling and model parameter estimation of two commonly employed inertial measurement units. Each unit comprises a tri-axial accelerometer, a tri-axial gyroscope, and a tri-axial magnetometer. In the first part of the thesis, deterministic modeling and calibration of the units are performed, based on real test data acquired from a flight motion simulator. The deterministic modeling and identification of accelerometers is performed based on a traditional model. A novel technique is proposed for the deterministic modeling of the gyroscopes, relaxing the test bed requirement and enabling their in-use calibration. This is followed by the presentation of a new sensor measurement model for magnetometers that improves the calibration error by modeling the orientation-dependent magnetic disturbances in a gimbaled angular position control machine. Model-based Levenberg-Marquardt and model-free evolutionary optimization algorithms are adopted to estimate the calibration parameters of sensors. In the second part of the thesis, stochastic error modeling of the two inertial sensor units is addressed. Maximum likelihood estimation is employed for estimating the parameters of the different noise components of the sensors, after the dominant noise components are identified. Evolutionary and gradient-based optimization algorithms are implemented to maximize the likelihood function, namely particle swarm optimization and gradient-ascent optimization. The performance of the proposed algorithm is verified through experiments and the results are compared to the classical Allan variance technique. The results obtained with the proposed approach have higher accuracy and require a smaller sample data size, resulting in calibration experiments of shorter duration. Finally, the two sensor units are compared in terms of repeatability, present measurement noise, and unaided navigation performance.

Keywords: Inertial sensors, deterministic error modeling, stochastic error modeling, in-field calibration, Levenberg-Marquardt algorithm, particle swarm optimization, gradient-ascent optimization, Allan variance, maximum likelihood estimation.

ÖZET

EYLEMSİZLİK DUYUCULARININ VE MANYETOMETRELERİN DETERMİNİSTİK VE STOKASTİK HATA MODELLEMESİ

Görkem Seçer
Elektrik ve Elektronik Mühendisliği, Yüksek Lisans
Tez Yöneticisi: Prof. Dr. Billur Barshan
Ağustos 2012

Bu tezde orta performanslı ve düşük maliyetli, yaygın kullanımdaki iki eylemsizlik duyucu ünitesinin deterministik ve stokastik hata modellemesi ele alınmıştır. Her bir ünite, üç boyutta ölçüm alabilen bir ivmeölçer, bir dönüölçer ve bir manyetometre içermektedir. Tezin ilk bölümünde, bir uçuş hareket simülatörü üzerinde kontrollü deneyler sonucunda elde edilen veriler kullanılarak ünitelerin deterministik hata modellenmesi ve kalibrasyonu gerçekleştirilmiştir. İvmeölçerler için klasik hata modelleri ve parametre kestirim yöntemleri kullanılmıştır. Dönüölçerlerin hata model parametrelerinin kestirimi için yeni bir yöntem önerilmiştir. Bu yeni yöntem, dönüölçerlerin klasik yöntemlerle kalibrasyonunda kullanılan açısal hız kontrol cihazlarına olan gereksinimi ortadan kaldırmıştır. Manyetometreler için ise gimballi sistemlerde bulunan açısal pozisyona bağlı olarak değişen bozucu manyetik etkileri de içeren yeni bir model öne sürülerek kalibrasyon hataları azaltılmıştır. Sensörlerin kalibrasyonu için model tabanlı Levenberg-Marquardt ve modelden bağımsız evrimsel eniyileme algoritmaları kullanılmıştır. Tezin ikinci kısmında, eylemsizlik duyucu ünitelerinin stokastik modellemesi yapılmıştır. Duyucu ölçümlerinin içerdiği baskın gürültü tipleri belirlendikten sonra, bu gürültülere ilişkin model parametreleri en büyük olabilirlik kestirimi yöntemiyle kestirilmiştir. Ölçümlerin olabilirlik fonksiyonunu en büyükleyen stokastik model parametrelerinin kestirimi için evrimsel ve gradyan-tabanlı eniyileme algoritmaları kullanılmıştır. Bu algoritmalar, parçacık sürü optimizasyonu ve artan gradyan optimizasyonudur. Önerilen algoritmanın başarımı deneylerle kanıtlanmış ve sonuçlar klasik Allan değişinti yöntemi ile karşılaştırılmıştır. Buna göre, önerilen yöntemler aracılığıyla daha kısa süreli deney verileri kulanılarak daha yüksek doğruluklu kestirimler elde edildiği gözlenmiştir. Sonuç olarak, iki ünite tekrar edilebilirlik, ölçümlerdeki gürültü miktarı ve yöngüdüm başarımı açısından karşılaştırılmıştır.

Anahtar sözcükler: Eylemsizlik duyucu üniteleri, deterministik hata modellemesi, stokastik hata modellemesi, kullanım sırasında hata kalibrasyonu, Levenberg-Marquardt algoritması, parçacık sürü optimizasyonu, artan gradyan optimizasyonu, Allan varyans, en büyük olabilirlik kestirimi.

Acknowledgement

First, I would like to thank my supervisor Prof. Dr. Billur Barshan for her guidance, patience, support, and criticism throughout my study.

I want to thank Assist. Prof. Dr. Sinan Gezici and Dr. Volkan Nalbantoğlu for their guidance and criticism.

I would also like to thank some friends for their support and friendship: Tuğrul Yıldırım, İsmail Uyanık, Deniz Kerimoğlu, Hasan Hamzaçebi, Veli Tayfun Kılıç, Çağatay Tanıl, and Gökhan Tüşün.

ROKETSAN Inc. who supported this work is acknowledged. I want to thank my coworkers Derya Ünsal and Can Taşan for helping me during experiments.

Most importantly, I would like to thank Çağlar Coşarderelioğlu for her everlasting support, love and encouragement. This thesis could not be completed without her.

Finally, I would like to thank to my family for their love, support, and encouragement. This thesis is dedicated to my late father Osman Seçer.

Contents

List of Figures

List of Tables

Chapter 1

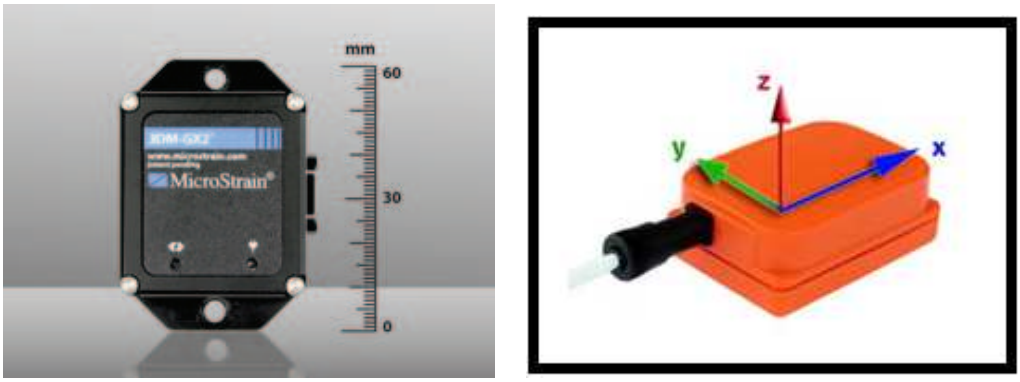
Introduction

Inertial sensors and magnetometers are measurement devices having a broad range of application areas. Basic types of inertial sensors are accelerometers, gyroscopes, and inclinometers.

The size, weight, and the cost of inertial sensors have decreased considerably during the last two decades. Formerly, these devices have been mainly used in aeronautics and maritime applications because of the high cost associated with the high accuracy requirements. The availability of lower cost, medium performance inertial sensor units has opened up new possibilities for their use. Some of the more recent application areas are physical therapy and home-based rehabilitation [12], medical diagnosis and treatment [13], telesurgery [14, 15], biomechanics [16, 17], detecting and classifying falls [18, 19, 20], remote monitoring of the physically or mentally disabled, the elderly, and children [21], ergonomics [22], shock and vibration analysis in the automotive industry, navigation of unmanned vehicles [23, 24, 25], state estimation and dynamic modeling of legged robots [26, 27], sports science [28], ballet and other forms of dance [29], animation and film making, computer games [30, 31], professional simulators, virtual reality, and stabilization of equipment through motion compensation.

The sensing units usually contain gyroscopes and accelerometers, and sometimes, magnetometers in addition. Some of these devices are sensitive around

a single axis whereas others are multi-axial (usually two- or three-axial). Two examples are shown in Figure 1.1.



(a) (b)

Figure 1.1: Illustrations of the IMUs used in the thesis [1, 2]. (a) MicroStrain 3DM-GX2 and (b) Xsens MTx.

Accelerometers are devices sensing the specific acceleration which is the acceleration relative to free fall [3]. In its simplest form, the accelerometer contains a proof mass connected via a spring to the case of the instrument as shown in Figure 1.2. If the accelerometer falls freely within a gravitational field, there will be no extension or compression of the spring since the case and the proof mass will fall together. Thus, the output of the sensor will remain at zero while the acceleration of the sensor will be equal to the gravitational acceleration g. Conversely, an accelerometer measures acceleration corresponding to the gravitational force stopping it from falling when it is stationary. Therefore, accurate knowledge of the gravitational field is essential to compensate this offset in the measurements.

Gyroscopes are devices sensing the angular rate. The most basic and original form of gyroscopes are mechanical and makes use of the inertial properties of a wheel or rotor spinning at high speed [32]. Schematic drawing of such a mechanical gyroscope sensing the angular rate of the input axis is shown in Figure 1.3. Torque is induced about the output axis when a torque is applied about the input axis because of the constantly spinning disk. This phenomenon is known as precession in rigid body dynamics [33]. Induced torque is proportional to the angular rate of the input axis. This rate can be measured by sensing the induced

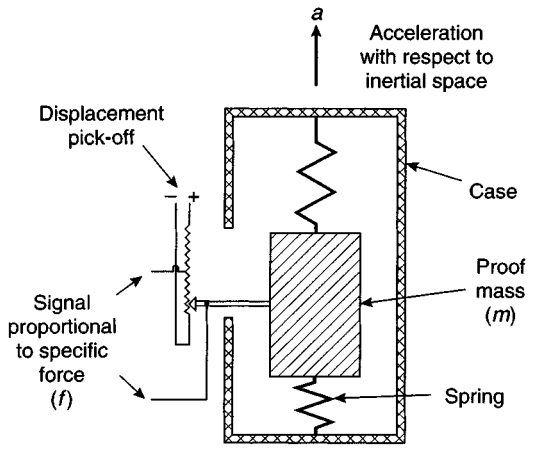


Figure 1.2: A simple accelerometer [3].

torque (i.e., sensing the tension on the restraining springs). The internal view of a rate gyroscope used in 1950s is illustrated in Figure 1.4.

Since inertial sensors provide rate output, their output needs to be integrated once or twice to get angular/linear position information. Thus, even very small errors at their output accumulate very quickly, and the output tends to drift in time.

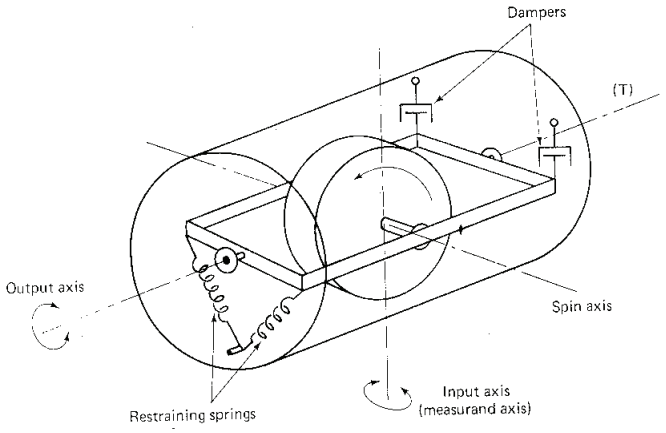


Figure 1.3: A simple mechanical spinning mass rate gyroscope [3].

In addition to inertial sensors, there are other types of sensors (e.g., a magnetic compass) used to determine the kinematics of a body. Magnetic field sensors called magnetometers are main elements of them. Magnetometers are instruments measuring the strength and direction of the nearby magnetic fields. They are

Figure 1.4: The JG7005 rate gyroscope used in 1950s [4].

used in a wide range of disciplines from archeology [34] to inertial navigation [35]. Their general application in inertial navigation is to determine the attitude by comparing their measurements with the Earth's own magnetic field [36].

None of these sensors are perfect. Their measurements deviate from actual signals as they suffer from certain errors. It is a commonly adopted approach to characterize and calibrate the sensors' errors accordingly in order to improve the measurements' accuracy. Some of these errors are constant while some change in time. Characterization of these changeable components are hard and require long term experiments. The most powerful of them is bias drift [23, 37]. It refers to average measurement of sensors when the input is zero. It depends on operating temperature of the sensors. Operating temperature of the device is determined by environmental factors (e.g., ambient temperature) and initial warm-up of the sensors. Hence, bias drift error generally cannot be compensated as much as other error terms and plays a significant role in the performance of the sensors.

Inertial sensors can be classified into one of the following performance categories [38]: marine, navigation, tactical, industrial, and consumer grades. Inertial navigation systems (INS) having marine grade inertial sensors provide the best accuracy (1.8 km per day) and cost over one million USDs on the average. Navigation grade INSs generally cost 100,000 USDs and have a typical error of 1.5 km per hour. Unaided inertial navigation operation for a few minutes can be achieved by tactical grade inertial sensors. These units typically cost between 5,000–30,000 USDs. Industrial grade INSs typically cost 7,000–10,000 USDs. They can only provide stand-alone inertial navigation solutions for a few seconds. They are typically used in pedestrian dead-reckoning systems, antenna tracking, and low-cost unmanned aerial vehicles. The lowest grade of inertial sensors is the consumer grade. Consumer grade inertial sensors attract the interest of researchers because of their decreasing cost by the developments in MEMS technology. However, they can be used for navigation purposes for a short period of time only if they are calibrated precisely. They are relatively inexpensive compared to the other classes. More detailed information on the performance categories of inertial sensors and their quantitative comparison can be found in [38].

In this thesis, we study the calibration problem of two widely used consumer grade MEMS-based tri-axial inertial measurement units (IMU) and compare them in terms of navigation performance: 3DM-GX2 of MicroStrain (U.S.A.) [39] and MTx of Xsens (The Netherlands) [40]. Both sensors are light, small, and depicted in Figure 1.1 [1, 2]. They have three orthogonal accelerometers, gyroscopes, and magnetometers. MTx is also a part of higher-level system Xbus Kit [41] that synchronizes multiple MTx units. General specifications of the units indicating their performance are also given in Table 1.1. More detailed specifications and manufacturers' calibration sheets of both units are provided in Appendix D.

It is obvious in the table that the error becomes considerably large in a few seconds if navigation is performed by any of these units. Hence, proper calibration of the units is needed. After finding the deterministic and stochastic calibration parameters, both standalone and aided (e.g., GPS) performance of the units can be improved. This is the main objective of this thesis. Now, we will provide a brief summary on the prior work.

(a)

MicroStrain	**accelerometer**	**gyroscope**	**magnetometer**
meas. range	$\pm 50\,\mathrm{m/s^2}$	$\pm 1200^\circ/\mathrm{sec}$	$\pm 1.2\,\mathrm{Gauss}$
bias stability	$\pm 0.05\,\mathrm{m/s^2}$	$\pm 0.2^\circ/\mathrm{sec}$	$0.01\,\mathrm{Gauss}$
nonlinearity	0.2%	0.2%	0.4%
max. data rate	1000 Hz		

(b)

Xsens	**accelerometer**	**gyroscope**	**magnetometer**
meas. range	$\pm 50\,\mathrm{m/s^2}$	$\pm 1200^\circ/\mathrm{sec}$	$\pm 0.75\,\mathrm{Gauss}$
bias stability	$\pm 0.02\,\mathrm{m/s^2}$	$\pm 1^\circ/\mathrm{sec}$	$0.001\,\mathrm{Gauss}$
nonlinearity	0.1%	0.2%	0.2%
max. data rate	512 Hz		

Table 1.1: General specifications of (a) MicroStrain 3DM-GX2 and (b) Xsens MTx units.

1.2 Earlier Work on Deterministic Calibration

When it comes to unaided operation of an INS, deterministic error calibration is a must, especially for systems having MEMS-based inertial sensors [42, 43, 44, 45, 46]. Accordingly, deterministic error terms need to be identified. Methods used for deterministic error model identification of inertial sensors can be classified into two categories: traditional and in-field methods.

Traditional methods are usually adopted in the aerospace industry, where navigation and tactical grade units are used, and satisfactory results are usually obtained. They rely on applying reference signals to the sensors and comparing measurements with the reference dataset. The test procedure changes depending on what type of machine is used to generate the excitation signals. In general, centrifuge and angular position control machines are used for accelerometers. When an angular position control machine (e.g., a flight motion simulator (FMS)) is used, the accelerometer is positioned and held stationary at reference orientations and calibration parameters are calculated based on sensor measurements and reference accelerations associated with the reference orientations and gravitational acceleration [3]. The limitation of this procedure is that acceleration

signals applied to the sensors lie between $-1g$ and $+1g$. This may cause inaccurate modeling outside the $[-1, +1]g$ interval. This method is called the $1g$ test. An angular position control machine is shown in Figure 1.5. As for the centrifuge machine case, reference acceleration signals are applied to the sensors by rotating the centrifuge [47]. Then, deterministic error terms can be identified by the same signal processing algorithm as used in the former procedure. Furthermore, excitation signals are usually not restricted to the $[-1, +1]g$ interval for this case which means that high acceleration values are sustainable [47]. For illustrative purposes, a centrifuge machine is shown in Figure 1.6.

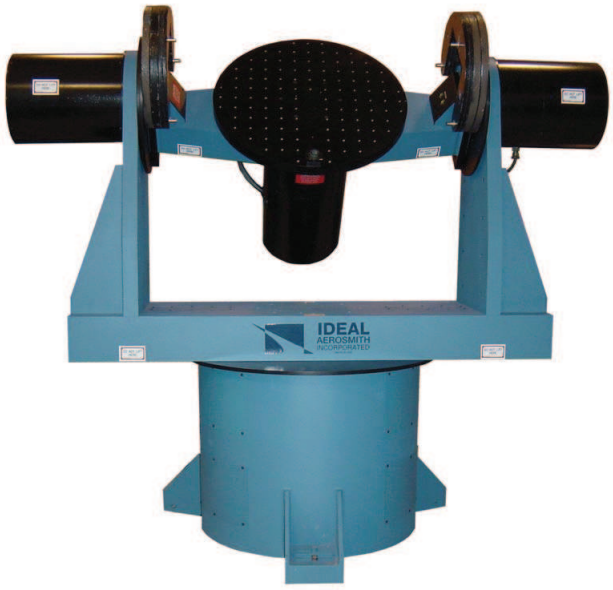


Figure 1.5: An angular position control machine used for inertial sensor calibration [5].

Similar methods are adopted for traditional gyroscope deterministic calibration as well. One of the two commonly employed procedures is based on the positioning gyroscopes at reference orientations and finding the unknown calibration parameters using the sensor measurements and the reference angular rates associated with the reference orientations and the Earth's angular velocity [3] which is analogous to the accelerometer case. However, this is not practical for MEMS-based gyroscopes as they cannot sense the Earth's turn rate [48] and a very limited excitation signal set can be used for deterministic error parameter

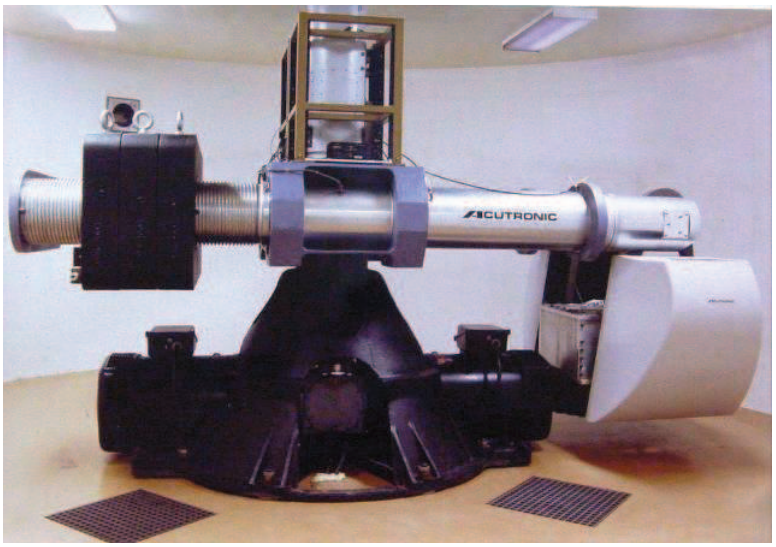


Figure 1.6: A centrifuge machine used for inertial sensor calibration [6].

identification as in the accelerometer case. This procedure can be realized by an angular position control machine. The second procedure is based on rotating gyroscopes at reference turn rates [49] using single axis rate tables. An example single axis rate table is shown in Figure 1.7. Gyroscopes do not experience any considerable calibration error at any turn rate by this method since the dataset used for calibration is not limited to a narrow interval as in the first procedure.

Figure 1.7: A rate table used for gyroscope calibration [7].

Magnetometer experiments need to be carefully designed since external factors, such as the magnetic permeability of the material of the test bed and the external magnetic sources such as electric motors, affect the magnetometer measurements [50, 51]. These effects are usually constant for a certain environment. Hence, magnetometer calibration parameters need to be estimated specific to the real application platform. The main principle of magnetometer error identification is holding magnetometers at known orientations and comparing their measurements with the Earth's true magnetic field at where experiments are conducted [50]. For example, magnetometers used in aircraft are usually calibrated by making certain movements, known as swinging, after being mounted on the aircraft [52, 53].

There has been more effort put on intelligent calibration procedures with lower cost especially after the development of MEMS inertial sensors since traditional methods depend on highly expensive and special machines. It would be senseless that customers, having a low budget and using these low-cost sensors, have to buy such expensive machines for deterministic calibration. Hence, intelligent and inexpensive calibration procedures called in-field calibration methods have been developed [54, 55, 56, 57] since the error terms are identified on the field and generally during the application. In-field calibration techniques have become more popular among researchers since low-cost inertial sensor design has accelerated and has been drawing a lot of attention in the industry. The working principle of the in-field calibration techniques depend on some facts or constraints specific to the application. One example is that the norm of the ideal accelerometer measurements has to be equal to the gravitational acceleration when the sensors are stationary.

1.3 Earlier Work on Stochastic Calibration

Accurate navigation performance cannot be achieved by the standalone utilization of inertial sensors even if deterministic calibration is done perfectly, which is not likely in practice due to repeatability issues [48]. The stochastic nature of the

measurements restricts the success of INSs. Inertial sensors are typically used in conjunction with other sensing systems, such as Global Positioning System (GPS) [58, 59], laser [60], odometry [61] for that reason. GPS is the most commonly used aid sensor among them [62]. GPSs can provide more accurate navigation information but they are not sufficient on their own due to their low update rate and frail/delicate structure [63] (i.e., blocking of satellite signals) in systems requiring real-time navigation. Furthermore, they become inoperable indoors. For these reasons, localization systems combining an IMU and GPS are favored. Such systems bring together the advantage of the IMU's real-time navigation capability and the accuracy of the GPS at low frequencies.

Kalman filter and its variations are often used in integrated INS/GPS navigation systems [64, 65, 66]. Therefore, the process and measurement noise models need to be constructed to ensure that the fusion algorithm works properly. Stochastic modeling and calibration of inertial sensors is an important step of this work.

Here, we provide a brief survey and review of the prior work on stochastic modeling and calibration of inertial sensors and magnetometers. Approaches toward identification of stochastic model parameters of inertial sensors are mainly focused on the Allan variance (AV), which is a type of statistical analysis tool. It has been adopted firstly by time and frequency standards community for the characterization of frequency instability of oscillators [67]. Using the AV for stochastic identification of inertial sensors is mainly based on fitting the theoretical AV of the noise content in the measurements of inertial sensors to the actual AV obtained through experiments. Detailed discussion on the theoretical AV of the noise processes in inertial sensor output can be found in [68, 69]. The first work in which AV is used for the stochastic analysis of inertial sensors is [70]. Then, the AV has become a very popular stochastic parameter estimation tool for inertial sensors and it has been acknowledged to be a standard method for stochastic calibration of inertial sensors by the IEEE [37, 10]. The main problem of the AV method is its limited accuracy. Some of the noise terms in the inertial sensor outputs have slow dynamics [71] so lengthy datasets may be needed to involve them in the stochastic model depending on the operation time requirement

of the INS. However, an exceedingly long dataset has to be acquired to make fair estimates due to the accuracy issue of the AV. This increases both the duration of experiments, storage requirements of the observations, and the computational cost. In this regard, some improvements to the AV have been proposed in the literature and novel forms of AV using the dataset more efficiently are suggested: modified AV [72] and overlapping AV [73]. However, they mostly suffer from computational time issue. Comparing the price paid for the computational time and the improvement obtained by these enhanced AV techniques lead people to prefer standard AV method. A general estimation scheme of the parameters by AV is presented in [74]. Although AV is the most widely used technique, adaptive methods are also studied in the literature and used sometimes. Online techniques such as adaptive Kalman filtering are commonly employed since simultaneous sensor fusion and stochastic parameter estimation can be performed [66, 75, 76, 77]. Besides modeling measurement noise and including it in a fusion filter, another approach is to remove the noise in the measurements. In references [78, 79], wavelet analysis is used for this purpose and it is shown that substantial improvements are obtained in both standalone and aided navigation solutions.

As seen from the literature survey, a limited number methods has been considered in the literature for the stochastic identification of the noise parameters of inertial sensors. The underlying reason might be that most inertial sensors had not required detailed calibration because the sensors had already been highly accurate before MEMS inertial sensors were developed, and scientists had sticked to the reliable and traditional methods. Furthermore, there is still a lack of advanced methods meeting the stochastic identification needs of MEMS inertial sensors. Hence, implementation of novel methods in stochastic model identification is worth trying because of their success in other application areas.

1.4 Contributions and Organization of the Thesis

Regarding the deficiencies of deterministic and stochastic error identification methods, we propose some novel techniques to improve their results. Our main contributions in this thesis can be summarized as follows:

- We propose a novel in-field calibration algorithm, which enables the calibration of MEMS gyroscopes by simple and hand-made rotations. This relaxes the special machinery requirement of the gyroscope calibration problem. The algorithm is based on the attitude of the sensors (e.g., one complete revolution of the gyroscope about one of its mechanical case axes made by using the hands) and makes use of the particle swarm optimization (PSO) technique since it would be pretty hard to set up a derivative-based optimization algorithm. At the end of the calibration, minor residual errors are achieved. This demonstrates the practical potential of the proposed algorithm.

- We adopt the general approach (traditional $1g$ test) for accelerometer deterministic error identification. Effectiveness of the method is shown through experiments.

- To the best of our knowledge, a method maximizing the both exact and approximate likelihood functions after deriving their expressions for the noise terms in inertial and magnetic sensors' outputs does not exist. In this thesis, both exact and approximate maximum likelihood estimation (MLE) schemes are derived for stochastic identification after a statistically equivalent autoregressive-moving average (ARMA) noise process is developed. We use two distinct algorithms for the maximization of the likelihood function: gradient-ascent optimization (GAO) and PSO. It is proven by experiments that much better results in terms of accuracy, consistency, and time consumption are obtained with these approaches compared to the classical methods (i.e., AV).

Before proceeding to the detailed discussion of the topics, a brief outline of the thesis is given: In Chapter 2, we begin with developing the deterministic sensor model and then provide the experimental results of our proposed algorithms. In Chapter 3, we first develop the necessary framework for MLE by providing an ARMA model regarding the noise content of the inertial sensors and magnetometers. We then compare the experimental results of the MLE technique employing two different optimization algorithms and the traditional AV technique in terms of processing time, accuracy, and consistency. In Chapter 4, we compare our two IMUs in terms of measurement quality regarding the results of deterministic and stochastic identification and make the concluding remarks. We also provide directions for future research in the same chapter.

Chapter 2

Deterministic Modeling and Calibration

To compensate for the localization errors originating from the drifting behavior of inertial sensors, a sensor error model is built. In this context, the general measurement model of an inertial sensor can be expressed as

$$\vec{e}_m = o\left(\vec{e}_t\right) + \vec{v}_m, \tag{2.1}$$

where $\vec{e}_m, \vec{e}_t, \vec{v}_m \in \mathbb{R}^3$ denote the output of the sensor, the true value of the excitation signal, and the stochastic noise, respectively. Moreover, $o(.) : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is a general functional operator modeling the behavior of inertial sensors. Magnetometers are modeled differently since there exist some additional factors affecting their output.

The parameters involved in the above model need to be estimated accurately. Most of the previous works [3, 37, 80] approach the calibration problem by separating the problem into two distinct parts as deterministic and stochastic model identification because of their different mathematical characteristics. In this thesis, we follow the same approach and consider deterministic and stochastic modeling separately.

Before moving on to a detailed description of the study, the notation used

throughout the thesis is described: $\vec{a}_m$, $\vec{\omega}_m$ and $\vec{h}_m$ denote the accelerometer, gyroscope, and magnetometer measurement vectors, respectively. The $\vec{a}_t$, $\vec{\omega}_t$, and $\vec{h}_t$ denote the true specific acceleration, angular rate, and magnetic field strength vectors. Any vector $\vec{\nu}$ expressed in the frame $\mathcal{F}$ is denoted by $\vec{\nu}^{\mathcal{F}}$ and the direction cosine matrix (DCM) between any two frames $\mathcal{F}_1$ and $\mathcal{F}_2$ is denoted by $\boldsymbol{C}_{\boldsymbol{\mathcal{F}_1}}^{\boldsymbol{\mathcal{F}_2}}$ where $\vec{\nu}^{\mathcal{F}_2} = \boldsymbol{C}_{\boldsymbol{\mathcal{F}_1}}^{\boldsymbol{\mathcal{F}_2}} \vec{\nu}^{\mathcal{F}_1}$. Orthonormal basis vectors of the x, y, and z axes of any frame $\mathcal{F}$ are respectively shown by $\vec{i}^{\mathcal{F}}, \vec{j}^{\mathcal{F}},$ and $\vec{k}^{\mathcal{F}}$. Furthermore, a_x, a_y, a_z, g_x, g_y, g_z and m_x, m_y, m_z abbreviations found in the tables in this thesis represent the x, y, z axes accelerometers, gyroscopes, and magnetometers, respectively.

2.1 A Deterministic Model for Accelerometers and Gyroscopes

Accelerometer and gyroscope outputs can be sufficiently modeled using polynomials [37]. In most practical applications, 2nd and higher-order terms are neglected. In this regard, the following equation is used to model $o(.)$ in Equation (2.1) for accelerometers and gyroscopes:

$$\vec{e}_m = (\boldsymbol{I} + \boldsymbol{S})\vec{e}_t + \vec{b} + \vec{v}_m \quad \text{where} \quad \boldsymbol{S} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix} \quad \vec{b} = \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix} \tag{2.2}$$

Here $\vec{b}$ and $\boldsymbol{S}$ respectively denote the bias vector and the scale factor error matrix. $\boldsymbol{I}$ is a 3×3 identity matrix.

In general, sensitivity axes of inertial sensors are often not coincident with the axes of the body whose motion they are supposed to detect. Therefore, the transformation between those two sets of axes needs to be determined beforehand. Otherwise, the sensor model, given above, would be insufficient for calibration. For this purpose, we define several sets of axes:

- **Non-orthogonal sensor sensitivity frame ($\hat{s}$ frame):** This frame represents the set of actual sensitivity axes of the sensor. Deviation from

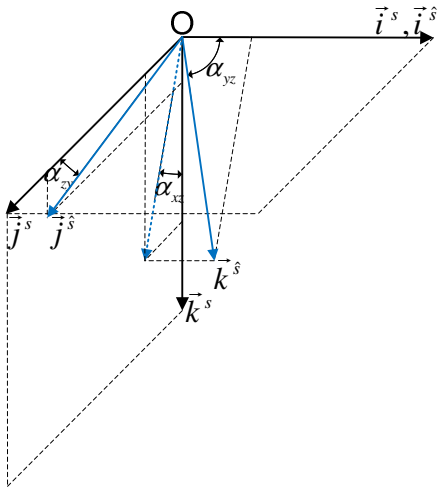


Figure 2.1: The s and $\hat{s}$ frames.

orthogonality stems from manufacturing tolerances in general. Its effects on navigation performance are not trivial.

- **Orthogonal sensor sensitivity frame (s frame):** This frame is the orthogonalized version of the $\hat{s}$ frame. Without loss of generality, components of this frame can be described as follows, as illustrated in Figure 2.1:
  - $\vec{i}^{s}$ is coincident with $\vec{i}^{\hat{s}}$.
  - $\vec{j}^{s}$ lies along the remaining perpendicular component of $\vec{j}^{\hat{s}}$ after its projection onto $\vec{i}^{s}$.
  - $\vec{k}_s$ is on the same direction as the component of $\vec{k}_{\hat{s}}$ perpendicular to the plane spanned by $\vec{i}^{s}$ and $\vec{j}^{s}$.

  The kinematic transformation between the frames $\hat{s}$ and s is given below, where $\vec{\nu}^{\hat{s}} = \boldsymbol{T}\vec{\nu}^{s}$:

$$\boldsymbol{T} = \begin{bmatrix} 1 & 0 & 0 \\ \sin(\alpha_{zy}) & \cos(\alpha_{zy}) & 0 \\ \cos(\alpha_{yz}) & \sin(\alpha_{yz})\sin(\alpha_{xz}) & \sin(\alpha_{yz})\cos(\alpha_{xz}) \end{bmatrix} \tag{2.3}$$

- **Sensor enclosure frame (p frame):** This frame is made up of the orthogonal axes system of the sensor mechanical casing. Due to manufacturing

tolerances and packaging issues, it cannot be aligned with the s frame in practice. This situation can be represented by the DCM corresponding to a series of rotations about the axes as expressed below:

$$\boldsymbol{C_p^s = R_x R_y R_z}, \tag{2.4}$$

where

$$\boldsymbol{R_x} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix}, \quad \boldsymbol{R_y} = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix},$$

$$\boldsymbol{R_z} = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

are the basic rotation matrices about the x, y, z axes, respectively.

- **Body frame ($\boldsymbol{b}$ frame):** This frame is comprised of the orthogonal axes of the platform to which the inertial sensors are attached. This frame should be known so that the body can be navigated.

After all of the deterministic factors mentioned above are considered, the resulting form of the sensor measurement model can be expressed as follows:

$$\vec{e}_m = \boldsymbol{(I + S)TC_p^s C_b^p}\,\vec{e}_t + \vec{b} + \vec{v}_m, \tag{2.5}$$

where $\vec{e}_m$ can be replaced with either $\vec{a}_m$ or $\vec{\omega}_m$, while $\vec{e}_t$ can be replaced with either $\vec{a}_t$ or $\vec{\omega}_t$.

The error terms in Equation (2.5), which is the sensor measurement model for accelerometers and gyroscopes, can be summarized as follows:

- $\boldsymbol{S}$ is the scale factor error matrix and represents the measurement error of the sensors in proportion to the input signal.
- $\boldsymbol{T}$ is the non-orthogonalization matrix related to frame $\hat{s}$. It is alternatively known as the cross-axis sensitivity matrix.

- $\boldsymbol{C_p^s}$ is the misalignment matrix and represents the imperfect alignment of the frames p and s.
- $\boldsymbol{C_b^p}$ is the transformation matrix between the frames b and p.
- $\vec{b}$ is the bias error vector and represents the constant measurement errors on all axes. The bias errors usually change with the operating temperature of the sensor.
- $\vec{v}_m$ is the additive measurement noise vector.

2.2 A Deterministic Model For Magnetometers

As stated before, magnetometers are used to find the attitude of frame b with respect to the Earth's frame of reference by measuring the Earth's magnetic field vector, denoted by $\vec{h}_e^{ned}$. The so-called North-East-Down (NED) frame is selected for this purpose which is depicted in Figure 2.2.

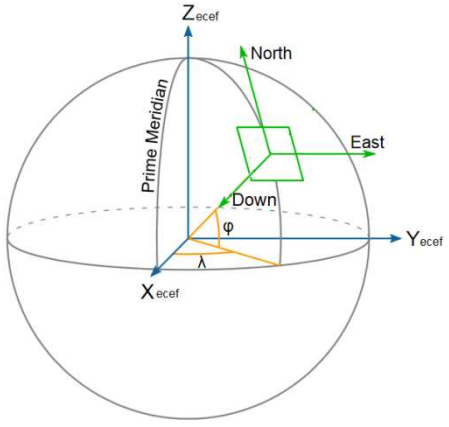


Figure 2.2: The frame *ned* with its basis vectors (adopted from [8]).

We use the superscript *ned* for the NED frame. The unit vectors $\vec{i}^{ned}$, $\vec{j}^{ned}$, and $\vec{k}^{ned}$ lie respectively along the north, east, and down directions. In the real world, magnetometers are not solely exposed to $\vec{h}_e^{ned}$ as expected. Particularly, this problem occurs in indoor environments where there may be other external impacts changing the magnetic field vector, measured by the sensor. Those effects

strongly depend on the presence of ferromagnetic materials in the vicinity of the sensor and the presence of sources that radiate magnetic fields. Errors are grouped into two types as soft and hard iron errors [51]. A more detailed discussion on these error types is given below:

- **Hard iron error ($\boldsymbol{\delta\vec{B}}$):** They are defined as the time-invariant, unwanted magnetic fields generated by the ferromagnetic materials with permanent magnetic fields that are part of the structure of the platform on which the sensors are placed or equipment installed near the magnetometer [81]. The resultant magnetic field is the superposition of $\vec{h}_e^{ned}$ and $\delta\vec{B}$. $\delta\vec{B}$ can be represented by the following vector:

$$\delta\vec{B} = \begin{bmatrix} \delta_x & \delta_y & \delta_z \end{bmatrix}^T \tag{2.6}$$

- **Soft iron error ($\boldsymbol{K_{si}}$):** They are introduced into the system by the interaction of the external magnetic field with the ferromagnetic materials in the vicinity of the sensor [51]. Magnetic permeability of the materials has a direct influence on this interaction. $\boldsymbol{K_{si}}$ can be represented by the following symmetric matrix:

$$\boldsymbol{K_{si}} = \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{13} & k_{23} & k_{33} \end{bmatrix} \tag{2.7}$$

With the addition of $\boldsymbol{K_{si}}$, $\delta\vec{B}$, and the transformation matrix $\boldsymbol{C_{ned}^b}$ projecting $\vec{h}_e^{ned}$ onto the b frame, Equation (2.5) is extended and the magnetometer measurements can be modeled by the following equation [51, 53]:

$$\vec{h}_m = \boldsymbol{(I+S)T}(\boldsymbol{K_{si}C_p^sC_b^pC_{ned}^b}\,\vec{h}_e^{ned} + \delta\vec{B}) + \vec{b} + \vec{v}_m \tag{2.8}$$

2.3 Deterministic Calibration of the Sensors

The navigation errors tend to accumulate very rapidly and consequently the output drifts in time (i.e., proportionate with time cube for translational position

and time square for angular position) [64]. Drift errors are more enhanced especially when consumer grade MEMS type of inertial sensors are used. Thus, precise calibration of deterministic model parameters is essential.

It can be seen from Equations (2.5) and (2.8) that the sensor dynamics for inertial sensors and magnetometers have some similarity. Therefore, similar techniques can be used for the calibration of these sensors. The most straightforward method is to apply a specific set of reference signals and observe the corresponding outputs of the sensors. By comparing the observations and the reference data set, calibration parameters are estimated. We give a brief description of this approach for the three sensor types below.

- **accelerometer:** The traditional method for accelerometers is known as both multi-position calibration and the $1g$ test [3]. Accelerometers are held stationary at different and known orientations throughout this test. As a result, calibration is performed according to the sensor measurements and the local gravity vector, denoted by $\vec{g}_L$ [59, 66, 67, 70].

- **gyroscope:** The calibration of general purpose and inexpensive gyroscopes (i.e., MEMS gyroscopes) requires the application of different angular velocities whereas high-precision gyroscopes (i.e., fiber optic gyroscopes), that are capable of measuring the Earth's turn rate, can be calibrated by the multi-position method. Calibration parameters are determined by processing the gyroscope measurements with respect to the applied angular rate values [66, 67, 68, 69, 70, 74].

- **magnetometer:** Magnetometers are positioned at known orientations in a similar way to accelerometers. Unknown model parameters can be estimated by comparing the magnetometer measurements and the magnetic field vector $\vec{h}_e^n$ at the point where the experiments are conducted [51, 53, 82, 83].

When traditional approaches are utilized, a machine controlling the angular position and velocity is needed for the calibration of these sensors. The precision of the machine in position and velocity control directly affects the estimation accuracy

of the sensor model parameters and the related cost increases proportionately. This cost has motivated researchers to develop in-field calibration methods that do not require any external equipment. For accelerometers and magnetometers, in-field calibration methods rely on the fact that the magnitude of the input signal is equal to the magnitude of $\vec{g}_L$ and $\vec{h}_e$ under the condition that the sensors are stationary. On the other hand, more advanced techniques are needed for gyroscope calibration that are usually based on comparing the computed attitude with the true attitude obtained by simple, hand-made rotations [55, 56, 57].

Besides the calibration test procedures, another challenge for deterministic model identification is to develop robust and accurate parameter estimation algorithms. Various techniques have been studied in the literature. Simplicity of the estimation algorithm greatly depends on the complexity of the sensor model. Thus, batch least-squares like fundamental linear methods are adopted when orthogonality and misalignment errors are ignored (soft and hard iron errors are also assumed to be zero for magnetometers) and measurement equations reduce to a linear system of equations [57, 84, 85]. In [86], rank constraints of the linear system of equations are exploited for parameter estimation. However, it is essential to use more complex algorithms that consider the nonlinearities in the sensor dynamics. Otherwise, the measurement errors cannot be adequately compensated for. In this regard, ellipsoid parameter estimation techniques are used quite extensively since both Equations (2.5) and (2.8) are a kind of ellipsoid analytical expressions. These techniques are divided into two categories as geometric and algebraic fit methods [87] and are based on different aspects of the calibration models. Regarding Sections 2.1 and 2.2, $\vec{e}_m$ corresponding to a specific time instant k can be expressed in general terms as

$$\vec{e}_m[k] = \boldsymbol{H}\vec{e}_t[k] + \hat{\vec{b}} + \vec{v}_m. \tag{2.9}$$

Since Equation (2.9) defines an ellipsoid, ellipsoid parameter estimation techniques can be employed. For the general sensor model given in the above equation, the two approaches can be summarized as:

- Geometric techniques try to find the calibration parameters by

$$\arg\min_{\boldsymbol{H},\hat{\vec{b}}} \sum_{k} \parallel \vec{e}_m[k] - \boldsymbol{H}\vec{e}_t[k] - \hat{\vec{b}} \parallel . \tag{2.10}$$

- Algebraic techniques rely on a different model, arranged form of Equation (2.9), and try to estimate the calibration parameters by

$$\arg\min_{\boldsymbol{H},\hat{\vec{b}}} \sum_{k} \left[(\vec{e}_m[k] - \hat{\vec{b}})^T (\boldsymbol{H^{-1}})^{\boldsymbol{T}} \boldsymbol{H^{-1}} (\vec{e}_m[k] - \hat{\vec{b}}) - \vec{e}_t[k]^T \vec{e}_t[k] \right]^2 . \tag{2.11}$$

It is shown in [87] that geometric techniques are superior to algebraic techniques in terms of fitness accuracy but require exact knowledge of $\vec{e}_t$, not available in some cases. References [51, 82, 88] report on some of the works on sensor calibration that employ ellipsoid parameter estimation methods.

In this thesis, we have used Acutronic's high precision, three-degree-of-freedom flight motion simulator (FMS) to conduct deterministic calibration experiments for our sensors. In Table 2.1, technical specifications of the FMS can be found. Furthermore, the FMS and its rotation axes are illustrated in Figure 2.3.

For calibration purposes, both MicroStrain and Xsens IMUs are mounted to the fixture plate of the FMS, located on the shaft of the inner axis, at the same time. This is illustrated in Figure 2.4. Then, a trajectory of the axes of the FMS is determined for the experiments, which is called a calibration procedure. The calibration procedure, loaded into the FMS controller computer after programming, is summarized below:

1. The inner axis of the FMS is aligned with the level (ground) as shown in Figure 2.5.

2. The inner axis of the FMS is rotated by 270° in 12 steps. FMS is held stationary at each of those steps for 5 seconds.

3. The inner axis of the FMS is aligned with the gravity vector $\vec{g}_L$ as shown in Figure 2.6. It is assumed in this thesis that $\vec{g}_L$ points perpendicular to the level. This is explained in Section 2.3.1.

	roll (inner axis)	**pitch (middle axis)**	**yaw (outer axis)**
orthogonality (arcsec)	–	5	5
wobble (arcsec)	2	3	5
angular freedom	continuous	continuous	continuous
positioning accuracy (arcsec)	1	1.5	1.5
rate range (deg/sec)	±1000	±500	±300
rate resolution (deg/sec)	0.00001	0.00001	0.00001
rate accuracy (%)	0.0001	0.0001	0.0001
acceleration (deg/sec^2)	10000	1500	400
bandwith (−3 dB) (Hz)	50	22	30

Table 2.1: Specifications of the FMS [11].

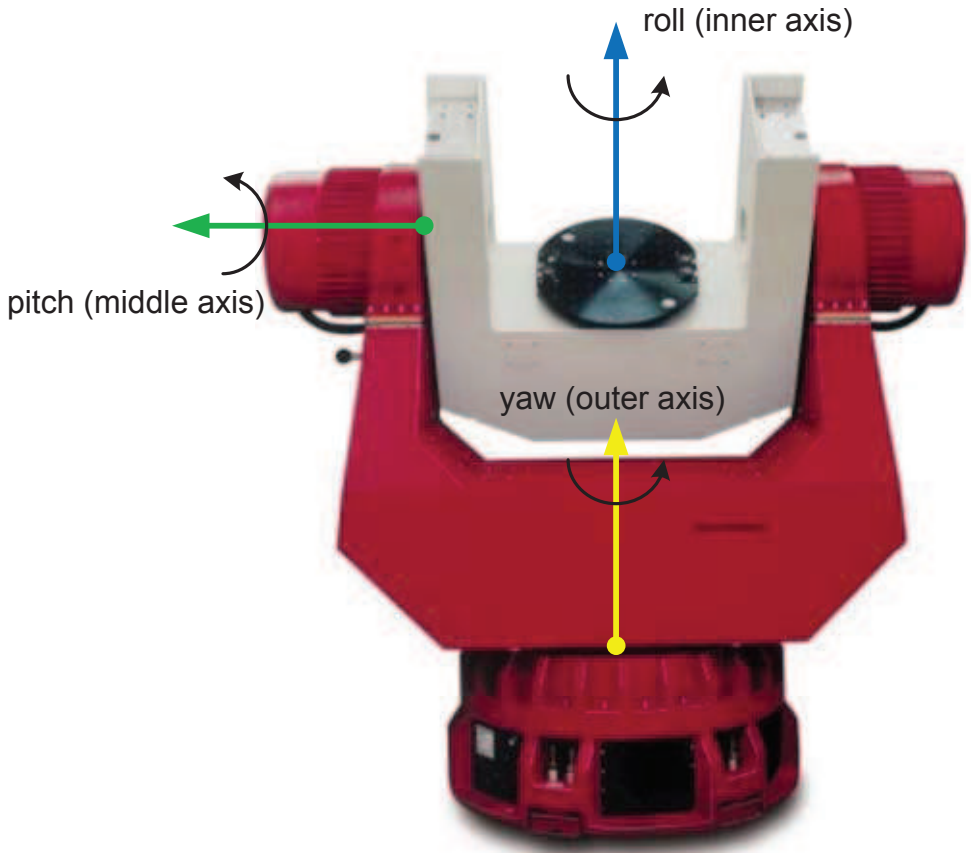
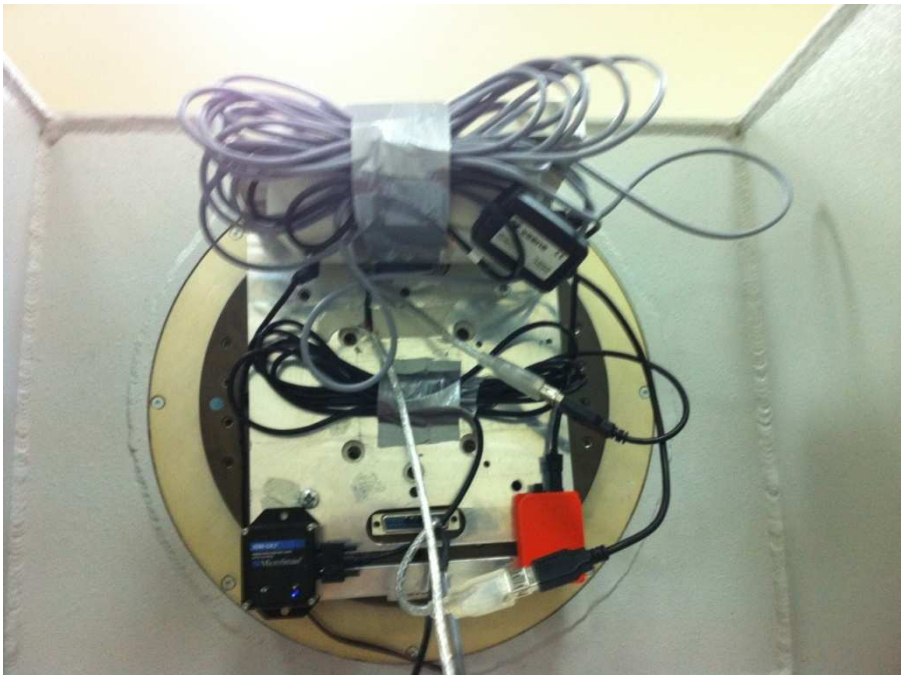


Figure 2.3: Acutronic FMS overview (adopted from [9]).

(a) overview



(b) close-up

Figure 2.4: (a) Overview and (b) close-up views of fixture plate onto which MicroStrain and Xsens IMUs are mounted.

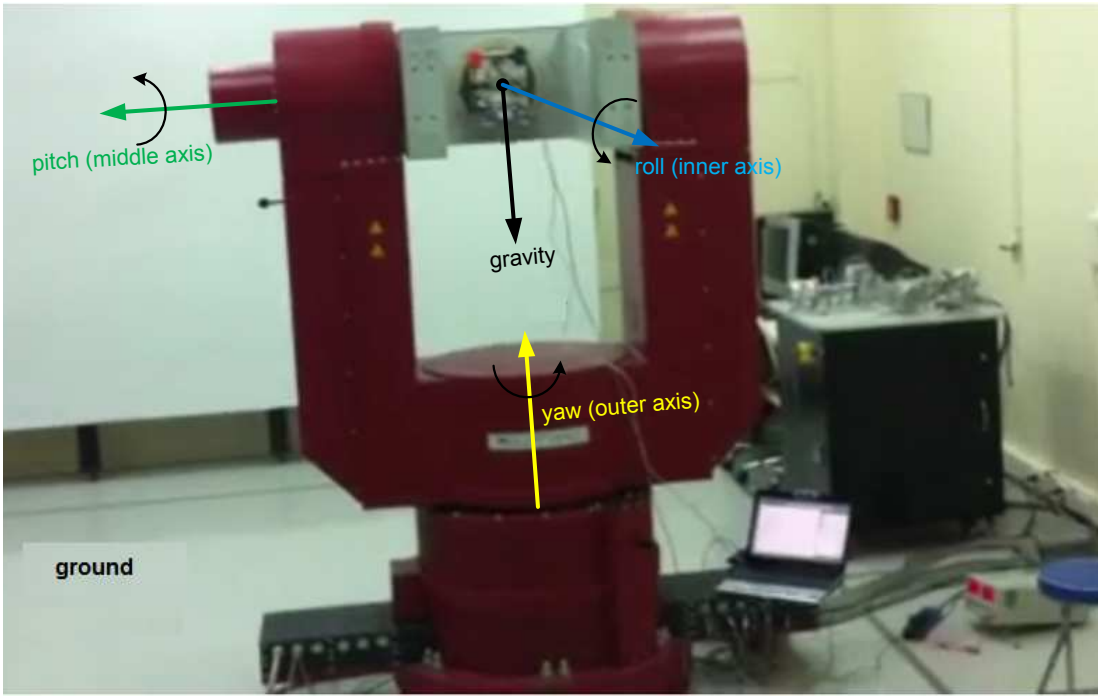


Figure 2.5: FMS at calibration procedure step 1.

4. The FMS makes a half turn around its middle axis while it stops and waits for 5 seconds at each step of 22.5°.

5. The FMS is taken back to its angular position at step 3.

6. The inner axis of the FMS is rotated by 90°.

7. TheFMS performs the same motion as in step 4.

The main concern while designing this scenario is to ensure that the accelerometers and magnetometers experience a complete reference signal set for calibration. The acceleration values of both IMUs are illustrated together in Figure 2.7. It can also be noted that this procedure is a type of multi-position calibration method and calibration of gyroscopes using the same approach cannot simply be realized because the angular rates of the FMS axes are unknown and the Earth's turn rate cannot be sensed by our low-cost consumer grade gyroscopes.

During the calibration tests, accelerometer, gyroscope, and magnetometer outputs of both MicroStrain and Xsens units are simultaneously recorded at a

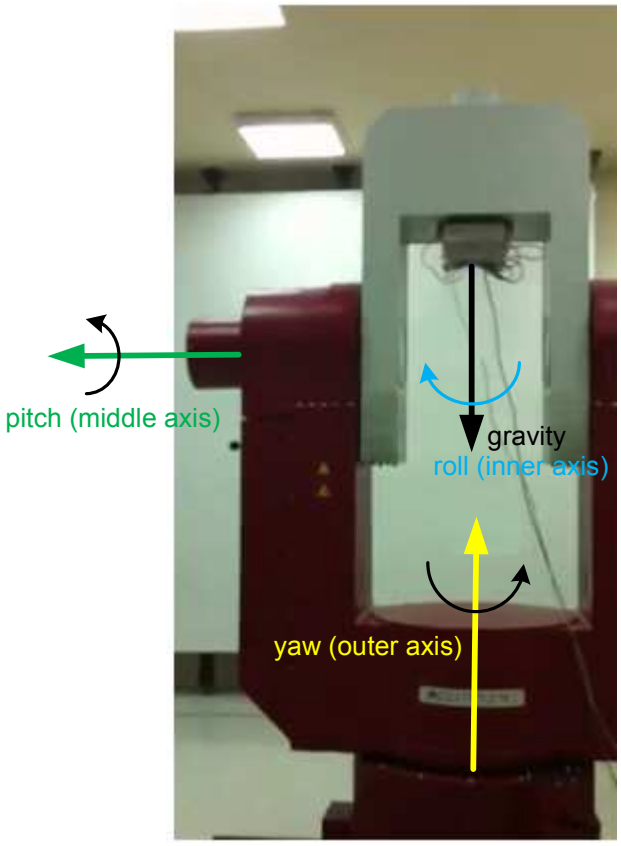
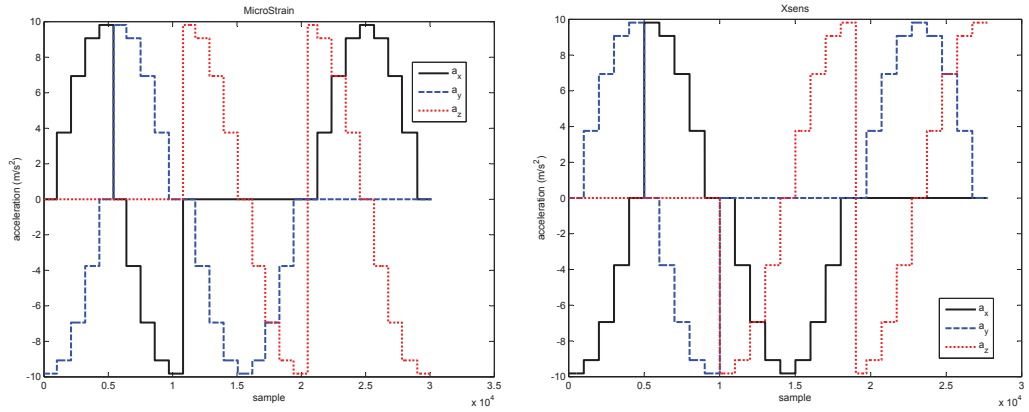


Figure 2.6: FMS at calibration procedure step 3.

sampling rate of 100 Hz. Since only the orientations of the sensors' mechanical enclosures with respect to the FMS frame, represented by the transformation $\boldsymbol{C_b^p}$, when the angular rate of the FMS is zero, are known, the subset of all the measurements that belong to those moments is kept while the rest are discarded. This subset is chronologically rearranged and time indices in the subset are renumbered as a consecutive array. The number of samples in the final form of the measurements is denoted by N. The exact mathematical relation between the frames *ned* and b, shown on the FMS in Figure 2.8, is unknown but its structure is known so that $\boldsymbol{C_{ned}^b}$ can be represented in the parametric way as given in Equation (2.12). Furthermore, $\vec{g}_L^{ned}$ and $\vec{h}_e^{ned}$ are known since the location and orientation with respect to the level of the facility, at where the experiments have been performed, is known.



(a) (b)

Figure 2.7: Signal sets applied to (a) MicroStrain and (b) Xsens accelerometers in frame b.

$$\boldsymbol{C_b^{ned}} = \begin{bmatrix} \cos\Psi & \sin\Psi & 0 \\ -\sin\Psi & \cos\Psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.12}$$

Nonlinear optimization techniques can be used to estimate the model parameters of accelerometers and magnetometers by minimizing the error between the actual and estimated sensor measurements according to Equations (2.5) and (2.8) since $\vec{g}_L^{ned}$ and $\vec{h}_e^{ned}$ are known. We use the Levenberg-Marquardt algorithm (LMA) [89] for this purpose. Background information on the LMA is provided in Appendix A. On the other hand, the true angular rates are not available as mentioned before. This prevents adopting the same algorithm for gyroscope calibration. Therefore, the error of the estimated angular position rather than the angular rate is selected as the performance criterion.

An analytical relation between the actual and computed orientations should be developed to use the LMA or a similar optimization algorithm for estimating the gyroscope calibration parameters. However, since the derivation of this error model is not an easy task, to relax this difficulty we use a model-free calibration algorithm. Evolutionary optimization algorithms satisfy this requirement. Due to its implementation simplicity and known success, we have employed PSO for the identification of gyroscope model parameters.

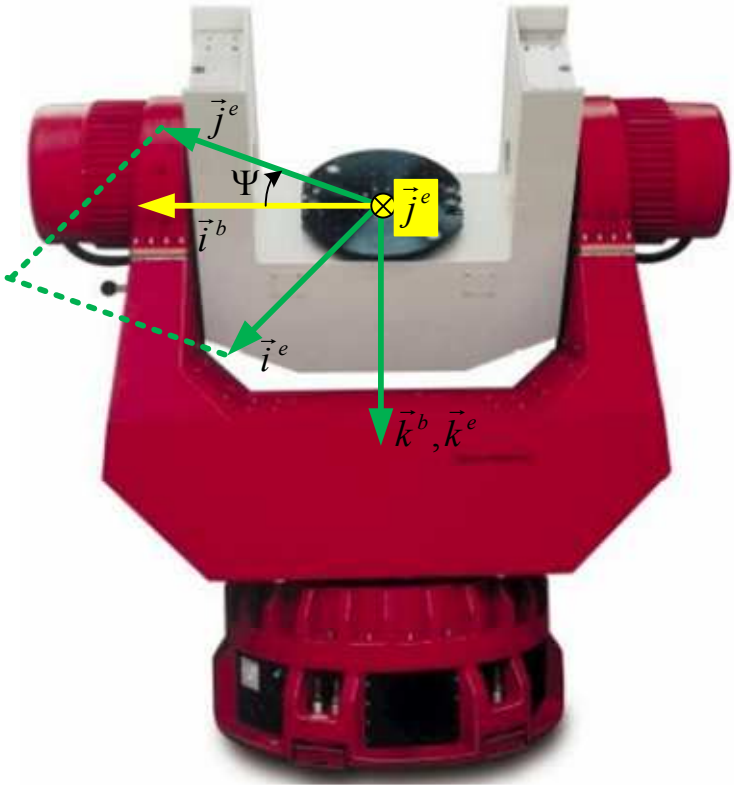


Figure 2.8: The configuration of *ned* and b frames (adopted from [9]).

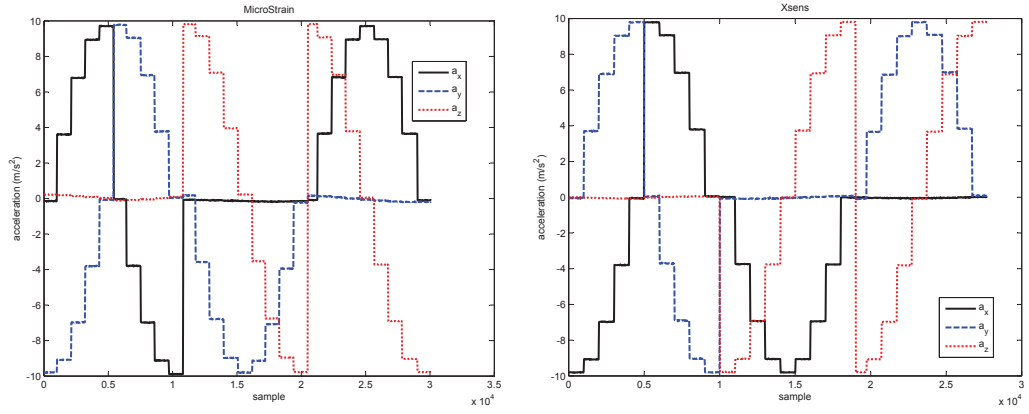
2.3.1 Accelerometer Calibration

The following definitions are needed to ensure that the LMA operates properly for accelerometer calibration.

- The vector $\vec{y}$ comprises of accelerometer measurements. For our tri-axial accelerometers, it consists of a total of $3N$ elements.

$$\vec{y} = \begin{bmatrix} \vec{a}_m^T[1] & \vec{a}_m^T[2] & \cdots & \vec{a}_m^T[N] \end{bmatrix}^T \tag{2.13}$$

  The variable $\vec{a}_m[k] \;\; \forall\, k \in N$ denotes the output vector of accelerometers at time step k. Measurement sets of MicroStrain and Xsens IMUs are depicted in Figure 2.9.



(a) (b)

Figure 2.9: Measurements of (a) MicroStrain and (b) Xsens accelerometers in frame b.

- $f(.)$ is determined as

$$f(\vec{\theta}) = \begin{bmatrix} o\left(\boldsymbol{C}_{\boldsymbol{b}}^{\boldsymbol{p}}[1]\vec{g}_L^b\right)^T & o\left(\boldsymbol{C}_{\boldsymbol{b}}^{\boldsymbol{p}}[2]\vec{g}_L^b\right)^T & \cdots & o\left(\boldsymbol{C}_{\boldsymbol{b}}^{\boldsymbol{p}}[N]\vec{g}_L^b\right)^T \end{bmatrix}^T, \tag{2.14}$$

  where $o(.)$ is the sensor model of accelerometers and gyroscopes given in Equation (2.5), and $\boldsymbol{C}_{\boldsymbol{b}}^{\boldsymbol{p}}[k]$ represents the $\boldsymbol{C}_{\boldsymbol{b}}^{\boldsymbol{p}}$ at time instant k.

- In accordance with $f(.)$, $\vec{\theta}$ is determined as

$$\vec{\theta} = \begin{bmatrix} b_x & b_y & b_z & s_x & s_y & s_z & \phi & \theta & \psi & \alpha_{zy} & \alpha_{yz} & \alpha_{xz} \end{bmatrix}^T \tag{2.15}$$

where the description of its elements can be found in Section 2.1. For the ideal sensor that requires no calibration,

$$\vec{\theta} = \left[\begin{array}{cccccccccccccc} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 90 & 0 \end{array} \right]^T. \tag{2.16}$$

For this special case of $\vec{\theta}$, the right hand side of the equation is denoted by $\vec{\theta}_{w/o}$.

The value of $\vec{g}_L^{ned}$ at the location where the experiments are conducted can be found by [3]:

$$\vec{g}_L^{ned} = \vec{g}_0 - \frac{\| \omega_e^{ned} \|^2 (R_0 + \mathrm{h})}{2} \left[\begin{array}{ccc} \sin 2\Lambda & 0 & (1 + \cos 2\Lambda) \end{array} \right]^T, \tag{2.17}$$

where Λ, h, $\vec{g}_0$, $\vec{\omega}_e^{ned}$, and R_0 represent the latitude angle, altitude, standard gravity vector, the Earth's turn rate vector, and the radius of the Earth, respectively. Furthermore, the $\vec{g}_L^b$ in Equation (2.14) can be approximated by the $\vec{g}_L^{ned}$ after rounding its x-axis component to zero, assuming that our consumer grade accelerometers cannot sense such relatively small values because of the geometrical relation between the frames n and b, shown in Figure 2.8 and given in Equation (2.12), and the structure of $\vec{g}_L^{ned}$. Therefore, the $\vec{g}_L^b$ used in the calibration procedure is

$$\vec{g}_L^b = \left[\begin{array}{ccc} 0 & 0 & 9.8176 \end{array} \right]. \tag{2.18}$$

Before processing the acquired observations with LMA, we present the acceleration errors of the two units in Figure 2.10.

Input parameters of the LMA are selected empirically as follows: $\varpi = 1, \varepsilon_1 = 10^{-10}$, $\varepsilon_2 = 10^{-10}$ and $\vec{\theta}_0$ is initialized randomly (see Appendix A). It is observed that LMA converges to a minimum in about 15 iterations for the MicroStrain unit and 11 iterations for the Xsens units. The calibration parameters obtained at the end of the runs are given in Table 2.2. Using these calibration parameters, considerable improvement is obtained in the model fit and the errors are considerably reduced. These are shown in Figure 2.11 and Table 2.3. In Table 2.3, $f(\vec{\theta}_{w/o})$ is calculated by using that the $\vec{\theta}$ is equal to the right hand side of Equation (2.16) (i.e., the ideal sensor case where no errors of any type are present).

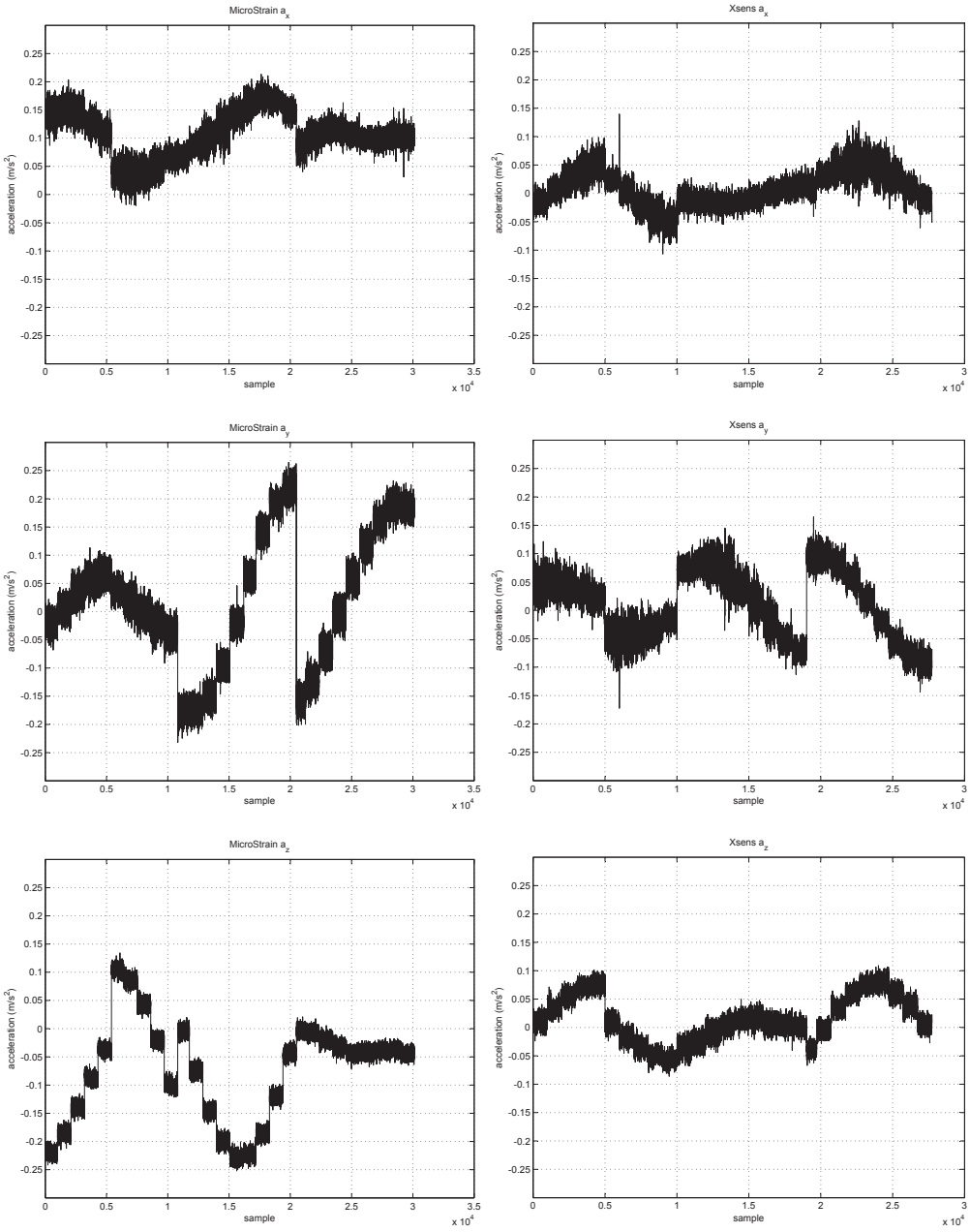


Figure 2.10: Uncalibrated acceleration measurement errors of all axes of both units.

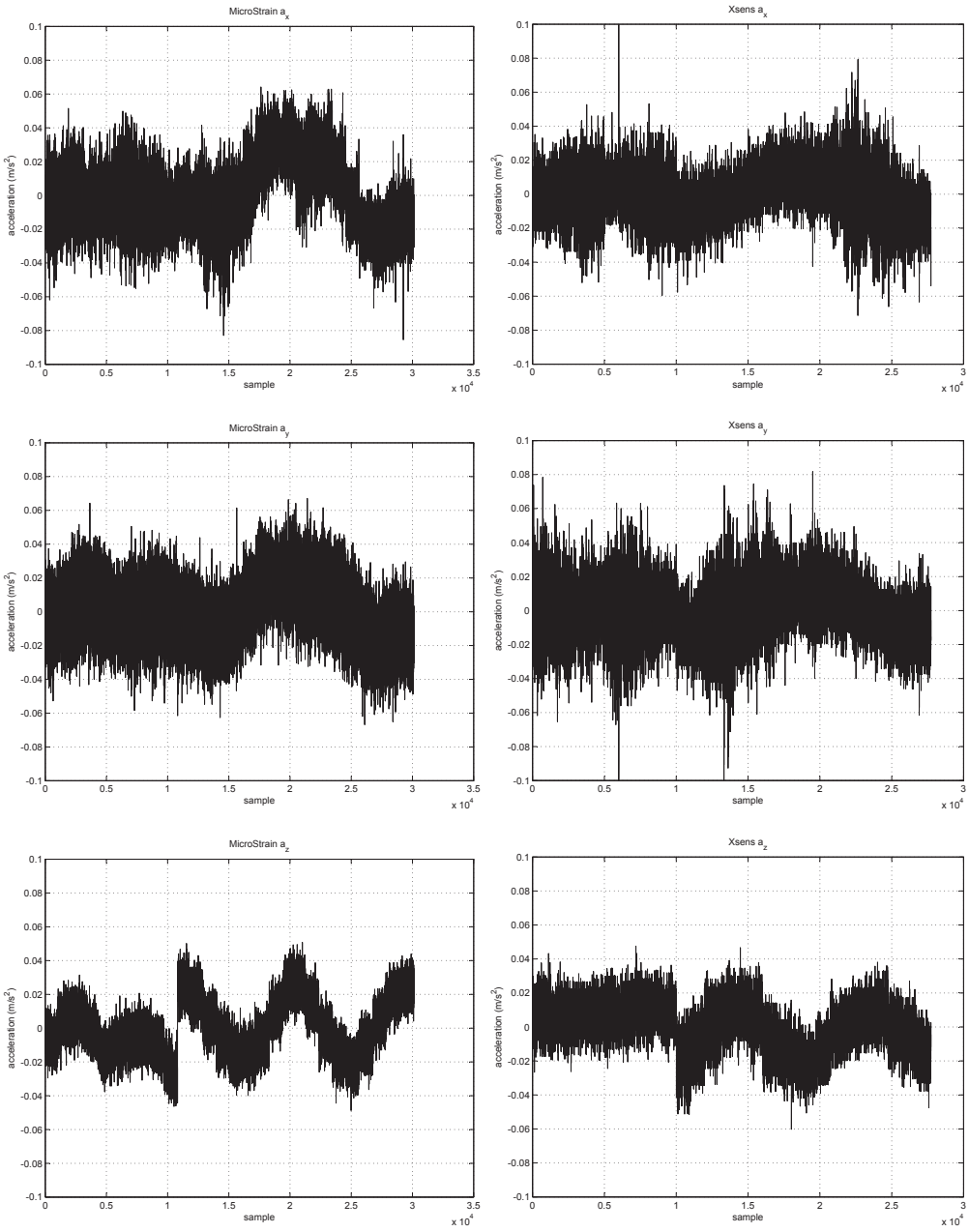


Figure 2.11: Calibrated acceleration measurement errors of all axes of both units.

(a)

	$\vec{b}$ $(\mathrm{m/s^2})$	$diag(\boldsymbol{S})$	ϕ (deg)	θ (deg)	ψ (deg)	α_{zy} (deg)	α_{yz} (deg)	α_{xz} (deg)
MicroStrain	$\begin{bmatrix}-0.094\\-0.013\\0.049\end{bmatrix}$	$\begin{bmatrix}-0.002\\-0.003\\-0.002\end{bmatrix}$	1.081	-0.133	0.323	0.029	90.026	0.123

(b)

	$\vec{b}$ $(\mathrm{m/s^2})$	$diag(\boldsymbol{S})$	ϕ (deg)	θ (deg)	ψ (deg)	α_{zy} (deg)	α_{yz} (deg)	α_{xz} (deg)
Xsens	$\begin{bmatrix}-0.003\\-0.001\\-0.002\end{bmatrix}$	$\begin{bmatrix}-0.002\\-0.002\\-0.002\end{bmatrix}$	0.485	-0.002	-0.292	-0.043	89.993	0.093

Table 2.2: Accelerometer calibration set of the (a) MicroStrain and (b) Xsens units.

According to the performance measures given in Table 2.3, it is observed that the Xsens accelerometer would have better navigation performance than MicroStrain without deterministic calibration. Moreover, the variation of the Xsens accelerometer measurements around the true values is still lower than MicroStrain after calibration, although the improvement achieved by deterministic calibration is greater for MicroStrain than Xsens.

(a)

MicroStrain	$\parallel \vec{y} - f(\vec{\theta}_{w/o}) \parallel$ $(\mathrm{m/s^2})$	$\parallel \vec{y} - f(\vec{\theta^*}) \parallel$ $(\mathrm{m/s^2})$
$\boldsymbol{a_x}$	19.97	3.41
$\boldsymbol{a_y}$	20.17	3.04
$\boldsymbol{a_z}$	20.03	2.83

(b)

Xsens	$\parallel \vec{y} - f(\vec{\theta}_{w/o}) \parallel$ $(\mathrm{m/s^2})$	$\parallel \vec{y} - f(\vec{\theta^*}) \parallel$ $(\mathrm{m/s^2})$
$\boldsymbol{a_x}$	5.30	2.47
$\boldsymbol{a_y}$	9.91	2.76
$\boldsymbol{a_z}$	7.13	2.32

Table 2.3: Measurement errors of the (a) MicroStrain and (b) Xsens accelerometers before and after deterministic calibration.

2.3.2 Gyroscope Calibration

Gyroscope measurements need to be compensated to correct attitude computations during navigation according to a corresponding calibration parameter set. As explained previously, Equation (2.5) can be used to model both accelerometers and gyroscopes, but gyroscope calibration parameters must be obtained in a different way than accelerometers since the reference angular rates are neither available nor computable. In that regard, the error in the computed b-to-p frame DCM, denoted by $\tilde{\boldsymbol{C}}_{\boldsymbol{b}}^{\boldsymbol{p}}$, is considered to be the performance criterion and is minimized by PSO.

Computation of the orientation based on gyroscope measurements requires integration of the DCM. The differential equation corresponding to the integration of the DCM can be expressed by

$$\dot{\boldsymbol{C}}_{\boldsymbol{\mathcal{F}_1}}^{\boldsymbol{\mathcal{F}_2}} = \boldsymbol{C}_{\boldsymbol{\mathcal{F}_1}}^{\boldsymbol{\mathcal{F}_2}} \boldsymbol{\Omega}_{\boldsymbol{\mathcal{F}_2},\boldsymbol{\mathcal{F}_1}}^{\boldsymbol{\mathcal{F}_1}}, \tag{2.19}$$

where $\boldsymbol{\Omega}_{\boldsymbol{\mathcal{F}_2},\boldsymbol{\mathcal{F}_1}}^{\boldsymbol{\mathcal{F}_1}}$ is the skew symmetric form of the angular rate vector of $\mathcal{F}_1$ frame with respect to $\mathcal{F}_2$ expressed in $\mathcal{F}_1$ denoted by $\vec{\omega}_{\mathcal{F}_2,\mathcal{F}_1}^{\mathcal{F}_1}$ [3]. The propagation of $\boldsymbol{C}_{\boldsymbol{\mathcal{F}_1}}^{\boldsymbol{\mathcal{F}_2}}$ between two consecutive time steps (t_{k-1} and t_k) can be expressed as [3]:

$$\boldsymbol{C}_{\boldsymbol{\mathcal{F}_1}}^{\boldsymbol{\mathcal{F}_2}}[k] = \boldsymbol{C}_{\boldsymbol{\mathcal{F}_1}}^{\boldsymbol{\mathcal{F}_2}}[k-1] \exp \int_{t_{k-1}}^{t_k} \boldsymbol{\Omega}_{\boldsymbol{\mathcal{F}_2},\boldsymbol{\mathcal{F}_1}}^{\boldsymbol{\mathcal{F}_1}}(t)\ dt. \tag{2.20}$$

If the sampling interval ($T_s = t_k - t_{k-1}$) is sufficiently small, Equation (2.20) can be approximated by

$$\boldsymbol{C}_{\boldsymbol{\mathcal{F}_1}}^{\boldsymbol{\mathcal{F}_2}}[k] = \boldsymbol{C}_{\boldsymbol{\mathcal{F}_1}}^{\boldsymbol{\mathcal{F}_2}}[k-1] \exp \left(T_s \boldsymbol{\Omega}_{\boldsymbol{\mathcal{F}_2},\boldsymbol{\mathcal{F}_1}}^{\boldsymbol{\mathcal{F}_1}}[k-1] \right) \tag{2.21}$$

After replacing $\mathcal{F}_1$ and $\mathcal{F}_2$ with frames p and b, we can make the transition from the general to our special case, computation of $\tilde{\boldsymbol{C}}_{\boldsymbol{b}}^{\boldsymbol{p}}$. Calibrated gyroscope measurements, denoted by $\tilde{\vec{\omega}}_t^p$, are used instead of $\vec{\omega}_t^p$ for the computation of $\tilde{\boldsymbol{C}}_{\boldsymbol{p}}^{\boldsymbol{b}}$ since $\vec{\omega}_t^p$ is unknown. The $\tilde{\vec{\omega}}_t^p$ can be obtained by compensating the gyroscope measurements $\vec{\omega}_m$ as follows:

$$\tilde{\vec{\omega}}_t^p = \boldsymbol{C}_{\boldsymbol{s}}^{\boldsymbol{p}}\, \boldsymbol{T}^{-\boldsymbol{1}} (\boldsymbol{I} + \boldsymbol{S})^{-\boldsymbol{1}} (\vec{\omega}_m - \vec{b}), \tag{2.22}$$

Equation (2.22) is derived from Equation (2.5). After $\tilde{\vec{\omega}}_t^p$ is obtained, $\tilde{\boldsymbol{C}}_{\boldsymbol{p}}^{\boldsymbol{b}}[\cdot]$ where we use $[\cdot]$ for the time step that $\tilde{\boldsymbol{C}}_{\boldsymbol{p}}^{\boldsymbol{b}}$ belongs to, can be calculated for a given initial orientation.

As stated before, we use PSO for gyroscope deterministic calibration parameter estimation. A brief description of the PSO and the selection of the configuration parameters can be found in Appendix B. The following assignments and configuration settings are realized to implement PSO for gyroscope calibration.

- The fitness function $h(.)$ described in Appendix B to be minimized by PSO is selected as the error between the estimated and the actual DCMs. This can be expressed as

$$\sum_{k=1}^{N} \parallel \boldsymbol{C}_{\boldsymbol{p}}^{\boldsymbol{b}}[k] - \tilde{\boldsymbol{C}}_{\boldsymbol{p}}^{\boldsymbol{b}}[k] \parallel_{fro}, \tag{2.23}$$

  where $\parallel . \parallel_{fro}$ denotes the Frobenius norm operator.

- The parameter set that PSO tries to optimize in the search space is the same as in the accelerometer calibration case since their underlying models are the same. Therefore, $\vec{\theta}$ is the same as in Equation (2.15).

- The population size in PSO is selected as 90. The population size is determined by trial and error and following guidelines provided in [90, 91].

- Initial positions of the particles are determined randomly in the search space.

- Inertia weight, social, and cognitive parameters are respectively selected as $m = 0.8,\ \varphi_p = 2,$ and $\varphi_g = 2$ as suggested in Appendix B.

Calibration parameter set obtained at the end of the PSO, denoted by $\vec{\theta}^*$ and the best $h(.)$ values without and with deterministic calibration using $\vec{\theta}^*$ are given in Tables 2.4 and 2.5. In Table 2.5, $h\left(\vec{\theta}_{w/o}\right)$ is calculated by using that the $\vec{\theta}$ is equal to the right hand side of Equation (2.16) (i.e., the ideal sensor case where no errors of any type are present).

In contrast to the accelerometers, the MicroStrain gyroscope has better error characteristics than Xsens in terms of accuracy for both the calibrated and uncalibrated cases. Finally, it can be stated that a significant reduction in the attitude error is achieved for both types of sensors by calibration.

(a)

	$\vec{b}$ (rad/s)	$diag(\boldsymbol{S})$	ϕ (deg)	θ (deg)	ψ (deg)	α_{zy} (deg)	α_{yz} (deg)	α_{xz} (deg)
MicroStrain	$\begin{bmatrix} -0.002 \\ 0.000 \\ -0.003 \end{bmatrix}$	$\begin{bmatrix} -0.002 \\ 0.001 \\ 0.000 \end{bmatrix}$	0.855	-0.117	0.240	0.049	90.326	-0.176

(b)

	$\vec{b}$ (rad/s)	$diag(\boldsymbol{S})$	ϕ (deg)	θ (deg)	ψ (deg)	α_{zy} (deg)	α_{yz} (deg)	α_{xz} (deg)
Xsens	$\begin{bmatrix} 0.000 \\ 0.001 \\ -0.002 \end{bmatrix}$	$\begin{bmatrix} 0.000 \\ -0.002 \\ 0.001 \end{bmatrix}$	0.156	-0.146	-0.202	0.010	89.991	-0.332

Table 2.4: Gyroscope calibration set of the (a) MicroStrain and (b) Xsens units.

	$h(\vec{\theta}_{w/o})$	$h(\theta^*)$
MicroStrain	12.09	0.64
Xsens	16.91	0.68

Table 2.5: Measurement errors before and after the calibration of both IMUs.

2.3.3 Magnetometer Calibration

The magnetometer error model is relatively more complicated compared to the accelerometer and gyroscope models since the soft iron error ($\boldsymbol{K_{si}}$), hard iron error ($\delta\vec{B}$), and $\boldsymbol{C_n^b}$ are also involved in the model of magnetometers. However, the accelerometers and magnetometers share a common part in terms of reference data availability: $\vec{h}_e^{ned}$ at the location where the experiments are conducted can be looked up as in the accelerometer case where $\vec{g}_L$ is known. Therefore, we decided to utilize LMA for the calibration model parameter estimation of magnetometers. The following assignments are made for the proper operation of LMA:

- The vector $\vec{y}$ is formed by arranging the $\vec{h}_m$s as

$$\vec{y} = \begin{bmatrix} \vec{h}_m^T[1] & \vec{h}_m^T[2] & \cdots & \vec{h}_m^T[N] \end{bmatrix}^T, \tag{2.24}$$

  where $\vec{h}_m[k] \;\; \forall\, k \in N$ denotes the output vector of the magnetometer at time step k.

- $f(.)$ can be expressed in the same way as Equation (2.14) if $o(.)$ is determined as in Equation (2.8). The $o(.)$ is a multi-input function for this case, which takes $\boldsymbol{C_b^p}[k]$ and $\vec{h}_e^{ned}$ as input. The function $f(.)$ used here can be expressed as

$$f(\vec{\theta}) = \left[\begin{array}{cccc} o\left(\boldsymbol{C_b^p}[1], \vec{h}_e^{ned}\right)^T & o\left(\boldsymbol{C_b^p}[2], \vec{h}_e^{ned}\right)^T & \cdots & o\left(\boldsymbol{C_b^p}[N], \vec{h}_e^{ned}\right)^T \end{array} \right]^T, \tag{2.25}$$

- The errors $\delta\vec{B}$ and $\vec{b}$ are augmented into a single bias vector $\tilde{\vec{b}}$, as shown below in order to avoid redundancy in the parameter set.

$$\tilde{\vec{b}} = (\boldsymbol{I} + \boldsymbol{S})\,\delta\vec{B} + \vec{b}$$

The new $\vec{\theta}$ is determined as in Equation (2.15).

$$\vec{\theta} = \left[\begin{array}{cccccccccccccc} \tilde{b}_x & \tilde{b}_y & \tilde{b}_z & s_x & s_y & s_z & \phi & \theta & \psi & \alpha_{zy} & \alpha_{yz} & \alpha_{xz} & \vec{k}_{ij}^T & \Psi \end{array} \right]^T, \tag{2.26}$$

where $\vec{k}_{ij} = \left[\begin{array}{cccccc} k_{11} & k_{12} & k_{13} & k_{22} & k_{23} & k_{33} \end{array} \right]^T$ consists of the elements in the upper right triangle of $\boldsymbol{K_{si}}$ in Equation (2.7).

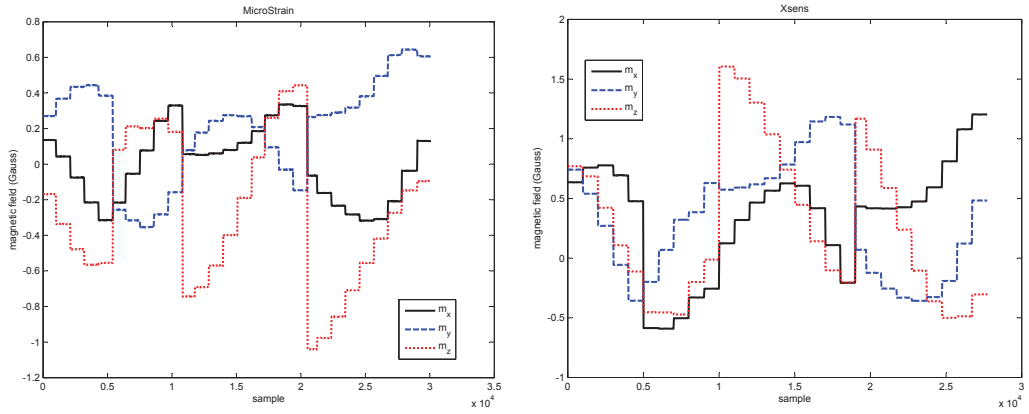
The value of $\vec{h}_e^{ned}$ at the location where experiments are conducted is found to be as follows from the World Magnetic Model 2010 [92]:

$$\vec{h}_e^{ned} = \left[\begin{array}{ccc} 0.2523 & 0.0217 & 0.4004 \end{array} \right]^T \tag{2.27}$$

Before providing the implementation results of the calibration, we show raw magnetometer measurements in Figure 2.12. The signal set looks highly corrupted at a first glance due to the high asymmetry. However, an exact interpretation about the irregularity of the dataset is not possible since $\boldsymbol{C_{ned}^b}$, which is needed to evaluate $\vec{h}_e^{ned}$ in our sensors' frame, is not known.

Configuration parameters and the initial parameter guess $\vec{\theta}_0$ of LMA are selected in the same way as in accelerometer deterministic model parameter identification, and LMA is run. Residual calibration errors are shown in Table 2.6.

The residual errors are not small unlike the inertial sensors, and because of this it is highly unlikely to get even a normal operation performance from our



(a) (b)

Figure 2.12: Magnetometer measurements of both IMUs.

magnetometers. When the dataset is carefully examined, there seems to be a time-varying bias component in the measurements. Such time-varying behavior is generally related to the temperature-dependent bias in the literature. However, it cannot be the explanation for this case since the operating temperature of the sensors do not change significantly because of the short duration of the deterministic experiments. Then, we focus on a different kind of phenomenon and consider the orientation-dependent magnetic fields radiated by the actuation systems in gimbaled systems (e.g., our FMS). We incorporated this effect within hard iron errors since the actuation systems are the dominant factors for the hard iron error vector. The augmented magnetometer model that we propose, can be expressed as

$$\vec{h}_m = (\boldsymbol{I} + \boldsymbol{S})\boldsymbol{T}(\boldsymbol{K_{si}}\boldsymbol{C_p^s}\boldsymbol{C_b^p}\boldsymbol{C_{ned}^b}\,\vec{h}_e^{ned} + \boldsymbol{C_b^p}\delta\vec{B}) + \vec{b} + \vec{v}_m. \tag{2.28}$$

Since the resultant effect of the hard iron errors on the measurements now changes with the $\boldsymbol{C_b^p}$, we separate the $\delta\vec{B}$ and the $\vec{b}$ which are combined during the first identification. The new $\vec{\theta}$ is as follows:

$$\vec{\theta} = \begin{bmatrix} \delta B_x & \delta B_y & \delta B_z & b_x & b_y & b_z & s_x & s_y & s_z & \phi & \theta & \psi & \alpha_{zy} & \alpha_{yz} & \alpha_{xz} & \vec{k}_{ij}^T & \Psi \end{bmatrix}^T. \tag{2.29}$$

The residual errors before and after calibration are given in Table 2.6.

Uncalibrated measurement errors are given in the first column of Table 2.6. Since the uncalibrated case is independent of the sensor model, it is not associated

(a)

MicroStrain	$\parallel \vec{y} - f(\vec{\theta}_{w/o}) \parallel$ (Gauss)	$\parallel \vec{y} - f(\vec{\theta^*}) \parallel$ (Gauss), (Equation (2.5))	$\parallel \vec{y} - f(\vec{\theta^*}) \parallel$ (Gauss), (Equation (2.28))
$\boldsymbol{m_x}$	17.7	9.30	6.49
$\boldsymbol{m_y}$	29.6	19.6	6.38
$\boldsymbol{m_z}$	67.5	14.7	7.80

(b)

Xsens	$\parallel \vec{y} - f(\vec{\theta}_{w/o}) \parallel$ (Gauss)	$\parallel \vec{y} - f(\vec{\theta^*}) \parallel$ (Gauss), (Equation (2.5))	$\parallel \vec{y} - f(\vec{\theta^*}) \parallel$ (Gauss), (Equation (2.28))
$\boldsymbol{m_x}$	148	18.3	8.96
$\boldsymbol{m_y}$	134	23.1	12.9
$\boldsymbol{m_z}$	151	30.1	10.9

Table 2.6: Measurement errors of the (a) MicroStrain and (b) Xsens accelerometers before and after deterministic calibration.

with any model equation. The $f(\vec{\theta}_{w/o})$ is calculated by using

$$\vec{\theta} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 90 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}^T, \quad (2.30)$$

which is the ideal sensor case where no errors of any type are present. The elements of the second and third columns of Table 2.6 are the residual calibration results corresponding to the models in Equations (2.5) and (2.28), respectively. The improvement in the calibration with the proposed magnetometer model is obvious. However, the residual errors are still not sufficient for using the magnetometers as a compass to find the attitude in a navigation system. In order to have a precise magnetometer calibration, which will be fully functional in a navigation system, for our case, it is necessary to develop new and more complex measurement error models.

Chapter 3

Stochastic Modeling and Calibration

Recalling that measurement noise vector $\vec{v}_m$ has three inter-independent measurement noise elements that are effective on each axis. Therefore, it is sufficient to work on a scalar noise model for the stochastic modeling and identification of 3-D inertial sensors and magnetometers. Algorithms for the identification of the key parameters of this scalar model can be repetitively applied to the sensors of each axis. Since the stochastic error model of the noise in the sensor outputs consist of various types of random processes [68], the total scalar noise term that we use throughout this section, is denoted by $v_{tot}(t)$. Quantization noise, white noise, bias instability, random walk, and ramp instability are the typical noise contributors in inertial sensors:

- Quantization noise is introduced into the measurements after analog sensor measurements are converted to digital signals of fixed number of bits determined by the resolution of the analog-to-digital converter. It is the equivalent model of the residual errors after sampling and quantization. Hence, any digital signal is susceptible to this noise. The value of quantization noise at time t is denoted by $v_q(t)$.

- White noise, present in most practical systems, is caused by the high-frequency noise terms whose correlation time is smaller than the sensor output sampling time as described in [74]. Photon shot noise in a fiber optic gyroscope is an example of high-frequency noise [93]. The value of white noise at time t is denoted by $v_w(t)$.

- Bias instability noise is originated by the electronics such as carbon resistors and semiconductor devices [94, 95] in inertial sensors. It is known as the most difficult noise to handle mathematically in stochastic modeling [93]. The value of bias instability noise at time t is denoted by $v_b(t)$.

- Random walk noise is the noise term whose origin is not certain for sure but generally associated with the aging effects in oscillators [93]. The value of random walk noise at time t is denoted by $v_{rw}(t)$.

- Ramp instability noise is indeed a model of a deterministic behavior but it is handled in stochastic modeling since it is possible to formulate a random model for the ramp instability noise. It is usually associated with the external environmental factors such as temperature variations [93]. The value of ramp instability noise at time t is denoted by $v_{ri}(t)$.

The resultant stochastic process, denoted by $v_{tot}(t)$, is equal to the sum of the corresponding noise terms as

$$v_{tot}(t) = v_q(t) + v_w(t) + v_b(t) + v_{rw}(t) + v_{ri}(t). \tag{3.1}$$

Theoretical PSDs of those stochastic signals can be found in Table 3.1 [37]. The objective of stochastic identification is to estimate the unknown descriptive parameters of their PSDs. These parameters, $\mathcal{Q}, \mathcal{N}, \mathcal{B}, \mathcal{K}$, and $\mathcal{R}$, are associated with the processes, $v_q(t), v_w(t), v_b(t), v_{rw}(t)$ and $v_{ri}(t)$, respectively.

The same additive relation also holds in the frequency domain, since all of the noise terms are independent of each other:

$$S_{tot}(f) = S_q(f) + S_w(f) + S_b(f) + S_{rw}(f) + S_{ri}(f), \tag{3.2}$$

where

$$S_x(f) = \int_{-\infty}^{+\infty} e^{-jw\pi f\tau} r_x(\tau) d\tau \tag{3.3}$$

is the PSD of the random process $v_x(t)$ whose autocorrelation function is $r_x(\tau) = E[v_x(t)v_x(t+\tau)]$.

Equation (3.3) allows the PSD to be utilized as a tool for stochastic identification by equating the theoretical PSD to the experimentally found one. However, the sensor output may not usually involve all of the stochastic processes in Equation (3.1). For this reason, overfitting of the model in Equation (3.2) becomes an issue and the stochastic parameters are estimated poorly. This issue can be overcome by first identifying the existing significant noise terms in the output and building the theoretical model accordingly.

AV is an another tool that serves the same purpose as the PSD but in the time domain. Compared to PSD analysis, it presents a more detailed understanding of the dataset and provides average estimation of the stochastic parameters. AV can be described as a method representing the root mean square (RMS) random drift error as a function of averaging time.

The AV of a dataset $\vec{\Omega}$ is denoted by $\sigma^2_\Omega(\tau)$, and the square root of the AV is called the Allan deviation (AD). Elements of $\vec{\Omega}$ are the consecutive sampled instances and can be shown by

$$\vec{\Omega} = \begin{bmatrix} \Omega_1 & \Omega_2 & \cdots & \Omega_N \end{bmatrix}^T, \tag{3.4}$$

where N is the total number of instances. The computation procedure of $\sigma^2_\Omega(\tau)$ is summarized below.

1.) $\vec{\Omega}$ is divided into clusters of length τ which are its sub-vectors. The kth cluster, denoted by $\vec{\Omega}_{\tau_k}$, where $k = 1, \ldots, M = \dfrac{N}{\tau}$, is expressed as

$$\vec{\Omega}_{\tau_k} = \begin{bmatrix} \Omega_{(k-1)\tau+1} & \Omega_{(k-1)\tau+2} & \Omega_{(k-1)\tau+3} & \cdots & \Omega_{k\tau} \end{bmatrix}^T, \tag{3.5}$$

2.) A new data series, denoted by $\tilde{\vec{\Omega}}_\tau$ is formed by the average values of the clusters. It can be expressed as

$$\tilde{\vec{\Omega}}_\tau = \begin{bmatrix} \tilde{\Omega}_{\tau_1} & \tilde{\Omega}_{\tau_2} & \tilde{\Omega}_{\tau_3} & \cdots & \tilde{\Omega}_{\tau_M} \end{bmatrix}^T, \tag{3.6}$$

where $\tilde{\Omega}_{\tau_k} = \frac{1}{\tau}\sum_{i=1}^{\tau}\vec{\Omega}_{\tau_k}[i]$.

3.) The AV $\sigma_\Omega^2(\tau)$ is computed by

$$\sigma_\Omega^2(\tau) = \frac{1}{2(M-1)}\sum_{k=1}^{M-1}(\tilde{\Omega}_{\tau_{k+1}} - \tilde{\Omega}_{\tau_k})^2. \tag{3.7}$$

The $\sigma_\Omega^2(\tau)$ is also related to $S_\Omega(f)$. This relationship can be expressed as

$$\sigma_\Omega^2(\tau) = 4\int_0^\infty S_\Omega(f)\frac{\sin^4(\pi f\tau)}{(\pi f\tau)^2}\mathrm{d}\tau. \tag{3.8}$$

Derivation of Equation (3.8) can be found in [69]. According to this equation, ADs of the random noise terms in the outputs of the inertial sensors can be conveniently calculated from their PSDs. PSDs and ADs of the random processes in Equation (3.1) are given in Table 3.1. Their derivation can be found in [74].

noise component	**symbol**	**PSD**	**AD**
quantization noise [1]	$v_q(t)$	$T_s\mathcal{Q}^2\frac{\sin^2(\pi f T_s)}{(\pi f T_S)^2}$	$\frac{\mathcal{Q}\sqrt{3}}{\tau}$
white noise	$v_w(t)$	$\mathcal{N}^2$	$\frac{\mathcal{N}}{\sqrt{\tau}}$
bias instability [1]	$v_b(t)$	$\frac{\mathcal{B}^2}{2\pi f}$	$\frac{\mathcal{B}\sqrt{2\ln 2}}{\sqrt{\pi}}$
random walk	$v_{rw}(t)$	$\frac{\mathcal{K}^2}{(2\pi f)^2}$	$\frac{\mathcal{K}\sqrt{\tau}}{\sqrt{3}}$
ramp instability	$v_{ri}(t)$	$\frac{\mathcal{R}^2}{(2\pi f)^3}$	$\frac{\mathcal{R}\tau}{\sqrt{2}}$

Table 3.1: PSDs and ADs of the noise terms of an inertial sensor.

It is not possible to perfectly estimate the AV of a real dataset since the accuracy of the AV estimation is limited by N and changes with τ. This can be shown as

$$\text{accuracy} \triangleq 1 - \frac{1}{\sqrt{2(M-1)}}. \tag{3.9}$$

[1] ADs of v_q and v_b are approximations. Their actual expressions can be found in [74].

For the most uniform case, all of the random noise terms can be present in the output of an inertial sensor. However, it is known from practice that different noise terms are dominant over different τ ranges and the slopes of the noise terms in the log-log plot of $\sigma_\Omega^2(\tau)$ versus τ are different. A typical log-log AD curve is illustrated in Figure 3.1.

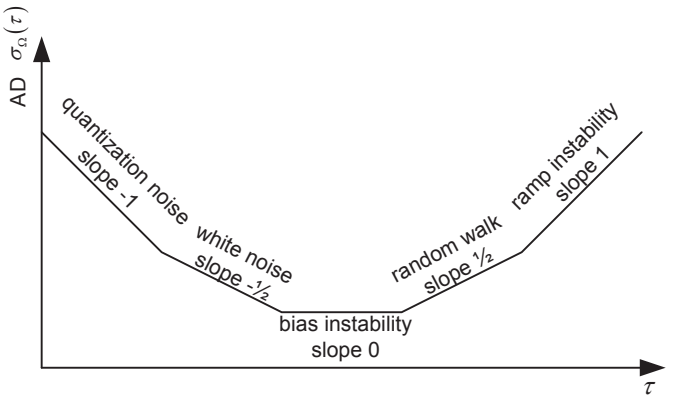


Figure 3.1: Typical AD curve (adapted from [10]).

To estimate the stochastic parameters, AV of the data is calculated for different τ values according to the three-step procedure outlined above. Next, the noise terms constituting the overall stochastic model are heuristically determined from the AD curve. Finally, the experimentally obtained AD curve is fitted to the theoretical AD of the underlying stochastic model.

During the stochastic calibration experiments, inertial sensors are kept stationary. However, the sensor outputs do not remain constant during the experiments because of their temperature-dependent behavior. This drift should be compensated for before proceeding to the AV-based stochastic analysis.

3.1 Maximum Likelihood Estimation of Stochastic Process Parameters

When the stochastic model of an INS is used in a navigation fusion filter, its dynamics are expressed in standard state-space format traditionally. Random walk

noise component	continuous-time dynamic model
quantization noise	$v_q(t) = \mathcal{Q}\sqrt{T_s}\,\dot{n}_q(t)$
white noise	$v_w(t) = \mathcal{N} n_w(t)$
bias instability	$\dot{v}_b(t) = -\beta v_b(t) + \mathcal{B}\sqrt{2\beta}\,n_b(t)$
random walk	$\dot{v}_{rw}(t) = \mathcal{K} n_{rw}(t)$
ramp instability	$\begin{bmatrix} \dot{v}_{ri}(t) \\ \ddot{v}_{ri}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & -\omega_0\sqrt{2} \end{bmatrix} \begin{bmatrix} v_{ri}(t) \\ \dot{v}_{ri}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ \mathcal{R}\sqrt{2\omega_0\sqrt{2}} \end{bmatrix} n_{ri}(t)$

Table 3.2: Continuous-time differential equations of all the random noise terms.

and white noise can be directly expressed in state-space form among the underlying random noise contributors. However, some approximation models should be used to represent the rest of the processes. Many different approximation models have been studied for bias and ramp instability [96, 97, 98]. First- or second-order Gauss-Markov processes are specified to be optimal in terms of accuracy and computational burden for bias and ramp instability, respectively. On the other hand, quantization noise can be approximated by the time derivative of the white noise process. Details of those approximation models can be found in [74].

Differential equations governing the dynamics of the associated random processes are given in Table 3.2 where $n_q(t), n_w(t), n_b(t), n_{rw}(t),$ and $n_{ri}(t)$ represent zero mean, unit variance white Gaussian noise processes that are independent of each other.

The state-space equations of a continuous-time system are:

$$\begin{aligned} \dot{\vec{x}}(t) &= \mathbf{A}\vec{x}(t) + \mathbf{B}\vec{u}(t) + \vec{n}_p(t) \\ \vec{y}(t) &= \mathbf{C}\vec{x}(t) + \mathbf{D}\vec{u}(t) + \vec{n}_m(t) \end{aligned} \tag{3.10}$$

where $\vec{n}_p(t)$ and $\vec{n}_m(t)$ denote the process and measurement noise terms distributed according to $\mathcal{N}(\vec{0}, \boldsymbol{\Sigma_p})$ and $\mathcal{N}(\vec{0}, \boldsymbol{\Sigma_m})$, respectively. In practice, inertial

sensor outputs are observed as discrete samples; therefore, discrete versions of the noise models are needed for evaluation and analysis. A systematic explanation of the discretization of a general continuous-time linear state-space system can be found in [99].

Assuming that the input $u(t)$ passes through a zero-order hold system, the system in Equation (3.11) can be discretized as:

$$\begin{aligned} \vec{x}[k+1] &= \mathbf{A_d}\,\vec{x}[k] + \mathbf{B_d}\,\vec{u}[k] + \vec{n}_p[k] \\ \vec{y}[k] &= \mathbf{C_d}\,\vec{x}[k] + \mathbf{D_d}\,\vec{u}[k] + \vec{n}_m[k] \end{aligned} \tag{3.11}$$

where $\vec{n}_p[k]$ and $\vec{n}_p[k]$ are now distributed according to $\mathcal{N}(\vec{0}, \bar{\boldsymbol{\Sigma}}_{\boldsymbol{p}})$ and $\mathcal{N}(\vec{0}, \bar{\boldsymbol{\Sigma}}_{\boldsymbol{m}})$ and

$$\begin{aligned} \mathbf{A_d} &= e^{\mathbf{A}T_s} \\ \mathbf{B_d} &= \left(\int_{\tau=0}^{T_s} e^{\mathbf{A}\tau} d\tau \right) \mathbf{B} \\ \mathbf{C_d} &= \mathbf{C} \\ \mathbf{D_d} &= \mathbf{D} \\ \bar{\boldsymbol{\Sigma}}_{\boldsymbol{p}} &= \int_{0}^{T_s} e^{\mathbf{A}\tau} \boldsymbol{\Sigma}_{\boldsymbol{p}} e^{\mathbf{A}^T\tau} d\tau \\ \bar{\boldsymbol{\Sigma}}_{\boldsymbol{m}} &= \boldsymbol{\Sigma}_{\boldsymbol{m}}. \end{aligned} \tag{3.12}$$

The method outlined above is not applicable for the discretization of the quantization noise since the dynamics equation of this component does not fit into the classical state-space model. For this reason, quantization noise is simply discretized by Euler's Backward Method [100]. Discrete state-space forms of the noise processes, obtained after the necessary calculations are given in Table 3.3.

Discrete-time difference equations given in Table 3.3 can be equivalently expressed in the discrete-time domain using the time shift operator z^{-1}. Associated discrete-time transfer functions of the noise terms can be found in Table 3.4.

In Tables 3.3 and 3.4, $n_q[k], n_w[k], n_b[k], n_{rw}[k],$ and $n_{ri}[k]$ denote the samples of the corresponding zero mean unit variance white Gaussian noise processes that are all inter-independent at time step k.

noise component	discrete-time dynamic models
quantization noise	$v_q[k] = \frac{\mathcal{Q}}{\sqrt{T_s}} \left(n_q[k] - n_q[k-1] \right)$
white noise	$\mathcal{N} n_w[k]$
bias instability	$v_b[k] = e^{-\beta T_s} \, v_b[k-1] + \mathcal{B} \sqrt{1 - e^{-2\beta T_s}} \, n_b[k-1]$
random walk	$v_{rw}[k] = v_{rw}[k-1] + \mathcal{K} \sqrt{T_s} \, n_{rw}[k-1]$
ramp instability	$\begin{bmatrix} v_{ri}[k] \\ \dot{v}_{ri}[k] \end{bmatrix} = \begin{bmatrix} 1 & e^{T_s} \\ e^{-\omega_0^2 T_s} & e^{-\omega_0 T_s \sqrt{2}} \end{bmatrix} \begin{bmatrix} v_{ri}[k-1] \\ \dot{v}_{ri}[k-1] \end{bmatrix} + \begin{bmatrix} 0 \\ \mathcal{R} \left(1 - e^{-2\sqrt{2}\omega_0 T_s} \right) \end{bmatrix} n_{ri}[k-1]$

Table 3.3: Discrete-time differential equations of all the random noise terms.

According to Equation (3.1), the time propagation equations can be integrated into a multi-input single-output (MISO) system difference equation as

$$\begin{aligned} v_{tot}[k] = & \frac{\mathcal{Q}}{\sqrt{T_s}} \left(1 - z^{-1} \right) n_q[k] + \mathcal{N} n_w[k] \\ & + \frac{\mathcal{B} \sqrt{1 - e^{-2\beta T_s}} z^{-1}}{1 - e^{-\beta T_s} z^{-1}} n_b[k] + \frac{\mathcal{K} \sqrt{T_s} z^{-1}}{1 - z^{-1}} n_{rw}[k] \\ & + \frac{\mathcal{R} e^{T_s} \left(1 - e^{-2\sqrt{2}\omega_0 T_s} \right) z^{-2}}{1 - \left(1 + e^{-\omega_0 T_s \sqrt{2}} \right) z^{-1} + \left(e^{T_s (1 - \omega_0^2) - e^{-\omega_0 T_s \sqrt{2}}} \right) z^{-2}} n_{ri}[k]. \end{aligned} \tag{3.13}$$

Equation (3.13) is sufficient to describe the dynamics of the stochastic part of the inertial sensor output for the most extensive case. After rewriting the equation using a common denominator for the different terms, we get the following:

$$v_{tot}[k] = \frac{\rho_q n_q[k] + \rho_w n_w[k] + \rho_b n_b[k] + \rho_{rw} n_{rw}[k] + \rho_{ri} n_{ri}[k]}{\left(1 - z^{-1} \right) \left(1 - e^{-\beta T_s} z^{-1} \right) D(z^{-1})}, \tag{3.14}$$

noise component	discrete-time difference equations
quantization noise	$\frac{\mathcal{Q}}{\sqrt{T_s}}\left(1-z^{-1}\right)n_q[k]$
white noise	$\mathcal{N}n_w[k]$
bias instability	$\frac{\mathcal{B}\sqrt{1-e^{-2\beta T_s}}\,z^{-1}}{1-e^{-\beta T_s}z^{-1}}\,n_b[k]$
random walk	$\frac{\mathcal{K}\sqrt{T_s}\,z^{-1}}{1-z^{-1}}\,n_{rw}[k]$
ramp instability	$\frac{\mathcal{R}e^{T_s}\left(1-e^{-2\sqrt{2}\omega_0 T_s}\right)z^{-2}}{1-z^{-1}\left(1+e^{-\omega_0 T_s\sqrt{2}}\right)+z^{-2}\left(e^{T_s(1-\omega_0^2)-e^{-\omega_0 T_s\sqrt{2}}}\right)}n_{ri}[k]$

Table 3.4: Discrete-time transfer functions of all the random noise terms.

$$
\begin{aligned}
\text{where } \rho_q &= \frac{\mathcal{Q}}{T_s}\left(1-z^{-1}\right)^2\left(1-e^{-\beta T_s}z^{-1}\right)D(z^{-1})\\
\rho_w &= \mathcal{N}\left(1-z^{-1}\right)\left(1-e^{-\beta T_s}z^{-1}\right)D(z^{-1})\\
\rho_b &= \mathcal{B}\left(1-z^{-1}\right)\sqrt{1-e^{-2\beta T_s}}z^{-1}D(z^{-1})\\
\rho_{rw} &= \mathcal{K}\sqrt{T_s}z^{-1}\left(1-e^{-\beta T_s}z^{-1}\right)D(z^{-1})\\
\rho_{ri} &= \mathcal{R}e^{T_s}\left(1-z^{-1}\right)\left(1-e^{-2\sqrt{2}\omega_0 T_s}\right)\left(1-e^{-\beta T_s}z^{-1}\right)z^{-2}\\
D(z^{-1}) &= 1-\left(1+e^{-\omega_0 T_s\sqrt{2}}\right)z^{-1}+\left(e^{T_s(1-\omega_0^2)}-e^{-\omega_0 T_s\sqrt{2}}\right)z^{-2}.
\end{aligned}
$$

On the other hand, Equation (3.14) is not the only way to model the dynamics of the stochastic nature of inertial sensors. It is also possible to form a single-input single-output (SISO) auto-regressive moving average (ARMA) model that satisfies the same statistical properties. This statistical equivalence is based on the fact that the moving average (MA) parts of both models have the same auto-correlation function [101]. After cross multiplying the terms in Equation (3.14), its right and left hand sides can be respectively represented by two new sequences denoted by $v_{\mathrm{MA}}[k]$ and $v_{\mathrm{AR}}[k]$ as follows:

$$
\begin{aligned}
v_{\mathrm{MA}}[k] &= \rho_q n_q[k]+\rho_w n_w[k]+\rho_b n_b[k]+\rho_{rw}n_{rw}[k]+\rho_{ri}n_{ri}[k]\\
v_{\mathrm{AR}}[k] &= v_{tot}[k]\left(1-z^{-1}\right)\left(1-e^{-\beta T_s}z^{-1}\right)\eta
\end{aligned}
\tag{3.15}
$$

Secondly, the autocorrelation function of $v_{\mathrm{MA}}[k]$ is deduced from Equation (3.15) and can be alternatively modeled by a single input fifth-order MA process $\tilde{v}_{\mathrm{MA}}[k]$ with the same autocorrelation function. This process is expressed in its general form as

$$\tilde{v}_{\mathrm{MA}}[k] = \left(a_0 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + a_4 z^{-4} + a_5 z^{-5}\right) n[k] \tag{3.16}$$

where $n[k]$ denotes the zero-mean and unit variance white Gaussian noise. Finally, the parameters of the process $\tilde{v}_{\mathrm{MA}}[k]$ can be determined by solving the autocorrelation equality condition: $r_{v_{\mathrm{MA}}}[j] = r_{\tilde{v}_{\mathrm{MA}}}[j] \;\; \forall\, j$. Here, we are going to continue with the process $\tilde{v}_{\mathrm{MA}}[k]$ in Equation (3.16) without giving the details of the autocorrelation equality solution.

The auto-regressive (AR) part of Equation (3.14) can be expressed as

$$v_{\mathrm{AR}}[k] = v_{tot}[k]\left(1 + \gamma_1 z^{-1} + \gamma_2 z^{-2} + \gamma_3 z^{-3} + \gamma_4 z^{-4}\right) \tag{3.17}$$

by expanding it and using the coefficients γ_i. As a result, the modified form of the entire stochastic dynamic equation of the stochastic sensor noise can be expressed as

$$\begin{aligned} &\left(1 + \gamma_1 z^{-1} + \gamma_2 z^{-2} + \gamma_3 z^{-3} + \gamma_4 z^{-4}\right) v_{tot}[k] \\ &= \left(a_0 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + a_4 z^{-4} + a_5 z^{-5}\right) n[k]. \end{aligned} \tag{3.18}$$

With the results obtained up to this point, the necessary background is developed for the derivation of the likelihood function and stochastic calibration. Before proceeding, we briefly describe the experiments performed for stochastic calibration and their related consequences. During the stochastic calibration experiments, inertial sensors are fixed to a table top and kept stationary for a period of time. The duration of the experiment is determined by the navigational requirements (e.g., operation time of the IMU). The stationarity in question leads to a definite measurement plus noise at a fixed temperature. Since the sensor's operating temperature inevitably changes in a real experiment depending upon the ambient temperature and the device heating up during operation, the measurement of each axis of an inertial sensor can be modeled as

$$\bar{v}_{tot}[k] = b_0 + \eta\left(T[k]\right) + v_{tot}[k] \tag{3.19}$$

where b_0 represents the sensor bias plus the constant sensible field, $T[k]$ is the operating temperature of the sensor, and $v_{tot}[k]$ is the output of the underlying stochastic process at time instant k. The temperature-dependent bias behavior effect is usually modeled by a lth-order polynomial function of temperature [3, 37]:

$$\eta\left(T[k]\right) = \sum_{i=1}^{l} c_i T^i[k] \tag{3.20}$$

It is obvious that the physical measurement part (i.e., $b_0 + \eta(T[k])$ should be considered to make fair stochastic parameter estimates. This can be addressed in two ways:

- Estimation of those physical measurement parameters and their removal before stochastic calibration
- Concurrent estimation of physical measurement and stochastic parameters

Traditional approaches (e.g., AV) tackle this issue by the former technique and try to estimate stochastic model parameters by compensating for the deterministic part using a least-squares approach. In spite of proven success and the common usage of this strategy in practice, it may lead to biased MLEs of stochastic parameters. The former technique is adopted in this thesis due to the redundancy in the unknown parameters (i.e., effect of b_0 and $\eta(T[k])$ can be canceled by the coefficients of the ARMA process $v_{tot}[k]$). Hence, the physical measurement part must be subtracted out first. Assuming that N is the total number of samples collected from the sensor during an experiment, and physical measurement part is removed from $\bar{\vec{v}}_{tot}$ after the characterization of $b_0 + \eta(T[k])$ in the least-squares sense, MLE aims to find the parameter set maximizing the likelihood function denoted by $L\left(\bar{\vec{v}}_{tot}, \vec{\theta}\right)$, where

$$\begin{aligned} \bar{\vec{v}}_{tot} &= \begin{bmatrix} \bar{v}_{tot}[1] & \bar{v}_{tot}[2] & \cdots & \bar{v}_{tot}[N] \end{bmatrix}^T \\ \vec{\theta} &= \begin{bmatrix} \mathcal{Q} & \mathcal{N} & \mathcal{B} & \beta & \mathcal{K} & \mathcal{R} & \omega_0 \end{bmatrix}^T. \end{aligned} \tag{3.21}$$

$L\left(\bar{\vec{v}}_{tot}, \vec{\theta}\right)$ is determined to be the overall probability of $\bar{\vec{v}}_{tot}$ given $\vec{\theta}$, denoted by $\Pr\left(\bar{\vec{v}}_{tot} \mid \vec{\theta}\right)$. The aim of MLE is to maximize this conditional probability over

the parameter set $\vec{\theta}$:

$$\vec{\theta}^* = \arg\max_{\vec{\theta}} \ \Pr\left(\bar{\vec{v}}_{tot} \mid \vec{\theta}\right) \tag{3.22}$$

Now, we are going to use some facts from random vector theory to simplify Equation (3.22) to our case.

Let $\vec{V} \in \mathbb{R}^d$ be a continuous random vector with the probability density function $p_{\vec{V}}(\vec{v})$. The probability density function of $\vec{Z} = g(\vec{V})$ where $g(.) : \mathbb{R}^d \to \mathbb{R}^d$ can be calculated by

$$p_{\vec{Z}}(\vec{z}) = p_{\vec{V}}\left(g^{-1}(\vec{v})\right) |\boldsymbol{J_g}(\vec{z})| \tag{3.23}$$

where $\boldsymbol{J_g}(.)$ is the Jacobian matrix operator of $g^{-1}(.)$ with respect to $\vec{\theta}$.

In this context, if $\vec{V}$ and $\vec{Z}$ are respectively defined as $\vec{v}_{tot}$ and $\bar{\vec{v}}_{tot}$, $\boldsymbol{J_g}(.)$ becomes the identity matrix and the following conditional probabilities become equal:

$$\Pr\left(\vec{v}_{tot} \mid \vec{\theta}\right) = \Pr\left(\bar{\vec{v}}_{tot} \mid \vec{\theta}\right) \tag{3.24}$$

where $\vec{v}_{tot} = \begin{bmatrix} v_{tot}[1] & v_{tot}[2] & \cdots & v_{tot}[N] \end{bmatrix}^T$. Therefore, $\vec{\theta}^*$ referring to Equation (3.22) maximizes $\Pr\left(\vec{v}_{tot} \mid \vec{\theta}\right)$ as well.

Assuming that $n[k]$ given in Equation (3.16) is equal to 0 $\forall\, k < 1$, the input/output relation for the entire duration of the experiment can be expressed as

$$\boldsymbol{A}\tilde{\vec{n}} = \boldsymbol{S}\vec{v}_{tot} \tag{3.25}$$

where

- $\tilde{\vec{n}} = \begin{bmatrix} \tilde{n}[1] & \tilde{n}[2] & \cdots & \tilde{n}[N] \end{bmatrix}^T_{1\times N}$ and $\tilde{n}[k] = a_0 n[k]$.
- $\boldsymbol{A} = \begin{bmatrix} 1 & 0 & & & & \cdots & & & 0 \\ \tilde{a}_1 & 1 & & & & & & & \\ \tilde{a}_2 & \tilde{a}_1 & 1 & & & & & & \\ \tilde{a}_3 & \tilde{a}_2 & \tilde{a}_1 & 1 & & & & & \vdots \\ \tilde{a}_4 & \tilde{a}_3 & \tilde{a}_2 & \tilde{a}_1 & 1 & \ddots & & & \\ \tilde{a}_5 & \tilde{a}_4 & \tilde{a}_3 & \tilde{a}_2 & \tilde{a}_1 & 1 & & & \\ 0 & \tilde{a}_5 & \tilde{a}_4 & \tilde{a}_3 & \tilde{a}_2 & \tilde{a}_1 & 1 & & \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \tilde{a}_5 & \tilde{a}_4 & \tilde{a}_3 & \tilde{a}_2 & \tilde{a}_1 & 1 \end{bmatrix}_{N\times N}$

whose elements can be expressed by $\tilde{a}_k = \dfrac{a_k}{a_0} \;\; \forall\, k = 1, \ldots, 5$.

- $\boldsymbol{S} = \begin{bmatrix} 1 & 0 & & & \cdots & & & 0 \\ \gamma_1 & 1 & & & & & & \\ \gamma_2 & \gamma_1 & 1 & & & & & \\ \gamma_3 & \gamma_2 & \gamma_1 & 1 & \ddots & & & \vdots \\ \gamma_4 & \gamma_3 & \gamma_2 & \gamma_1 & 1 & & & \\ 0 & \gamma_4 & \gamma_3 & \gamma_2 & \gamma_1 & 1 & & \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \gamma_4 & \gamma_3 & \gamma_2 & \gamma_1 & 1 \end{bmatrix}_{N \times N}$

Since $\boldsymbol{A}$ is a full rank square matrix, Equation (3.25) can be rewritten as

$$\tilde{\vec{n}} = \boldsymbol{A^{-1}} \boldsymbol{S} \vec{v}_{tot} \tag{3.26}$$

It should be noted that the determinant of the Jacobian of the transformation between $\tilde{\vec{n}}$ and $\vec{v}_{tot}$ is unity since both $\boldsymbol{A}$ and $\boldsymbol{S}$ are matrices having unity determinant. Following this fact and Equation (3.23), one can obviously show that $\Pr\left(\vec{v}_{tot} \mid \vec{\theta}\right) = \Pr\left(\tilde{\vec{n}} \mid \vec{\theta}\right)$ holds. Therefore, $L\left(\bar{\vec{v}}_{tot}, \vec{\theta}\right)$ can be explicitly expressed as

$$\begin{aligned} L\left(\bar{\vec{v}}_{tot}, \vec{\theta}\right) &= \Pr\left(\vec{v}_{tot} \mid \vec{\theta}\right) = \Pr\left(\tilde{\vec{n}} \mid \vec{\theta}\right) \\ &= \left(2\pi a_0^2\right)^{-\frac{N}{2}} \exp\left[-\frac{1}{2a_0^2}\left(\boldsymbol{A^{-1}} \boldsymbol{S} \vec{v}_{tot}\right)^T \left(\boldsymbol{A^{-1}} \boldsymbol{S} \vec{v}_{tot}\right)\right] \end{aligned} \tag{3.27}$$

However, $L\left(\bar{\vec{v}}_{tot}, \vec{\theta}\right)$, given in this equation, is the approximate likelihood function because of the assumption $n[k] = 0 \;\; \forall\, k < 1$. To derive the exact likelihood function [102], we define two new vectors $\tilde{\vec{n}}^* = \begin{bmatrix} \tilde{n}[-4] & \tilde{n}[-3] & \tilde{n}[-2] & \tilde{n}[-1] & \tilde{n}[0] \end{bmatrix}^T$ and $\bar{\vec{n}} = \begin{bmatrix} \tilde{\vec{n}}^T & \tilde{\vec{n}}^{*T} \end{bmatrix}^T_{1 \times N+5}$. The relationship between $\bar{\vec{n}}$ and $\vec{v}_{tot}$ can be expressed as

$$\bar{\boldsymbol{A}} \bar{\vec{n}} = \boldsymbol{K} \tilde{\vec{n}}^* + \bar{\boldsymbol{S}} \vec{v}_{tot}, \tag{3.28}$$

where $\bar{\boldsymbol{A}}$, $\boldsymbol{K}$, and $\bar{\boldsymbol{S}}$ are simply structured matrices as follows:

$$\bar{\boldsymbol{A}} = \begin{bmatrix} \boldsymbol{I}_5 & \boldsymbol{0}_{5 \times N} \\ \boldsymbol{0}_{N \times 5} & \boldsymbol{A} \end{bmatrix}_{N+5 \times N+5}, \quad \boldsymbol{K} = \begin{bmatrix} \boldsymbol{I}_5 \\ \boldsymbol{0}_{N \times 5} \end{bmatrix}_{N+5 \times 5}, \quad \bar{\boldsymbol{S}} = \begin{bmatrix} \boldsymbol{0}_{5 \times N} \\ \boldsymbol{S} \end{bmatrix}_{N+5 \times N}.$$

The matrices $\boldsymbol{I}_5$ and $\boldsymbol{0}_{5 \times N}$ respectively denote the identity matrix of size 5 and

zero matrix of size $5 \times N$. This equation introduces the transformation between $\bar{\vec{n}}$ and $\left[\begin{array}{cc} \tilde{\vec{n}}^{*^T} & \vec{v}_{tot} \end{array} \right]$. Since the Jacobian of this transformation is unity, the following is satisfied:

$$\begin{aligned} \Pr\left(\bar{\vec{n}} \mid \vec{\theta}\right) &= \Pr\left(\tilde{\vec{n}}^*, \vec{v}_{tot} \mid \vec{\theta}\right) \\ &= \left(2\pi a_0^2\right)^{-\frac{N+5}{2}} \exp\left[-\frac{1}{2a_0^2}\left(\boldsymbol{K}\tilde{\vec{n}}^* + \bar{\boldsymbol{S}}\vec{v}_{tot}\right)^T \left(\boldsymbol{K}\tilde{\vec{n}}^* + \bar{\boldsymbol{S}}\vec{v}_{tot}\right)\right] \end{aligned} \tag{3.29}$$

If $\bar{\vec{n}}$ is considered to be the measurement noise, the $\tilde{\vec{n}}^*$ can be estimated by applying the generalized least-squares to Equation (3.28) as below:

$$\hat{\tilde{\vec{n}}}^* = -\left(\boldsymbol{K^T}(\bar{\boldsymbol{A}}\bar{\boldsymbol{A}}^{\boldsymbol{T}})^{-\boldsymbol{1}}\boldsymbol{K}\right)^{-\boldsymbol{1}}\boldsymbol{K^T}(\bar{\boldsymbol{A}}\bar{\boldsymbol{A}}^{\boldsymbol{T}})^{-\boldsymbol{1}}\bar{\boldsymbol{S}}\,\vec{v}_{tot} \tag{3.30}$$

An important property of the least-squares based estimators is the orthogonality principle which can be formulated for our case as:

$$\left(\tilde{\vec{n}}^* - \hat{\tilde{\vec{n}}}^*\right)^T \bar{\boldsymbol{S}}\,\vec{v}_{tot} = 0. \tag{3.31}$$

Using this property, the quadratic term $\left(\boldsymbol{K}\tilde{\vec{n}}^* + \bar{\boldsymbol{S}}\vec{v}_{tot}\right)^T \left(\boldsymbol{K}\tilde{\vec{n}}^* + \bar{\boldsymbol{S}}\vec{v}_{tot}\right)$ in Equation (3.29) can be rewritten as

$$\left(\boldsymbol{K}\hat{\tilde{\vec{n}}}^* + \bar{\boldsymbol{S}}\vec{v}_{tot}\right)^T \left(\boldsymbol{K}\hat{\tilde{\vec{n}}}^* + \bar{\boldsymbol{S}}\vec{v}_{tot}\right) + \left(\tilde{\vec{n}}^* - \hat{\tilde{\vec{n}}}^*\right)^T \boldsymbol{K}^T\boldsymbol{K}\left(\tilde{\vec{n}}^* - \hat{\tilde{\vec{n}}}^*\right). \tag{3.32}$$

Furthermore, $\Pr\left(\tilde{\vec{n}}^*, \vec{v}_{tot} \mid \vec{\theta}\right)$ can be decomposed as the product of $\Pr\left(\tilde{\vec{n}}^* \mid \vec{v}_{tot}, \vec{\theta}\right)$ and $\Pr\left(\vec{v}_{tot} \mid \vec{\theta}\right)$. The first quadratic term on the right hand side of Equation (3.32) characterizes $\Pr\left(\vec{v}_{tot} \mid \vec{\theta}\right)$ since that term is a function of signal $\vec{v}_{tot}$ but not of the starting residuals $\tilde{\vec{n}}^*$. Therefore, the exact likelihood function $\Pr\left(\vec{v}_{tot} \mid \vec{\theta}\right)$ can be expressed based on this fact as follows:

$$\begin{aligned} L\left(\bar{\vec{v}}_{tot}, \vec{\theta}\right) &= \Pr\left(\vec{v}_{tot} \mid \vec{\theta}\right) \\ &= \left(2\pi a_0^2\right)^{-\frac{N+5}{2}} \exp\left[-\frac{1}{2a_0^2}\left(\boldsymbol{K}\hat{\tilde{\vec{n}}}^* + \bar{\boldsymbol{S}}\vec{v}_{tot}\right)^T \left(\boldsymbol{K}\hat{\tilde{\vec{n}}}^* + \bar{\boldsymbol{S}}\vec{v}_{tot}\right)\right] \end{aligned} \tag{3.33}$$

In this thesis, the approximate likelihood function is adopted generally. The exact likelihood function is used in some special cases though. Since the maximization of the likelihood functions given in Equations (3.27) and (3.33) are

analytically intractable, iterative optimization algorithms can be used. Here, gradient-ascent optimization (GAO) and particle swarm optimization (PSO) methods are employed for this purpose after the likelihood function of the stochastic process $v_{tot}[k]$ denoted by $L\left(\bar{\vec{v}}_{tot}, \vec{\theta}\right)$ is selected as $f(\vec{\theta})$ for GAO and the fitness function $\mathcal{H}(.)$ for PSO, and $\vec{\theta}$ is selected as Equation (3.21).

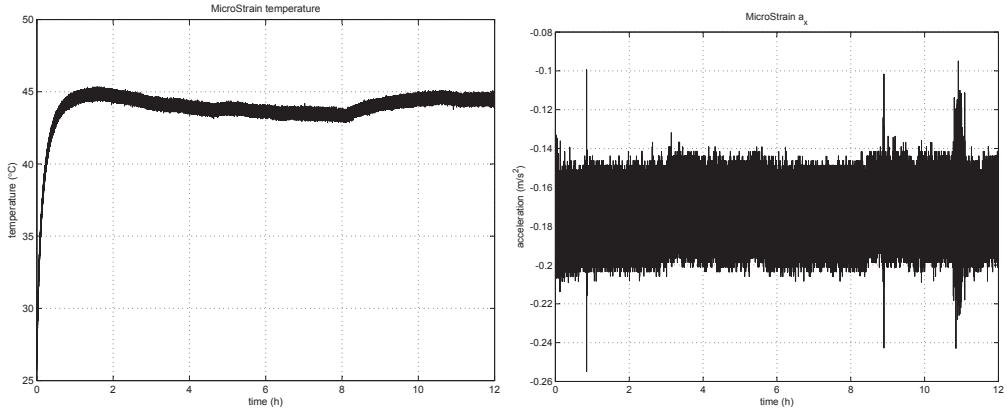
3.2 Experimental Results

For stochastic analysis, we have collected 12 hours of data on four different days at 100 Hz sampling rate from both MicroStrain and Xsens sensor units simultaneously. Therefore, a total of $N = 4,320,000$ observations are recorded from each sensor unit. During the experiments, inertial sensors are fixed to a stationary table. Simultaneously recorded data contain raw tri-axial accelerometer, gyroscope, and magnetometer outputs and temperature measurements. The datasets are processed by using AV, PSO-, and GAO-based MLE techniques for parameter estimation.

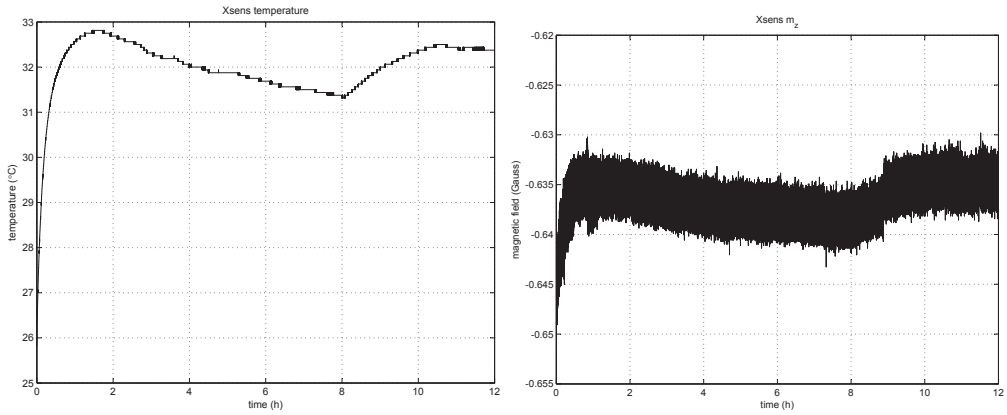
Throughout this section, a sample dataset is chosen randomly for both MicroStrain and Xsens units separately and figures corresponding to that dataset are presented each time an illustrative example is needed.

3.2.1 Allan Variance Analysis

First, the AVs of the datasets need to be examined. The first challenge to this task is the requirement to cancel out the temperature dependent bias drift and the constant bias. For this purpose, we have analyzed the variation of the sensor outputs with temperature. In Figure 3.2, the x-axis accelerometer outputs of MicroStrain 3DM-GX2 and Xsens units recorded on the third day of the experiments are shown. Variation of the sensor measurements with the operating temperature can be observed from the figures. It should be noted that the responses of the sensor units differ subject to their own characteristics, whereas both units have



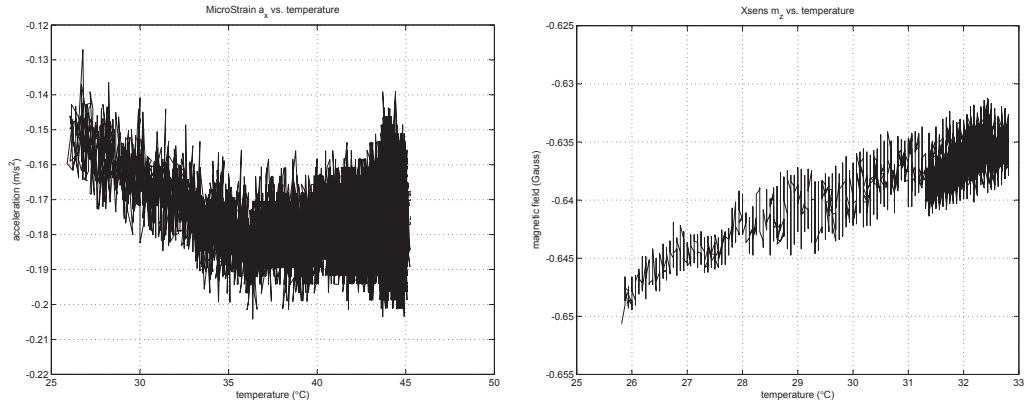
(a)



(b)

Figure 3.2: The third day measurements of (a) MicroStrain and (b) Xsens temperature sensors and accelerometers.

similar temperature trends since they are exposed to the same environmental conditions. However, it is obvious that the MicroStrain unit heats up more than the Xsens unit but this does not mean that the output of the MicroStrain unit is more dependent on temperature. Inherently, AVs of raw datasets cannot be estimated precisely due to the drift in the measurements caused by the operating temperature changes during the experiments. The damaging impacts of the drift error should be compensated for by estimating the temperature and developing temperature-dependent drift models. As mentioned before, exponential [23] and polynomial [37] functions are used for this purpose, respectively. To find a convenient bias drift model for our sensors, the variation of the measurements with operating temperature have been examined and it is inferred that second-order



(a) (b)

Figure 3.3: Accelerometer measurements of (a) MicroStrain and (b) Xsens units versus the operating temperature.

polynomials suitably model the output of the MicroStrain sensor whereas first-order polynomials are suitable for the Xsens unit. In Figure 3.3, the variation of drift as a function of temperature can be observed. Resulting sensor models based on Equation (3.19) are summarized below in Equation (3.34) where $\bar{v}_{tot}[k]$ denotes the output of any inertial sensor at time instant k.

$$\bar{v}_{tot}[k] = \begin{cases} b_0 + c_1 T[k] + c_2 T^2[k] + v_{tot}[k] & \text{for MicroStrain,} \\ b_0 + c_1 T[k] + v_{tot}[k] & \text{for Xsens.} \end{cases} \tag{3.34}$$

The drift models in Equation (3.34) can be expressed as a linear system of equations as in Equation (3.35) after all N measurements are acquired.

MicroStrain	**Xsens**
$\underbrace{\begin{bmatrix} 1 & T[1] & T^2[1] \\ 1 & T[2] & T^2[2] \\ \vdots & \vdots & \vdots \\ 1 & T[N] & T^2[N] \end{bmatrix}}_{\mathfrak{T}} \underbrace{\begin{bmatrix} b_0 \\ c_1 \\ c_2 \end{bmatrix}}_{\vec{\varphi}} = \bar{\vec{v}}_{tot}$	$\underbrace{\begin{bmatrix} 1 & T[1] \\ 1 & T[2] \\ \vdots & \vdots \\ 1 & T[N] \end{bmatrix}}_{\mathfrak{T}} \underbrace{\begin{bmatrix} b_0 \\ c_1 \end{bmatrix}}_{\vec{\varphi}} = \bar{\vec{v}}_{tot}$

(3.35)

In the next step, standard least-squares method is used to estimate the parameter vector $\vec{\varphi}$ that consists of b_0 and temperature correlation coefficients c_i.

(a)

MicroStrain	c_1		c_2	
	mean	**std.**	**mean**	**std.**
$\boldsymbol{a_x}$	$1.16 \cdot 10^{-2}$	$2.92 \cdot 10^{-3}$	$1.44 \cdot 10^{-4}$	$3.27 \cdot 10^{-5}$
$\boldsymbol{a_y}$	$-0.99 \cdot 10^{-2}$	$2.54 \cdot 10^{-3}$	$1.47 \cdot 10^{-4}$	$3.42 \cdot 10^{-5}$
$\boldsymbol{a_z}$	$0.11 \cdot 10^{-2}$	$1.17 \cdot 10^{-3}$	$-1.66 \cdot 10^{-4}$	$5.00 \cdot 10^{-5}$
$\boldsymbol{g_x}$	$1.93 \cdot 10^{-4}$	$3.52 \cdot 10^{-4}$	$-4.65 \cdot 10^{-9}$	$5.85 \cdot 10^{-6}$
$\boldsymbol{g_y}$	$-8.77 \cdot 10^{-5}$	$5.39 \cdot 10^{-4}$	$1.79 \cdot 10^{-9}$	$7.00 \cdot 10^{-6}$
$\boldsymbol{g_z}$	$7.01 \cdot 10^{-4}$	$8.34 \cdot 10^{-4}$	$-1.32 \cdot 10^{-9}$	$1.29 \cdot 10^{-5}$
$\boldsymbol{m_x}$	$-1.08 \cdot 10^{-3}$	$2.19 \cdot 10^{-3}$	$1.49 \cdot 10^{-5}$	$3.11 \cdot 10^{-5}$
$\boldsymbol{m_y}$	$-9.18 \cdot 10^{-4}$	$1.66 \cdot 10^{-3}$	$1.77 \cdot 10^{-5}$	$3.00 \cdot 10^{-5}$
$\boldsymbol{m_z}$	$8.78 \cdot 10^{-4}$	$5.65 \cdot 10^{-4}$	$-1.42 \cdot 10^{-5}$	$9.51 \cdot 10^{-6}$

(b)

Xsens	c_1	
	mean	**std.**
$\boldsymbol{a_x}$	$1.48 \cdot 10^{-3}$	$5.82 \cdot 10^{-4}$
$\boldsymbol{a_y}$	$3.79 \cdot 10^{-4}$	$7.69 \cdot 10^{-4}$
$\boldsymbol{a_z}$	$-1.05 \cdot 10^{-3}$	$9.04 \cdot 10^{-4}$
$\boldsymbol{g_x}$	$2.00 \cdot 10^{-4}$	$6.23 \cdot 10^{-5}$
$\boldsymbol{g_y}$	$-1.38 \cdot 10^{-3}$	$8.08 \cdot 10^{-5}$
$\boldsymbol{g_z}$	$-2.76 \cdot 10^{-4}$	$1.06 \cdot 10^{-4}$
$\boldsymbol{m_x}$	$1.46 \cdot 10^{-3}$	$5.99 \cdot 10^{-4}$
$\boldsymbol{m_y}$	$1.00 \cdot 10^{-3}$	$1.29 \cdot 10^{-3}$
$\boldsymbol{m_z}$	$2.52 \cdot 10^{-3}$	$8.06 \cdot 10^{-4}$

Table 3.5: Estimated temperature correlation parameters of (a) MicroStrain and (b) Xsens IMUs. The units of c_1 are $^\circ\mathrm{C}^{-1}$ ($\mathrm{m/s^2}$) for accelerometers, $^\circ\mathrm{C}^{-1}$ (rad/s) for gyroscopes, and $^\circ\mathrm{C}^{-1}$Gauss for magnetometers. The units of c_2 are the squares of c_1's.

The resulting estimate is denoted by $\hat{\vec{\varphi}}$.

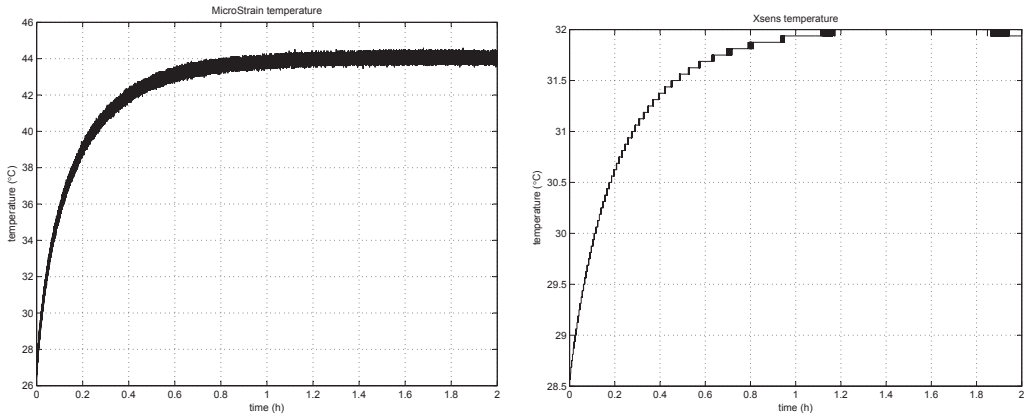
$$\hat{\vec{\varphi}} = \left(\boldsymbol{\mathcal{T}^T \mathcal{T}}\right)^{-1} \boldsymbol{\mathcal{T}^T} \bar{\vec{v}}_{tot} \tag{3.36}$$

Since the sensors' characteristic bias values (b_0) are determined by deterministic calibration procedures, only c_i coefficients are relevant here. The mean and the standard deviation (which is abbreviated by std. in the tables) values of the estimated c_is that are computed over all stationary experiments are displayed in Table 3.5.

Based on the results, the following observations are made:

- Estimation variance of most of the sensors is small.
- Temperature correlation parameters are different for each axis of an IMU.
- Since the estimation variance values are lower, the Xsens unit outperforms the MicroStrain unit in terms of repeatability.

As seen in Figure 3.2, the temperature rise during the initial two hours of the experiments shows a trend similar to the step response of a first-order system after the initial temperature—which is actually the ambient temperature—is subtracted out from the data. Furthermore, the ambient temperature is considered to be constant during the first two hours of the experiments whereas the operating temperature of the sensor units fluctuates afterwards due to the change in the ambient temperature. Temperature of the sensor units during the initial two hours are illustrated in Figure 3.4.



(a) (b)

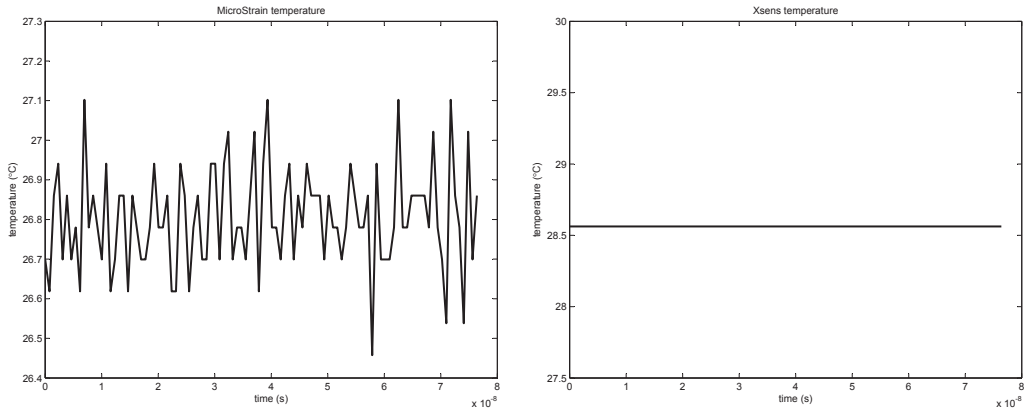
Figure 3.4: Temperature of (a) MicroStrain and (b) Xsens IMUs during the first two hours of operation.

If an analytical model is fitted to the change in the operating temperature of the sensor units after switching on, assuming that the ambient temperature stays constant, the sensor bias drift caused by the operating temperature variation can be estimated by propagating the model in time. However, the estimation accuracy decreases as time passes since the ambient temperature gradually changes. The estimation accuracy becomes quite poor compared to the beginning of the

experiment after the change in the ambient temperature reaches a certain level. This takes about one to two hours for our sensor units. Depending on the application, this duration is sufficient for the estimation of the thermal model parameters with a reasonable accuracy. The following model is used for this purpose [23]:

$$T[k] = T_a + G_T(1 - e^{-\frac{kT_S}{\tau_T}}) \tag{3.37}$$

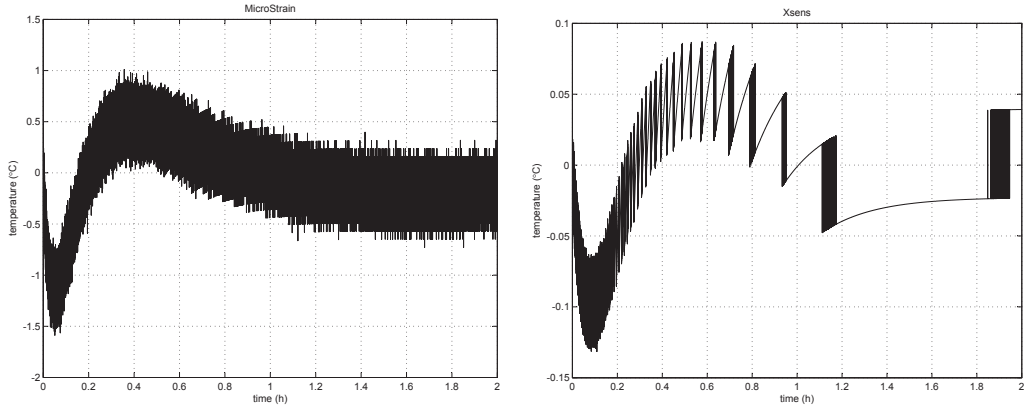
Here, T_a, T_s, G_T and τ_T represent the ambient temperature, sampling time which is equal to 0.01 s for our case, the gain of the temperature model, and the time constant of the system, respectively. The T_a and G_T can be readily extracted from the temperature data of the sensors. If the temperature readings belonging to the first couple of seconds of the experiments are inspected carefully, it is observed that the operating temperature of the sensor unit fluctuates around a constant value for a while and then starts to increase so that T_a can be estimated from the fluctuating portion. This fluctuating portion lasts nearly one second for our IMUs as illustrated in Figure 3.5.



(a) (b)

Figure 3.5: Temperature of (a) MicroStrain and (b) Xsens IMUs during the first second of operation.

After T_a is subtracted out, we use LMA to estimate G_T and τ_T. Residual fitting errors of each sensor unit are shown in Figure 3.6, and estimated parameters of the sensor units are given in Table 3.6. Each sensor of the MicroStrain unit has unique thermal model parameters while they are common for all subsensors of the Xsens sensor unit as seen in Table 3.6. It can be observed in the



(a) (b)

Figure 3.6: Fitting errors of the (a) MicroStrain and (b) Xsens IMUs.

table that the increase in the operating temperature after switching on is a lot more for the MicroStrain unit than the Xsens unit, although the change in the measurement of Xsens unit is greater than the MicroStrain unit as seen in Figure 3.3. This means that the MicroStrain unit is more robust against operating temperature changes compared to the Xsens unit. Furthermore, the variances of the parameter estimates of the MicroStrain unit are lower than those of the Xsens unit. In summary, the temperature-correlated bias of inertial sensors can be estimated for a certain period of time based on only the ambient temperature and this estimation is more precise for the MicroStrain unit than the Xsens.

Before proceeding to AV estimation of stochastic process parameters, bias drift compensation is applied to the datasets according to Equation (3.38) by simply subtracting the expected noise-free output set so that the remaining data pertains only to the stochastic noise. Data after compensation are denoted by $\hat{\vec{v}}_{tot}$ as seen from the equation below:

$$\hat{\vec{v}}_{tot} = \bar{\vec{v}}_{tot} - \boldsymbol{\mathfrak{T}}\hat{\vec{\varphi}} \tag{3.38}$$

In the next stage, AVs of the datasets are estimated. At the end of the AV estimation scheme, a set of AD values for the discrete averaging times are obtained by taking their square roots. According to those AD estimates, AD curves are formed so that intrinsic noise content of the data can be specified. Two examples

(a)

MicroStrain	G_T $(^\circ C)$		τ_T (s)	
	mean	**std.**	**mean**	**std.**
$\boldsymbol{a_x}$, $\boldsymbol{a_y}$, $\boldsymbol{a_z}$	18.051	0.992	592.324	16.232
$\boldsymbol{g_x}$	18.716	0.909	559.601	19.034
$\boldsymbol{g_y}$	17.846	0.843	599.502	23.652
$\boldsymbol{g_z}$	18.436	0.998	570.753	17.089
$\boldsymbol{m_x}$, $\boldsymbol{m_y}$, $\boldsymbol{m_z}$	18.262	0.934	580.207	18.768

(b)

	G_T $(^\circ C)$		τ_T (s)	
	mean	**std.**	**mean**	**std.**
Xsens	6.508	2.126	742.766	94.561

Table 3.6: Mean and standard deviation values of thermal model parameter estimates of the (a) MicroStrain and (b) Xsens IMU.

of estimated AD curves, pertaining to the experiments of day three, are illustrated in Figure 3.7 in dashed lines.

Noise content of the sensor outputs is determined by careful examination of the AD estimates. According to this study, it is determined that all sensors include white noise, bias instability noise, and random walk noise in their output. Then, the characteristic parameters of the noise terms are estimated by fitting the corresponding theoretical AD function to the estimated ADs of the datasets. The governing overall theoretical AD function is the summation of the individual ADs of these noise terms and can be expressed as

$$\frac{\mathcal{N}}{\sqrt{\tau}} + \mathcal{B}\sqrt{\frac{2\ln 2}{\pi}} + \mathcal{K}\sqrt{\frac{\tau}{3}}. \tag{3.39}$$

Based on this theoretical AD function associated with both IMUs, stochastic

parameters can be estimated by solving

$$\begin{bmatrix} 1/\sqrt{\tau_1} & \sqrt{2\ln 2/\pi} & \sqrt{\tau_1/3} \\ 1/\sqrt{\tau_2} & \sqrt{2\ln 2/\pi} & \sqrt{\tau_2/3} \\ \vdots & \vdots & \vdots \\ 1/\sqrt{\tau_M} & \sqrt{2\ln 2/\pi} & \sqrt{\tau_M/3} \end{bmatrix} \underbrace{\begin{bmatrix} \mathcal{N} \\ \mathcal{B} \\ \mathcal{K} \end{bmatrix}}_{\vec{\theta}_s} = \begin{bmatrix} \sigma_{\hat{v}_{tot}}(\tau_1) \\ \sigma_{\hat{v}_{tot}}(\tau_2) \\ \vdots \\ \sigma_{\hat{v}_{tot}}(\tau_M) \end{bmatrix} \tag{3.40}$$

in the same manner as in Equation (3.36). Here, M refers to the number of clusters corresponding to the permissible minimum accuracy as in Equation (3.7). 60% is determined as the lowest bound for accuracy. The mean and standard deviation values of the unknown $\mathcal{N}$, $\mathcal{B}$, and $\mathcal{K}$ are given in Table 3.7.

(a)

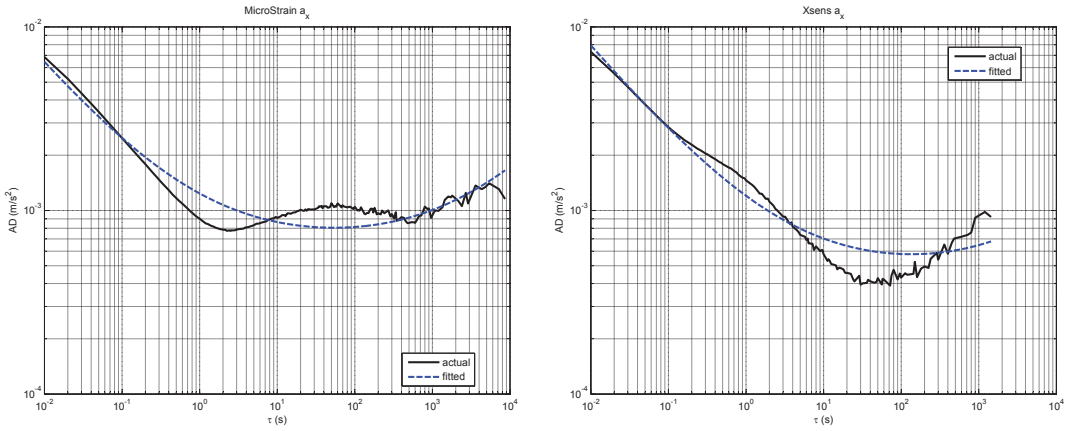
MicroStrain	$\mathcal{N}$		$\mathcal{B}$		$\mathcal{K}$	
	mean	**std.**	**mean**	**std.**	**mean**	**std.**
$\boldsymbol{a_x}$	$5.69 \cdot 10^{-4}$	$1.26 \cdot 10^{-5}$	$1.04 \cdot 10^{-3}$	$6.69 \cdot 10^{-5}$	$1.32 \cdot 10^{-5}$	$3.67 \cdot 10^{-6}$
$\boldsymbol{a_y}$	$1.05 \cdot 10^{-3}$	$9.78 \cdot 10^{-6}$	$3.36 \cdot 10^{-4}$	$4.29 \cdot 10^{-5}$	$2.08 \cdot 10^{-5}$	$5.39 \cdot 10^{-6}$
$\boldsymbol{a_z}$	$6.92 \cdot 10^{-4}$	$4.78 \cdot 10^{-6}$	$3.20 \cdot 10^{-4}$	$1.55 \cdot 10^{-5}$	$8.89 \cdot 10^{-6}$	$4.63 \cdot 10^{-6}$
$\boldsymbol{g_x}$	$5.90 \cdot 10^{-4}$	$5.32 \cdot 10^{-6}$	$5.10 \cdot 10^{-5}$	$3.07 \cdot 10^{-6}$	$2.59 \cdot 10^{-6}$	$5.27 \cdot 10^{-7}$
$\boldsymbol{g_y}$	$5.42 \cdot 10^{-4}$	$4.71 \cdot 10^{-6}$	$4.84 \cdot 10^{-5}$	$9.80 \cdot 10^{-6}$	$3.55 \cdot 10^{-6}$	$8.31 \cdot 10^{-7}$
$\boldsymbol{g_z}$	$5.16 \cdot 10^{-4}$	$3.70 \cdot 10^{-7}$	$6.49 \cdot 10^{-5}$	$3.92 \cdot 10^{-6}$	$3.09 \cdot 10^{-6}$	$7.39 \cdot 10^{-7}$
$\boldsymbol{m_x}$	$9.08 \cdot 10^{-5}$	$7.85 \cdot 10^{-6}$	$5.11 \cdot 10^{-5}$	$4.65 \cdot 10^{-5}$	$4.26 \cdot 10^{-6}$	$2.57 \cdot 10^{-6}$
$\boldsymbol{m_y}$	$8.66 \cdot 10^{-5}$	$2.45 \cdot 10^{-5}$	$7.92 \cdot 10^{-5}$	$1.34 \cdot 10^{-4}$	$3.91 \cdot 10^{-6}$	$3.69 \cdot 10^{-6}$
$\boldsymbol{m_z}$	$9.85 \cdot 10^{-5}$	$1.18 \cdot 10^{-6}$	$7.06 \cdot 10^{-5}$	$8.27 \cdot 10^{-6}$	$3.72 \cdot 10^{-6}$	$1.42 \cdot 10^{-6}$

(b)

Xsens	$\mathcal{N}$		$\mathcal{B}$		$\mathcal{K}$	
	mean	**std.**	**mean**	**std.**	**mean**	**std.**
$\boldsymbol{a_x}$	$7.62 \cdot 10^{-4}$	$1.81 \cdot 10^{-5}$	$6.41 \cdot 10^{-4}$	$4.73 \cdot 10^{-5}$	$1.63 \cdot 10^{-6}$	$4.05 \cdot 10^{-6}$
$\boldsymbol{a_y}$	$8.34 \cdot 10^{-4}$	$2.01 \cdot 10^{-5}$	$1.92 \cdot 10^{-4}$	$1.01 \cdot 10^{-5}$	$5.84 \cdot 10^{-6}$	$2.50 \cdot 10^{-6}$
$\boldsymbol{a_z}$	$6.81 \cdot 10^{-4}$	$7.09 \cdot 10^{-6}$	$6.74 \cdot 10^{-4}$	$5.81 \cdot 10^{-5}$	$2.46 \cdot 10^{-5}$	$3.56 \cdot 10^{-6}$
$\boldsymbol{g_x}$	$6.42 \cdot 10^{-4}$	$9.93 \cdot 10^{-7}$	$3.43 \cdot 10^{-5}$	$2.49 \cdot 10^{-6}$	$2.13 \cdot 10^{-6}$	$5.57 \cdot 10^{-7}$
$\boldsymbol{g_y}$	$6.26 \cdot 10^{-4}$	$1.25 \cdot 10^{-6}$	$2.72 \cdot 10^{-5}$	$2.14 \cdot 10^{-6}$	$2.35 \cdot 10^{-6}$	$1.66 \cdot 10^{-7}$
$\boldsymbol{g_z}$	$5.72 \cdot 10^{-4}$	$8.31 \cdot 10^{-7}$	$2.72 \cdot 10^{-5}$	$1.54 \cdot 10^{-6}$	$1.87 \cdot 10^{-6}$	$3.22 \cdot 10^{-7}$
$\boldsymbol{m_x}$	$6.98 \cdot 10^{-5}$	$1.15 \cdot 10^{-5}$	$9.30 \cdot 10^{-5}$	$8.10 \cdot 10^{-5}$	$7.91 \cdot 10^{-6}$	$7.30 \cdot 10^{-6}$
$\boldsymbol{m_y}$	$4.89 \cdot 10^{-5}$	$3.45 \cdot 10^{-5}$	$1.28 \cdot 10^{-4}$	$1.95 \cdot 10^{-4}$	$7.33 \cdot 10^{-6}$	$7.58 \cdot 10^{-6}$
$\boldsymbol{m_z}$	$7.24 \cdot 10^{-5}$	$3.51 \cdot 10^{-6}$	$3.69 \cdot 10^{-5}$	$1.31 \cdot 10^{-5}$	$1.18 \cdot 10^{-5}$	$1.97 \cdot 10^{-6}$

Table 3.7: Estimated stochastic process parameters of the (a) MicroStrain and (b) Xsens IMUs through AV analysis.

Values related to estimation repeatability in Table 3.7 (i.e., standard deviation) suggest that the estimates are not sufficiently consistent for the particular sensors of both IMUs even if the actual ADs are fitted very well for each individual dataset of all sensors, as shown in Figure 3.7. In other words, estimation variance of the stochastic process parameters of some sensors are high (e.g., estimation of $\mathcal{K}$ of the x-axis accelerometer of the Xsens unit). Thus, identified parameters do not yield a reliable and qualified stochastic model for related sensors. Fitted AD curves according to the estimates and the actual AD curves are shown together in Figure 3.7 for randomly selected datasets as stated before.



(a) (b)

Figure 3.7: Actual and fitted AD curves of the (a) MicroStrain and (b) Xsens IMUs.

3.2.2 Results of PSO- and GAO-based MLE

In the second part of our work on stochastic identification, PSO- and GAO-based MLE algorithms are implemented. For the implementation, initialization strategy for the MLE algorithms, the likelihood function specific to our case and the likelihood function computation routines need to be developed. As can be seen from the preliminary PSO and GAO discussions, initial guesses are made randomly in the entire search space. On the other hand, it is known that having even crude knowledge about the optimal parameter set and making initial guesses accordingly greatly enhances the algorithms' convergence speed. Therefore, we decided

to use AV as a prior processing technique to have a fair initial guess and initialize the particle positions accordingly. However, we cannot obtain an estimate for β of the first-order Gauss-Markov process, which is the approximation of the bias instability noise given in Table 3.2, after the AV analysis. The parameter β value which leads to the minimum error between the first-order Gauss Markov model and the original bias instability model, is used as initial guess. To find that value, the PSD of the output of the approximate bias instability process given in Table 3.2, which will be denoted by $S_{\tilde{b}}(f)$, can be written as

$$S_{\tilde{b}}(f) = \frac{2\mathcal{B}^2\beta}{\beta^2 + (2\pi f)^2}. \tag{3.41}$$

The error between the original and approximate bias instability models can be expressed by the squared difference between their PSDs as given in the equation below:

$$\mathcal{B}^4 \left[\frac{\beta^2 - 4\pi\beta f + (2\pi f)^2}{2\pi f\left(\beta^2 + (2\pi f)^2\right)} \right]^2 \tag{3.42}$$

If the expression given in this equation is integrated over the frequency region, the approximation error of the first-order Gauss-Markov process is obtained as a function of β. Furthermore, reducing the integration band to [0.01, 100] Hz is fairly sufficient from a practical point of view since the bias instability noise is effective over this frequency band [74, 97]. We will not provide the mathematical details of this definite integral since it is not the main focus of this section. If the resultant error expression after the integration is minimized with respect to β by a simple gradient-descent algorithm, the optimal value of β satisfying the minimum error condition for approximate bias instability noise model is obtained as 0.15, and this value is used in the further work as an initial guess.

The final form of the unknown parameter vector $\vec{\theta}$, after β is imported, is given below:

$$\vec{\theta} = \begin{bmatrix} \mathcal{Q} & \mathcal{N} & \mathcal{B} & \beta & \mathcal{K} \end{bmatrix}^T. \tag{3.43}$$

The elements of this vector are the characteristic parameters of the corresponding random processes, quantization noise, bias instability noise, white noise, respectively. The relation between these parameters and the associated random processes can be seen in Tables 3.1–3.4. The objective of PSO- and GAO-based

stochastic identification is to estimate these parameters. Previous analysis gives us a crude idea about the possibly optimal value $\vec{\theta}$, and this idea can be used to initialize PSO and GAO.

As for the likelihood function, we prefer to work with the approximate one, given in Equation (3.27) because the missing initial samples are not significant compared to our huge sample set. We start with formulating $\vec{v}_{tot}$ to define the computation scheme of the chosen likelihood function since the temperature and bias models of both IMUs are known from the previous discussion. In this regard, $\vec{v}_{tot}$ can be calculated according to Equation (3.38) where $\mathfrak{T}$ and $\vec{\varphi}$ specific to each IMU are defined in Equation (3.35).

The $\vec{v}_{tot}$ samples propagate in time according to the following stochastic discrete-time difference equation:

$$v_{tot}[k] = \mathcal{N} n_w[k] + \frac{\mathcal{B}\sqrt{1-e^{-2\beta T_s}}}{1-e^{-\beta T_s} z^{-1}} n_b[k] + \frac{\mathcal{K}\sqrt{T_s} z^{-1}}{1-z^{-1}} n_{rw}[k]. \tag{3.44}$$

After the necessary arrangements, a system equivalent to Equation (3.44) can be described by

$$\begin{aligned} v_{tot}[k]\left(1-e^{-\beta T_s} z^{-1}\right)\left(1-z^{-1}\right) &= \mathcal{N}\left(1-e^{-\beta T_s} z^{-1}\right)\left(1-z^{-1}\right) n_w[k] \\ &+ \mathcal{B}\sqrt{1-e^{-2\beta T_s}} z^{-1}\left(1-z^{-1}\right) n_b[k] \\ &+ \mathcal{K}\sqrt{T_s} z^{-1}\left(1-e^{-\beta T_s} z^{-1}\right) n_{rw}[k]. \end{aligned} \tag{3.45}$$

As stated before, the conversion of the multi-input stochastic system(whose governing equations are given above) to the equivalent single-input stochastic system as we did while deriving Equation (3.18) hinges upon the fact that the input parts of both systems, which are the aforementioned MA parts should have the same autocorrelation function. The $v_{\mathrm{MA}}[k]$ for our IMU's case can be expressed as

$$\begin{aligned} v_{\mathrm{MA}}[k] &= \mathcal{N}\left[1-\left(1+e^{-\beta T_s}\right) z^{-1} + e^{-\beta T_s} z^{-2}\right] n_w[k] \\ &+ \mathcal{B}\sqrt{1-e^{-2\beta T_s}}\left(z^{-1}-z^{-2}\right) n_b[k] \\ &+ \mathcal{K}\sqrt{T_s}\left(z^{-1}-e^{-\beta T_s} z^{-2}\right) n_{rw}[k]. \end{aligned} \tag{3.46}$$

The elements of the autocorrelation function of the process $v_{\mathrm{MA}}[k]$, denoted by $r_{v_{\mathrm{MA}}}$ are given as:

$$\begin{aligned}
r_{v_{\mathrm{MA}}}[0] &= \mathcal{N}^2\left[1+\left(1+e^{-\beta T_s}\right)^2+e^{-2\beta T_s}\right]+2\mathcal{B}^2\left(1-e^{-2\beta T_s}\right) \\
&\quad + T_s\mathcal{K}^2\left(1+e^{-2\beta T_s}\right) \\
r_{v_{\mathrm{MA}}}[1] &= -\mathcal{N}^2\left(1+e^{-\beta T_s}\right)^2-\mathcal{B}^2\left(1-e^{-2\beta T_s}\right)-T_s\mathcal{K}^2e^{-\beta T_s} \\
r_{v_{\mathrm{MA}}}[2] &= \mathcal{N}^2e^{-\beta T_s} \\
r_{v_{\mathrm{MA}}}[k] &= 0 \;\; \forall\, k>2.
\end{aligned} \tag{3.47}$$

Therefore, the SISO system statistically equivalent to $v_{tot}[k]$ which is recognized to be an ARMA$(2,2)$ process can be expressed in parametric form as

$$\left(1+\gamma_1 z^{-1}+\gamma_2 z^{-2}\right)v_{tot}[k]=\left(a_0+a_1z^{-1}+a_2z^{-2}\right)n[k] \tag{3.48}$$

where $n[.]$ denotes zero-mean, unit variance white Gaussian noise. Coefficients of the system given in Equation (3.48) can be computed by the following steps:

- Coefficients γ_1 and γ_2 can be found by expanding the left-hand side of Equation (3.45), which is the AR part of the entire stochastic process $v_{tot}[k]$. Therefore, their computations are straightforward and easy, as evident from Equation (3.49) given below.

$$\begin{aligned}
\gamma_1 &= 1+e^{-\beta T_s} \\
\gamma_2 &= e^{-\beta T_s}
\end{aligned} \tag{3.49}$$

- The a_0, a_1 and a_2 are the coefficients of $\tilde{v}_{\mathrm{MA}}[k]$, which is the single-input MA process equivalent of $v_{\mathrm{MA}}[k]$ expressed in Equation (3.46). Their analytical solutions are based on the equivalence of autocorrelations as mentioned before and can be established through the following steps:

  - In compliance with Equations (3.47) and (3.48), the elements of $r_{v_{\mathrm{MA}}}$ can be alternatively found by

$$\begin{aligned}
r_{v_{\mathrm{MA}}}[0] &= a_0^2+a_1^2+a_2^2 \\
r_{v_{\mathrm{MA}}}[1] &= a_0a_1+a_1a_2 \\
r_{v_{\mathrm{MA}}}[2] &= a_0a_2
\end{aligned} \tag{3.50}$$

– If Equation (3.50) is arranged conveniently, the following equations are obtained:

$$\begin{aligned} a_0 + a_1 + a_2 &= \sqrt{r_{v_{\mathrm{MA}}}[0] + 2r_{v_{\mathrm{MA}}}[1] + 2r_{v_{\mathrm{MA}}}[2]} \\ a_0 - a_1 + a_2 &= \sqrt{r_{v_{\mathrm{MA}}}[0] - 2r_{v_{\mathrm{MA}}}[1] + 2r_{v_{\mathrm{MA}}}[2]} \end{aligned} \tag{3.51}$$

– The a_1 can be readily solved as

$$a_1 = \frac{\sqrt{r_{v_{\mathrm{MA}}}[0] + 2r_{v_{\mathrm{MA}}}[1] + 2r_{v_{\mathrm{MA}}}[2]} - \sqrt{r_{v_{\mathrm{MA}}}[0] - 2r_{v_{\mathrm{MA}}}[1] + 2r_{v_{\mathrm{MA}}}[2]}}{2} \tag{3.52}$$

– If the necessary manipulations are done after substituting a_1 in Equation (3.51), the reduced linear system of equations given below is obtained:

$$\begin{aligned} a_0 + a_2 &= \frac{r_{v_{\mathrm{MA}}}[1]}{a_1} \\ a_0 - a_2 &= \sqrt{r_{v_{\mathrm{MA}}}[0] - a_1^2 - 2r_{v_{\mathrm{MA}}}[2]} \end{aligned} \tag{3.53}$$

– If the two independent relations in Equation (3.53) are solved together, unknown a_0 and a_2 coefficients are obtained as:

$$\begin{aligned} a_0 &= \frac{1}{2}\left(\frac{r_{v_{\mathrm{MA}}}[1]}{a_1} + \sqrt{r_{v_{\mathrm{MA}}}[0] - a_1^2 - 2r_{v_{\mathrm{MA}}}[2]}\right) \\ a_2 &= \frac{1}{2}\left(\frac{r_{v_{\mathrm{MA}}}[1]}{a_1} - \sqrt{r_{v_{\mathrm{MA}}}[0] - a_1^2 - 2r_{v_{\mathrm{MA}}}[2]}\right) \end{aligned} \tag{3.54}$$

All of the parameters of Equation (3.48) can be found by Equations (3.52) and (3.54) provided that $\mathcal{N}$, $\mathcal{B}$, $\mathcal{K}$, β, and T_s are given. Particle positions are used for this purpose during PSO while they can be interpreted from the current solution vector during GAO. Using those parameters, Equation (3.48) can be modified as

$$\left(1 + \gamma_1 z^{-1} + \gamma_2 z^{-2}\right) v_{tot}[k] = \left(1 + \tilde{a}_1 z^{-1} + \tilde{a}_2 z^{-2}\right) \tilde{n}[k], \tag{3.55}$$

where $\tilde{n}$ is the zero-mean white Gaussian noise with variance a_0^2 and $\tilde{a}_1 = \dfrac{a_1}{a_0}$ and $\tilde{a}_2 = \dfrac{a_2}{a_0}$.

Furthermore, the $\boldsymbol{A}$ and $\boldsymbol{S}$ matrices relevant to our case need to be formed to be able to compute $L\left(\vec{\bar{v}}_{tot}, \vec{\theta}\right)$ according to Equation (3.22). Assuming that

$\tilde{n}[k] = 0 \ \forall k < 1$, $\boldsymbol{A}$ and $\boldsymbol{S}$ preserve their special matrix structures so that they are expressed as

$$\boldsymbol{A} = \begin{bmatrix} 1 & 0 & & \cdots & & 0 \\ \tilde{a}_1 & 1 & & & & \\ \tilde{a}_2 & \tilde{a}_1 & 1 & \ddots & & \vdots \\ 0 & \tilde{a}_2 & \tilde{a}_1 & 1 & & \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \tilde{a}_2 & \tilde{a}_1 & 1 \end{bmatrix} \text{ and } \boldsymbol{S} = \begin{bmatrix} 1 & 0 & & \cdots & & 0 \\ \gamma_1 & 1 & & & & \\ \gamma_2 & \gamma_1 & 1 & \ddots & & \vdots \\ 0 & \gamma_2 & \gamma_1 & 1 & & \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \gamma_2 & \gamma_1 & 1 \end{bmatrix}. \quad (3.56)$$

At this point, we obtain the computational procedure for the approximate likelihood function which is needed during the execution of both GAO- and PSO-based MLE methods. Therefore, given a point $\vec{\theta}_i$ in the search space of unknown parameters, the approximate likelihood function can be computed now. The flowchart of the computational procedure for the approximate and exact likelihood functions is given in Figure 3.8. Furthermore, we also need to specify the values of the configuration parameters associated with the optimization algorithms (e.g., stopping criteria ϵ_1 for GAO-based MLE and inertia parameter m for PSO-based MLE) to perform stochastic identification. Details on configuring the settings of PSO and GAO properly are discussed in Appendices B and C, respectively. The configuration parameters are set as follows:

- **GAO-based MLE:** $\alpha_0 = 50$, $\tau = 0.8$, $k_{max} = 1,000$, $\epsilon_1 = \epsilon_2 = 0.005$, and $c = 0.0001$.
- **PSO-based MLE:** $S = 70$, $m = 0.8$, $\vec{b}_l = -14\vec{\theta}_0$, $\vec{b}_u = 16\vec{\theta}_0$, $k_{max} = 1,000$, $\Delta_{max} = 0.1$, $\phi_p = 2$, and $\phi_g = 2$, where $\vec{\theta}_0$ is the initial parameter guess made randomly.

Now, we are ready to perform PSO and GAO. Stochastic model parameters associated with all datasets are estimated using these two methods. In the following, first the results of PSO- and GAO-based MLE are given. Then, some performance curves related to the estimated parameters are shown to compare the two MLE methods and the AV approach among themselves, based on both performance and other criteria such as processing time and robustness.

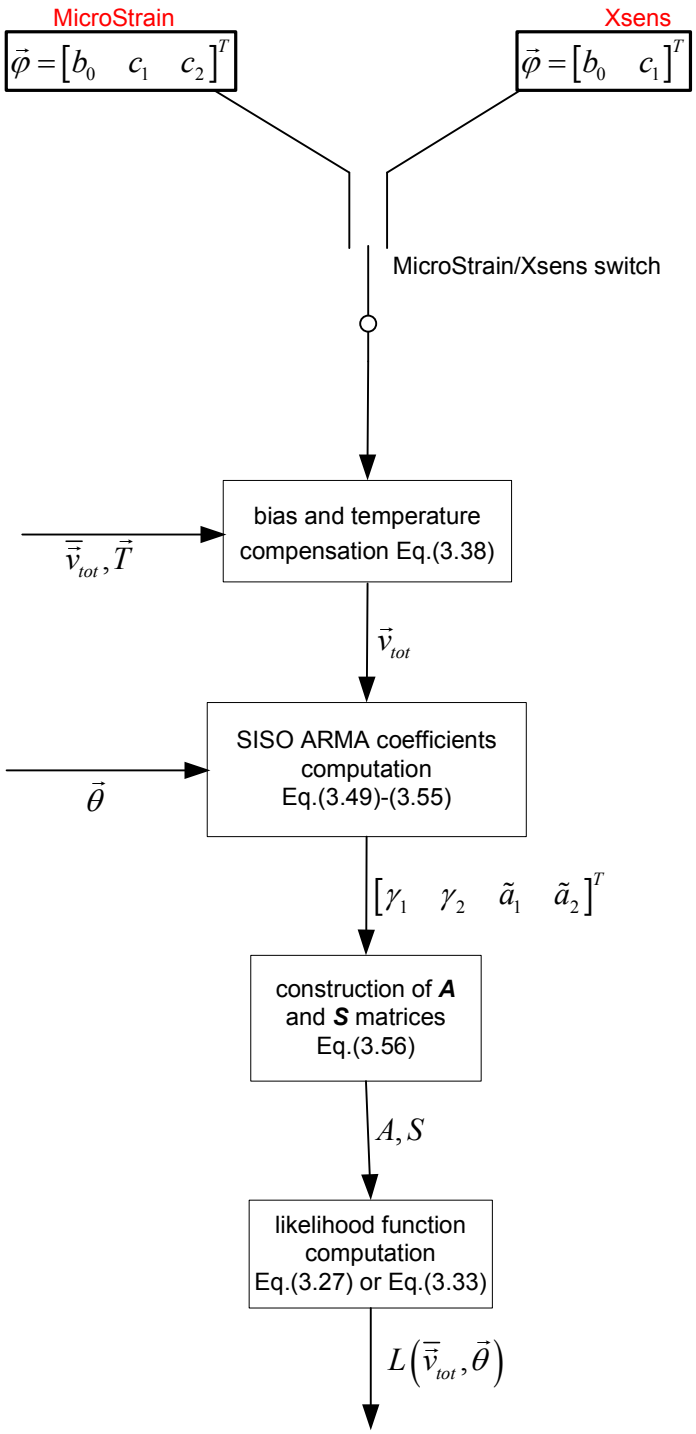


Figure 3.8: The flowchart for the computation of the approximate likelihood function.

The mean and the standard deviation values of the $\vec{\theta}$ estimates of the MicroStrain and Xsens IMUs obtained by PSO- and GAO- based MLE methods are given in Tables 3.9 and 3.10. It can be readily observed in these tables that the MLE techniques outperform the AV technique in terms of estimation consistency. Therefore, it can be interpreted that MLE methods provide more reliable stochastic parameter estimation compared to the AV. Furthermore, MLE methods individually seem to have equally variant estimations. However, it is surprising that the stochastic parameters of the different sensors in the same IMU, found by PSO-based MLE, vary dramatically whereas it is the other way around for GAO-based MLE. In our opinion, this points out that GAO-based MLE is one step ahead of PSO-based MLE in terms of estimation consistency.

Another important criterion to make a comparison between the identification methods is the likelihood function itself. In this regard, average log-likelihood values obtained by all estimation methods for both IMUs are given in Table 3.8.

(a)

MicroStrain	**AV** ($\times 10^8$)	**PSO-based MLE** ($\times 10^7$)	**GAO-based MLE** ($\times 10^7$)
$\boldsymbol{a_x}$	-3.00	1.516	1.508
$\boldsymbol{a_y}$	-2.68	1.288	1.289
$\boldsymbol{a_z}$	-2.07	1.519	1.515
$\boldsymbol{g_x}$	-1.84	1.609	1.607
$\boldsymbol{g_y}$	-1.90	1.641	1.639
$\boldsymbol{g_z}$	-1.82	1.669	1.667
$\boldsymbol{m_x}$	-2.18	2.377	2.376
$\boldsymbol{m_y}$	-3.17	2.367	2.367
$\boldsymbol{m_z}$	-2.11	2.342	2.341

(b)

Xsens	**AV** ($\times 10^8$)	**PSO-based MLE** ($\times 10^7$)	**GAO-based MLE** ($\times 10^7$)
$\boldsymbol{a_x}$	-1.99	1.474	1.474
$\boldsymbol{a_y}$	-1.90	1.406	1.452
$\boldsymbol{a_z}$	-2.51	1.480	1.399
$\boldsymbol{g_x}$	-1.88	1.563	1.568
$\boldsymbol{g_y}$	-1.88	1.568	1.579
$\boldsymbol{g_z}$	-1.87	1.610	1.618
$\boldsymbol{m_x}$	-2.47	2.416	2.460
$\boldsymbol{m_y}$	-1.14	2.473	2.512
$\boldsymbol{m_z}$	-2.23	2.437	2.459

Table 3.8: Log-likelihood values obtained by the different estimation techniques for (a) MicroStrain and (b) Xsens.

(a)

MicroStrain	$\mathcal{N}$		$\mathcal{B}$		β		$\mathcal{K}$	
	mean	**std.**	**mean**	**std.**	**mean**	**std.**	**mean**	**std.**
$\boldsymbol{a_x}$	$3.70 \cdot 10^{-3}$	$3.66 \cdot 10^{-3}$	$4.50 \cdot 10^{-3}$	$2.76 \cdot 10^{-3}$	112.92	112.69	$3.42 \cdot 10^{-4}$	$2.54 \cdot 10^{-4}$
$\boldsymbol{a_y}$	$1.22 \cdot 10^{-2}$	$7.94 \cdot 10^{-5}$	$1.08 \cdot 10^{-2}$	$1.34 \cdot 10^{-4}$	$5.68 \cdot 10^{-2}$	$9.64 \cdot 10^{-3}$	$6.20 \cdot 10^{-5}$	$1.70 \cdot 10^{-7}$
$\boldsymbol{a_z}$	$1.31 \cdot 10^{-8}$	≈ 0	$7.26 \cdot 10^{-3}$	$6.93 \cdot 10^{-5}$	246.70	12.77	$2.19 \cdot 10^{-4}$	$4.59 \cdot 10^{-5}$
$\boldsymbol{g_x}$	$5.42 \cdot 10^{-3}$	$1.33 \cdot 10^{-4}$	$1.78 \cdot 10^{-3}$	$3.55 \cdot 10^{-4}$	76.11	13.25	$1.40 \cdot 10^{-5}$	$3.27 \cdot 10^{-6}$
$\boldsymbol{g_y}$	$5.74 \cdot 10^{-9}$	≈ 0	$5.34 \cdot 10^{-3}$	$2.91 \cdot 10^{-5}$	289.95	6.75	$2.31 \cdot 10^{-5}$	$4.27 \cdot 10^{-8}$
$\boldsymbol{g_z}$	$6.15 \cdot 10^{-9}$	$5.38 \cdot 10^{-9}$	$5.06 \cdot 10^{-3}$	$8.29 \cdot 10^{-6}$	221.26	0.33	$2.13 \cdot 10^{-5}$	$3.59 \cdot 10^{-6}$
$\boldsymbol{m_x}$	$9.79 \cdot 10^{-4}$	$8.74 \cdot 10^{-8}$	$9.31 \cdot 10^{-5}$	$3.65 \cdot 10^{-5}$	0.805	0.74	$1.34 \cdot 10^{-5}$	$5.59 \cdot 10^{-6}$
$\boldsymbol{m_y}$	$1.01 \cdot 10^{-3}$	$2.44 \cdot 10^{-6}$	$9.22 \cdot 10^{-5}$	$2.99 \cdot 10^{-5}$	0.662	0.64	$3.83 \cdot 10^{-6}$	$3.73 \cdot 10^{-6}$
$\boldsymbol{m_z}$	$1.07 \cdot 10^{-3}$	$5.44 \cdot 10^{-6}$	$2.18 \cdot 10^{-4}$	$1.16 \cdot 10^{-5}$	1.215	$4.47 \cdot 10^{-2}$	$1.63 \cdot 10^{-5}$	$4.88 \cdot 10^{-6}$

(b)

Xsens	$\mathcal{N}$		$\mathcal{B}$		β		$\mathcal{K}$	
	mean	**std.**	**mean**	**std.**	**mean**	**std.**	**mean**	**std.**
$\boldsymbol{a_x}$	$7.40 \cdot 10^{-3}$	$4.95 \cdot 10^{-5}$	$3.00 \cdot 10^{-3}$	$1.03 \cdot 10^{-4}$	14.61	1.91	$1.80 \cdot 10^{-4}$	$3.59 \cdot 10^{-6}$
$\boldsymbol{a_y}$	$4.30 \cdot 10^{-3}$	$2.01 \cdot 10^{-5}$	$7.10 \cdot 10^{-3}$	$1.62 \cdot 10^{-4}$	195.28	$7.19 \cdot 10^{-3}$	$8.37 \cdot 10^{-5}$	$7.64 \cdot 10^{-6}$
$\boldsymbol{a_z}$	$3.60 \cdot 10^{-3}$	$3.62 \cdot 10^{-3}$	$5.60 \cdot 10^{-3}$	$2.30 \cdot 10^{-3}$	182.51	80.9	$2.02 \cdot 10^{-4}$	$2.42 \cdot 10^{-5}$
$\boldsymbol{g_x}$	$6.40 \cdot 10^{-3}$	$2.15 \cdot 10^{-6}$	$1.02 \cdot 10^{-4}$	$3.00 \cdot 10^{-5}$	$3.24 \cdot 10^{-2}$	$1.18 \cdot 10^{-2}$	$5.53 \cdot 10^{-6}$	$1.92 \cdot 10^{-6}$
$\boldsymbol{g_y}$	$6.20 \cdot 10^{-3}$	$8.59 \cdot 10^{-6}$	$1.12 \cdot 10^{-4}$	$3.99 \cdot 10^{-6}$	$4.68 \cdot 10^{-2}$	$9.67 \cdot 10^{-3}$	$1.01 \cdot 10^{-5}$	$1.00 \cdot 10^{-6}$
$\boldsymbol{g_z}$	$5.70 \cdot 10^{-3}$	$1.79 \cdot 10^{-6}$	$1.14 \cdot 10^{-4}$	$6.69 \cdot 10^{-6}$	$5.24 \cdot 10^{-2}$	$2.27 \cdot 10^{-2}$	$8.94 \cdot 10^{-6}$	$6.85 \cdot 10^{-7}$
$\boldsymbol{m_x}$	$4.15 \cdot 10^{-4}$	$1.79 \cdot 10^{-5}$	$8.49 \cdot 10^{-4}$	$2.65 \cdot 10^{-5}$	118.81	$2.06 \cdot 10^{-1}$	$1.53 \cdot 10^{-4}$	$1.15 \cdot 10^{-4}$
$\boldsymbol{m_y}$	$7.46 \cdot 10^{-4}$	$1.30 \cdot 10^{-6}$	$6.65 \cdot 10^{-5}$	$2.88 \cdot 10^{-7}$	0.133	$2.63 \cdot 10^{-3}$	$1.67 \cdot 10^{-5}$	$6.56 \cdot 10^{-6}$
$\boldsymbol{m_z}$	$8.60 \cdot 10^{-4}$	$3.27 \cdot 10^{-5}$	$1.15 \cdot 10^{-4}$	$1.02 \cdot 10^{-5}$	$9.56 \cdot 10^{-2}$	$5.52 \cdot 10^{-3}$	$3.05 \cdot 10^{-5}$	$2.78 \cdot 10^{-6}$

Table 3.9: Estimated stochastic process parameters of (a) MicroStrain and (b) Xsens IMUs through PSO-based MLE.

(a)

MicroStrain	$\mathcal{N}$		$\mathcal{B}$		β		$\mathcal{K}$	
	mean	**std.**	**mean**	**std.**	**mean**	**std.**	**mean**	**std.**
$\boldsymbol{a_x}$	$7.30 \cdot 10^{-3}$	$6.67 \cdot 10^{-5}$	$1.41 \cdot 10^{-3}$	$4.05 \cdot 10^{-4}$	$1.04 \cdot 10^{-1}$	$3.86 \cdot 10^{-3}$	$2.58 \cdot 10^{-4}$	$2.17 \cdot 10^{-4}$
$\boldsymbol{a_y}$	$1.22 \cdot 10^{-2}$	$6.63 \cdot 10^{-5}$	$1.13 \cdot 10^{-4}$	$1.77 \cdot 10^{-4}$	$1.47 \cdot 10^{-1}$	$1.49 \cdot 10^{-3}$	$3.14 \cdot 10^{-4}$	$3.05 \cdot 10^{-5}$
$\boldsymbol{a_z}$	$7.28 \cdot 10^{-3}$	$8.01 \cdot 10^{-5}$	$7.75 \cdot 10^{-4}$	$7.62 \cdot 10^{-5}$	$1.45 \cdot 10^{-1}$	$5.54 \cdot 10^{-4}$	$7.20 \cdot 10^{-5}$	$6.00 \cdot 10^{-6}$
$\boldsymbol{g_x}$	$5.76 \cdot 10^{-3}$	$8.02 \cdot 10^{-5}$	$1.48 \cdot 10^{-4}$	$2.48 \cdot 10^{-5}$	$1.42 \cdot 10^{-1}$	$3.19 \cdot 10^{-3}$	$1.07 \cdot 10^{-5}$	$2.76 \cdot 10^{-6}$
$\boldsymbol{g_y}$	$5.36 \cdot 10^{-3}$	$7.44 \cdot 10^{-5}$	$2.05 \cdot 10^{-4}$	$6.92 \cdot 10^{-5}$	$1.50 \cdot 10^{-1}$	$3.61 \cdot 10^{-3}$	$8.37 \cdot 10^{-6}$	$7.27 \cdot 10^{-6}$
$\boldsymbol{g_z}$	$5.07 \cdot 10^{-3}$	$1.09 \cdot 10^{-5}$	$1.93 \cdot 10^{-4}$	$4.99 \cdot 10^{-6}$	$1.50 \cdot 10^{-1}$	$1.36 \cdot 10^{-3}$	$1.68 \cdot 10^{-5}$	$6.07 \cdot 10^{-6}$
$\boldsymbol{m_x}$	$9.81 \cdot 10^{-4}$	$1.77 \cdot 10^{-6}$	$1.32 \cdot 10^{-4}$	$2.26 \cdot 10^{-4}$	$1.50 \cdot 10^{-1}$	$3.36 \cdot 10^{-5}$	$7.31 \cdot 10^{-5}$	$4.62 \cdot 10^{-5}$
$\boldsymbol{m_y}$	$1.00 \cdot 10^{-3}$	$3.89 \cdot 10^{-6}$	$8.46 \cdot 10^{-5}$	$5.90 \cdot 10^{-5}$	$1.50 \cdot 10^{-1}$	$1.56 \cdot 10^{-5}$	$8.69 \cdot 10^{-5}$	$1.39 \cdot 10^{-4}$
$\boldsymbol{m_z}$	$1.08 \cdot 10^{-3}$	$1.89 \cdot 10^{-5}$	$1.76 \cdot 10^{-4}$	$1.98 \cdot 10^{-4}$	$1.50 \cdot 10^{-1}$	$1.13 \cdot 10^{-4}$	$1.45 \cdot 10^{-4}$	$1.22 \cdot 10^{-4}$

(b)

Xsens	$\mathcal{N}$		$\mathcal{B}$		β		$\mathcal{K}$	
	mean	**std.**	**mean**	**std.**	**mean**	**std.**	**mean**	**std.**
$\boldsymbol{a_x}$	$7.75 \cdot 10^{-3}$	$4.63 \cdot 10^{-5}$	$1.13 \cdot 10^{-3}$	$1.52 \cdot 10^{-3}$	$1.43 \cdot 10^{-1}$	$6.34 \cdot 10^{-3}$	$3.66 \cdot 10^{-3}$	$2.13 \cdot 10^{-4}$
$\boldsymbol{a_y}$	$8.36 \cdot 10^{-3}$	$1.56 \cdot 10^{-4}$	$4.86 \cdot 10^{-4}$	$2.50 \cdot 10^{-5}$	$1.46 \cdot 10^{-1}$	$2.84 \cdot 10^{-4}$	$3.40 \cdot 10^{-5}$	$8.16 \cdot 10^{-6}$
$\boldsymbol{a_z}$	$1.21 \cdot 10^{-2}$	$8.41 \cdot 10^{-3}$	$9.26 \cdot 10^{-4}$	$1.10 \cdot 10^{-3}$	$1.46 \cdot 10^{-1}$	$2.63 \cdot 10^{-3}$	$7.66 \cdot 10^{-3}$	$1.50 \cdot 10^{-4}$
$\boldsymbol{g_x}$	$6.37 \cdot 10^{-3}$	$9.69 \cdot 10^{-6}$	$1.38 \cdot 10^{-4}$	$2.51 \cdot 10^{-5}$	$1.48 \cdot 10^{-1}$	$7.27 \cdot 10^{-4}$	$9.69 \cdot 10^{-6}$	$8.96 \cdot 10^{-7}$
$\boldsymbol{g_y}$	$6.22 \cdot 10^{-3}$	$1.62 \cdot 10^{-5}$	$1.23 \cdot 10^{-4}$	$1.17 \cdot 10^{-5}$	$1.48 \cdot 10^{-1}$	$5.30 \cdot 10^{-4}$	$9.86 \cdot 10^{-6}$	$2.04 \cdot 10^{-6}$
$\boldsymbol{g_z}$	$5.68 \cdot 10^{-3}$	$7.60 \cdot 10^{-6}$	$1.25 \cdot 10^{-4}$	$9.26 \cdot 10^{-6}$	$1.48 \cdot 10^{-1}$	$2.97 \cdot 10^{-4}$	$7.42 \cdot 10^{-6}$	$1.59 \cdot 10^{-6}$
$\boldsymbol{m_x}$	$8.42 \cdot 10^{-4}$	$1.13 \cdot 10^{-4}$	$2.66 \cdot 10^{-4}$	$2.86 \cdot 10^{-4}$	$1.49 \cdot 10^{-1}$	$7.02 \cdot 10^{-4}$	$2.21 \cdot 10^{-4}$	$2.45 \cdot 10^{-4}$
$\boldsymbol{m_y}$	$7.47 \cdot 10^{-4}$	$8.36 \cdot 10^{-6}$	$1.44 \cdot 10^{-4}$	$1.66 \cdot 10^{-4}$	$1.49 \cdot 10^{-1}$	$1.24 \cdot 10^{-3}$	$7.39 \cdot 10^{-5}$	$1.15 \cdot 10^{-4}$
$\boldsymbol{m_z}$	$9.08 \cdot 10^{-4}$	$7.30 \cdot 10^{-5}$	$1.15 \cdot 10^{-5}$	$2.18 \cdot 10^{-5}$	$1.50 \cdot 10^{-1}$	$6.92 \cdot 10^{-6}$	$5.11 \cdot 10^{-5}$	$5.24 \cdot 10^{-6}$

Table 3.10: Estimated stochastic process parameters of (a) MicroStrain and (b) Xsens IMUs through GAO-based MLE.

It is observed in Table 3.8 that larger likelihood values are acquired by the proposed MLE methods compared to the traditional AV technique. This means that the MLE leads to more precise estimates. If the success of the MLE methods is examined, it is observed that neither of them is superior to the other since the log-likelihood values obtained with them are very close to each other. More specifically, GAO-based MLE generally outperforms the PSO-based MLE for the Xsens unit while it is the other way around for the MicroStrain unit as can be seen in Figure 3.9. For the MicroStrain unit, PSO-based MLE provides tracking of the even very slow changes of the true PSD while the PSD obtained by GAO-based MLE does not fit very well to the actual PSD since it does not follow those changes in PSD. On the other hand, for the Xsens unit, GAO-based MLE provides a PSD very close to the true one whereas the PSD obtained by PSD-based MLE is biased. The success of the MLE methods is obvious in these figures.

The last criterion is the elapsed time for the convergence of the algorithms. Average processing durations are given in Table 3.11 for this purpose.

average processing time (s)		
AV	**PSO-based MLE**	**GAO-based MLE**
3	$86,400$	$7,200$

Table 3.11: Average processing time of different estimation techniques for 12 hours of data.

Compared to AV, MLE techniques require an order of magnitude more time to converge. Although it seems to be a serious disadvantage at a first glance, it is not an issue in practice since stochastic identification is an offline task that needs to be done only once before putting the sensors into use. However, we still consider what happens in the case for which we have a lower number of samples to process. Can we obtain the same results by MLE techniques with a lower absolute processing time because of the lower number of samples taken into account? We reran MLE techniques for two cases without using the results of the AV analysis as initial guesses: data of six minutes and data of 24 minutes. We prefer to use the exact likelihood function given in Equation (3.33) since the effect of the assumption $n[k] = 0 \ \forall\, k < 1$, that we made for approximate likelihood function, might be significant for our test cases having small sample sets. Log-likelihood values obtained after the runs are shown in Tables 3.12 and 3.13.

(a)

MicroStrain	**AV** ($\times 10^8$)	**PSO-based MLE** ($\times 10^7$)	**GAO-based MLE** ($\times 10^7$)
$\boldsymbol{a_x}$	-1.790	1.517	1.515
$\boldsymbol{a_y}$	-2.160	1.286	1.286
$\boldsymbol{a_z}$	-1.691	1.515	1.514
$\boldsymbol{g_x}$	-2.443	1.601	1.601
$\boldsymbol{g_y}$	-2.370	1.633	1.633
$\boldsymbol{g_z}$	-2.209	1.665	1.663
$\boldsymbol{m_x}$	-3.517	2.373	2.374
$\boldsymbol{m_y}$	-4.251	2.366	2.365
$\boldsymbol{m_z}$	-1.977	2.342	2.341

(b)

Xsens	**AV** ($\times 10^8$)	**PSO-based MLE** ($\times 10^7$)	**GAO-based MLE** ($\times 10^7$)
$\boldsymbol{a_x}$	-2.538	1.477	1.474
$\boldsymbol{a_y}$	-2.030	1.454	1.452
$\boldsymbol{a_z}$	-1.875	1.480	1.479
$\boldsymbol{g_x}$	-2.026	1.568	1.568
$\boldsymbol{g_y}$	-1.990	1.579	1.579
$\boldsymbol{g_z}$	-2.020	1.618	1.618
$\boldsymbol{m_x}$	-1.367	2.460	2.347
$\boldsymbol{m_y}$	-1.569	2.509	2.509
$\boldsymbol{m_z}$	-1.090	2.453	2.454

Table 3.12: Log-likelihood values obtained by different estimation techniques with data of six minute duration for the (a) MicroStrain and (b) Xsens units.

(a)

MicroStrain	**AV** ($\times 10^8$)	**PSO-based MLE** ($\times 10^7$)	**GAO-based MLE** ($\times 10^7$)
$\boldsymbol{a_x}$	-2.176	1.517	1.515
$\boldsymbol{a_y}$	-2.311	1.288	1.288
$\boldsymbol{a_z}$	-1.831	1.515	1.515
$\boldsymbol{g_x}$	-2.126	1.606	1.605
$\boldsymbol{g_y}$	-2.142	1.637	1.637
$\boldsymbol{g_z}$	-2.019	1.669	1.666
$\boldsymbol{m_x}$	-6.931	2.375	2.375
$\boldsymbol{m_y}$	-6.665	2.367	2.366
$\boldsymbol{m_z}$	-2.088	2.341	2.329

(b)

Xsens	**AV** ($\times 10^8$)	**PSO-based MLE** ($\times 10^7$)	**GAO-based MLE** ($\times 10^7$)
$\boldsymbol{a_x}$	-2.236	1.480	1.469
$\boldsymbol{a_y}$	-1.924	1.457	1.414
$\boldsymbol{a_z}$	-1.818	1.481	1.478
$\boldsymbol{g_x}$	-1.962	1.569	1.558
$\boldsymbol{g_y}$	-1.944	1.579	1.560
$\boldsymbol{g_z}$	-1.950	1.619	1.607
$\boldsymbol{m_x}$	-1.642	2.444	2.432
$\boldsymbol{m_y}$	-1.404	2.507	2.474
$\boldsymbol{m_z}$	-1.344	2.445	2.419

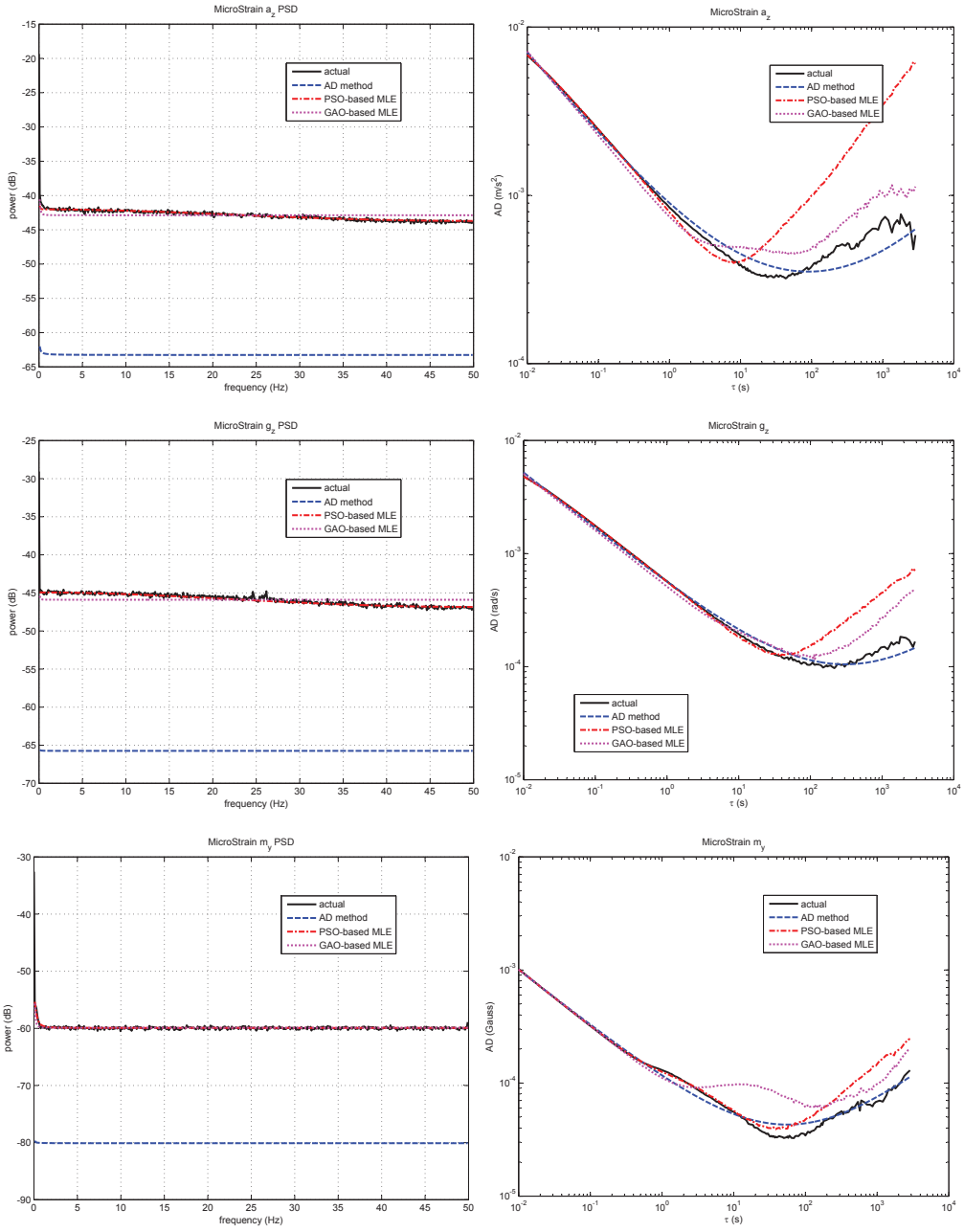
Table 3.13: Log-likelihood values obtained by different estimation techniques with data of 24 minute duration for (a) MicroStrain and (b) Xsens units.

As seen in Tables 3.12 and 3.13, MLE methods still provide highly accurate stochastic identification even with the very small number of samples, whereas the results of the AV method are unsatisfactory due to the accuracy issue of the AV. Log-likelihood values achieved by the MLE methods using fewer samples are close to the results of MLE methods using the entire sample set and they are even better than the AV method using the entire sample set. Therefore, it can be stated that MLE methods can reach greater success with a relatively small number of samples meaning that the duration of the experiments for the stochastic identification can be very short. We finally compare the algorithms in terms of the total amount of time spent for the experiment and the average processing duration. We consider the data of six minute duration for MLE methods and 12 hours of data for the AV method and show the results in Table 3.14.

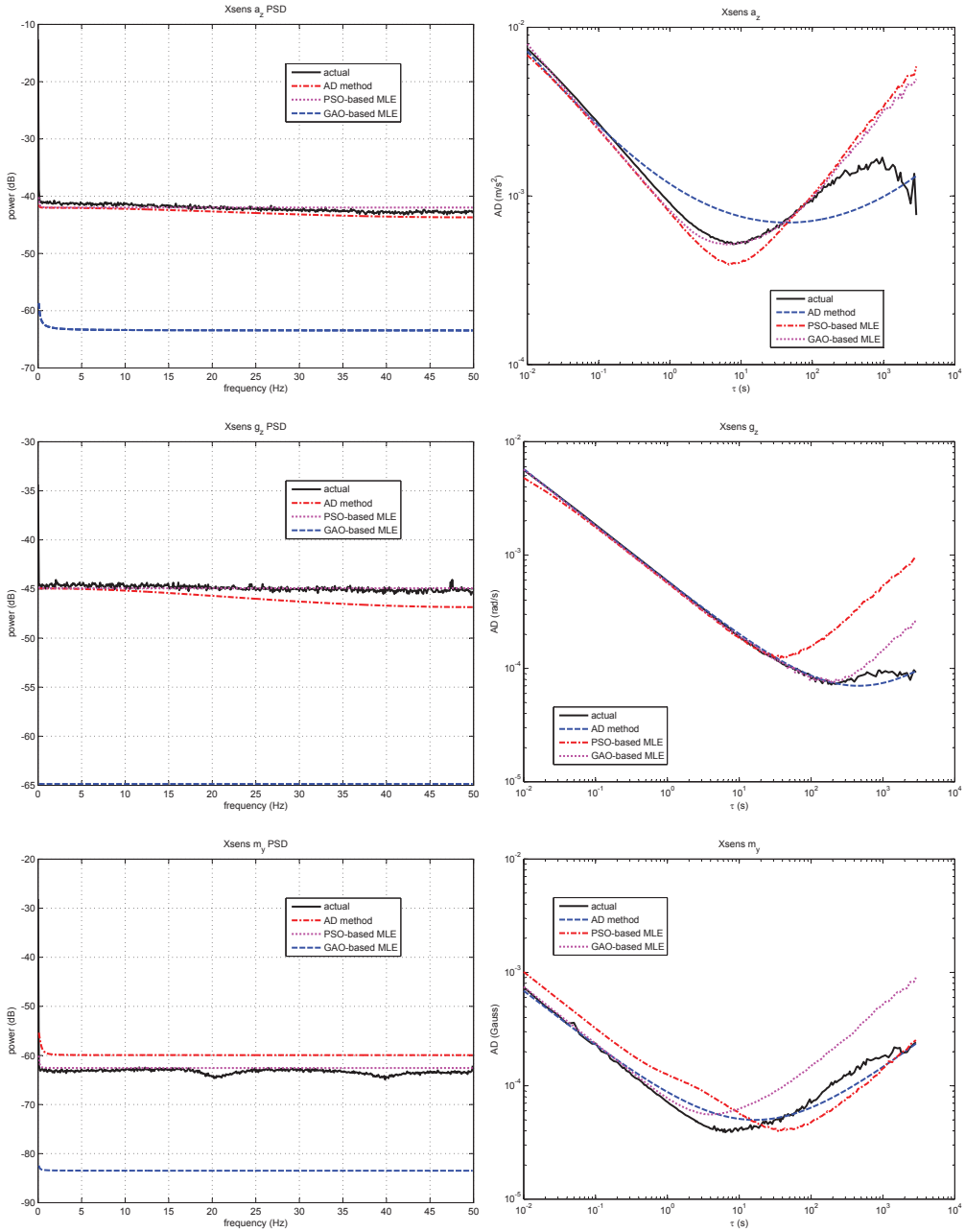
total amount of time (s)		
AV	**PSO-based MLE**	**GAO-based MLE**
$43,203$	$1,080$	$1,800$

Table 3.14: The sum of the duration of the experiment and the average processing duration.

As a result, our proposed MLE methods can provide more consistent and accurate results with less time consumption because they need fewer measurements compared to classical estimation techniques (e.g., AV estimation).


MicroStrain a_z PSD
power (dB)
frequency (Hz)
actual
AD method
PSO-based MLE
GAO-based MLE
MicroStrain a_z
AD (m/s^2)
τ (s)
MicroStrain g_z PSD
MicroStrain g_z
AD (rad/s)
MicroStrain m_y PSD
MicroStrain m_y
AD (Gauss)

(a)



(b)

Figure 3.9: Performance curves of different estimation techniques for the (a) MicroStrain and the (b) Xsens unit.

Chapter 4

CONCLUSIONS AND FUTURE WORK

In this thesis, we presented deterministic and stochastic error models for inertial sensors and magnetometers and error model parameter estimation results. In the first part, we formulated the deterministic measurement model of the sensors using the results from earlier works. Then, we presented some identification methods for the deterministic error parameters based on experimentally acquired data. A classical method was preferred for accelerometers since the reference data were available and satisfactory results were obtained. However, the classical approach was not sufficient for gyroscopes due to the reference data limitation of the FMS device that was used for the experiments (i.e., instantaneous angular rates were not available). Therefore, we developed a novel estimation technique based on the comparison of the computed attitude and the reference attitude that the FMS provides. We used PSO, which is a model-free evolutionary optimization technique, since the derivation of the attitude model based on sensor error parameters would have been a highly complex task. We achieved sufficient modeling accuracy with the proposed approach which can also be implemented for in-field calibration problems. Magnetometers were the worst among all the sensors and we could not manage to obtain modeling errors as small as those of inertial sensors. Traditional magnetometer model-based identification led to

poor results due to the unmodeled effects (e.g., orientation-dependent hard iron errors). Hence, we proposed an extended sensor model for the magnetometers and managed to improve the residual calibration errors by 50%.

In the second part, the objective was to model the measurement noise of the sensors and identify the associated model parameters. To the best of our knowledge, generalized exact and approximate likelihood functions have not been derived and MLE has not been employed accordingly for all possible noise terms in inertial and magnetic sensors . There is one study [103] which adopts the likelihood function maximization idea but uses the simplified noise models that do not take the effects of all noise terms into account. It is also stated in the same paper that the proposed method leads to poor estimates when a complex stochastic model is used for the total measurement noise. In this thesis, unlike the traditional techniques, the likelihood function maximization approach that considers the generalized noisy case, is proposed and employed. First, the procedure for MLE technique was developed after the measurement noise had been modeled by an ARMA model which can be suitably adopted in INS/GPS filters such as the Kalman filter. Before the MLE was employed using the experimental data collected during the stationary position of the sensors, the mean and the temperature-dependent part of the data had been removed. Two different optimization algorithms (PSO and GAO) were implemented, and the results were compared to the classical AV-based estimation approach in terms of some performance measures: quality of the estimates, fitted and actual data PSD and AV curves, likelihood values, and the total amount of time spent for the experiments and calibration. It has been shown that MLE methods outperform the AV-based estimation in all aspects. More precise and consistent stochastic parameter estimates can be obtained using a dataset that is 80% smaller through the use of MLE techniques.

It is not possible to compare the error model's whole parameter estimates obtained in this thesis with the specifications and data provided by the manufacturer (see Appendix D) since given information on measurement errors of sensors is very limited. Regarding reference data, we can compare in terms of following two error features:

- **bias stability**: Bias stability refers to the minimum of AD of a sensor over the entire operating temperature of a sensor. It is the least amount of noise that one can achieve by averaging sensor measurements. Reference bias stability values are given in Table 1.1. Estimated bias stability values can be seen in Figure 3.9. Obtained bias stability estimates are much less than manufacturer specified values. A possible explanation of this can be the worsening effect of temperature on bias stability that is taken into account by manufacturers. However, we do not have any observation about the change of bias stability with temperature as we did not have a chance to repeat our experiments at different ambient temperatures.

- **misalignment**: This error term can be compared only for Xsens unit since reference data is provided only by them. It is stated in the specifications sheet of Xsens IMU that the misalignment is not more than 0.1° in all axes for all sensors. However, it is observed in this thesis that estimated misalignment angles exceed 0.1° mostly. This exceeding difference is less than 0.05° for gyroscope while it is approximately three times greater for accelerometers. Exact values of misalignment angle estimates are given in Tables 2.2 and 2.4.

The success of the results of both deterministic and stochastic model identification problems can also be investigated in actual standalone and aided INSs in future work. Some effort must be absolutely put on the more complex modeling of the magnetometers to further improve the calibration accuracy. Furthermore, the proposed calibration method of gyroscopes can be implemented for a real in-field calibration problem dataset and compared to the other techniques.

Bibliography

[1] Microstrain, "3DM-GX2® — Microstrain." http://www.microstrain.com/inertial/3DM-GX2.

[2] Xsens, "Products - Xsens." http://www.xsens.com/en/general/mtx.

[3] D. H. Titterton and J. L. Weston, *Strapdown Inertial Navigation Technology*. Reston, VA, U.S.A.: The American Institute of Aeronautics and Astronautics, 2nd ed., 1979.

[4] The Institute of Navigation Museum, "Autopilot Gyroscope." http://www.ion.org/museum/item_view.cfm?cid=1&scid=16&iid=40.

[5] Ideal Aerosmith, "Cost effective UAV hardware-in-the-loop flight motion simulation system." http://www.ideal-aerosmith.com/old_newsletter/2005november/1573FMS.html.

[6] National Instruments, "Rensselaer Polytechnic Institute 150 g-ton centrifuge." http://sine.ni.com/cms/images/casestudies/bloomycentria.jpg?size.

[7] Ideal Aerosmith, "1270VS single-axis rate table." http://www.ideal-aerosmith.com/motion/1270VS.aspx.

[8] Wikipedia, "Geodetic system - Wikipedia, the free encyclopedia." http://en.wikipedia.org/wiki/Geodetic_system.

[9] Acutronic, "3-Axis Motion Simulator Series AC3357 - ACUTRONIC / CH." http://www.acutronic.com/ch/products/3-axis/product-detail/

view/3-axis-motion-simulator-series-ac3357.html?nar=2&cHash=c49a557c2b8ee11462027ecccf338bd8.

[10] IEEE, *IEEE Standard Specification Format Guide and Test Procedure for Linear, Single-Axis, Digital, Torque-Balance Accelerometer*. 1992.

[11] Acutronic, "Inertial Guidance Test Instrument, Three Axis Motion Simulator Series, AC3357," datasheet, Bubikon, Switzerland, June 2011.

[12] E. Jovanov, A. Milenkovic, C. Otto, and P. C de Groen, "A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation," *Journal of NeuroEngineering and Rehabilitation*, vol. 2, no. 6, March 2005.

[13] W. H. Wu, A. A. T. Bui, M. A. Batalin, D. Liu, and W. J. Kaiser, "Incremental diagnosis method for intelligent wearable sensor system," *IEEE Transactions on Information Technology in Biomedicine*, vol. 11, no. 5, pp. 553–562, September 2007.

[14] W. T. Ang, P. K. Khosla, and C. N. Riviere, "Design of all-accelerometer inertial measurement unit for tremor sensing in hand-held microsurgical instrument," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2, (Tapei, Taiwan), pp. 1781–1786, 14–19 September 2003.

[15] W. T. Ang, P. K. Pradeep, and C. N. Riviere, "Active tremor compensation in microsurgery," in *Proceedings of the 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 1, (San Francisco, CA), pp. 2738–2741, 1–5 September 2004.

[16] A. M. Sabatini, Inertial sensing in biomechanics: a survey of computational techniques bridging motion analysis and personal navigation, pp. 70–100. *Computational Intelligence for Movement Sciences: Neural Networks and Other Emerging Techniques*, Hershey, PA, U.S.A.: Idea Group Publishing, 2006.

[17] F. Audigié, P. Pourcelot, C. Degueurce, D. Geiger, and J. M. Denoix, "Fourier analysis of trunk displacements: a method to identify the lame limb in trotting horses," *Journal of Biomechanics*, vol. 35, no. 9, pp. 1173–1182, September 2002.

[18] K. Hauer, S. E. Lamb, E. C. Jorstad, C. Todd, and C. Becker, "Systematic review of definitions and methods of measuring falls in randomised controlled fall prevention trials," *Age and Ageing*, vol. 35, no. 1, pp. 5–10, January 2006.

[19] N. Noury, A. Fleury, P. Rumeau, A. K. Bourke, G. O. Laighin, V. Rialle, and J. E. Lundy, "Fall detection—principles and methods," in *Proceedings of the 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, (Lyon, France), pp. 1663–1666, August 2007.

[20] M. Kangas, A. Konttila, P. Lindgren, I. Winblad, and T. Jämsä, "Comparison of low-complexity fall detection algorithms for body attached accelerometers," *Gait & Posture*, vol. 28, no. 2, pp. 285–291, August 2008.

[21] M. J. Mathie, B. G. Celler, N. H. Lovell, and A. C. F. Coster, "Classification of basic daily movements using a triaxial accelerometer," *Medical and Biological Engineering and Computing*, vol. 42, no. 5, pp. 679–687, September 2004.

[22] J. Pärkkä, M. Ermes, P. Korpipää, J. Mäntyjärvi, J. Peltola, and I. Korhonen, "Activity classification using realistic data from wearable sensors," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 1, pp. 119–128, January 2006.

[23] B. Barshan and H. F. Durrant-Whyte, "Inertial navigation systems for mobile robots," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 1, pp. 328–342, June 1995.

[24] B. Barshan and H. F. Durrant-Whyte, "Evaluation of a solid-state gyroscope for robotics applications," *IEEE Transactions on Instrumentation and Measurement*, vol. 44, no. 1, pp. 61–67, February 1995.

[25] C.-W. Tan and S. Park, "Design of accelerometer-based inertial navigation systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 54, no. 6, pp. 2520–2530, December 2005.

[26] J. G. Nichol, S. P. N. Singh, K. J. Waldron, L. R. Palmer, III, and D. E. Orin, "System design of a quadrupedal galloping machine," *International Journal of Robotics Research*, vol. 23, no. 10–11, pp. 1013–1027, October-November 2004.

[27] P. C. Lin, H. Komsuoglu, and D. E. Koditschek, "Sensor data fusion for body state estimation in a hexapod robot with dynamical gaits," *IEEE Transactions on Robotics*, vol. 22, no. 5, pp. 932–943, October 2006.

[28] M. Ermes, J. Pärkkä, J. Mäntyjärvi, and I. Korhonen, "Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions," *IEEE Transactions on Information Technology in Biomedicine*, vol. 12, no. 1, pp. 20–26, January 2008.

[29] R. Aylward and J. A. Paradiso, "Sensemble: A wireless, compact, multi-user sensor system for interactive dance," in *Proceedings of the Conference on New Interfaces for Musical Expression*, (Paris, France), pp. 134–139, 4–8 June 2006.

[30] J. Lee and I. Ha, "Real-time motion capture for a human body using accelerometers," *Robotica*, vol. 19, no. 6, pp. 601–610, September 2001.

[31] T. Shiratori and J. K. Hodgins, "Accelerometer-based user interfaces for the control of a physically simulated character," *ACM Transactions on Graphics (SIGGRAPH Asia 2008)*, vol. 27, no. 5, December 2008.

[32] B. Barshan, *Gyroscopes*, vol. 8 of *Wiley Encyclopedia of Electrical and Electronics Engineering*, pp. 545–567. Hoboken, NJ, U.S.A.: John Wiley & Sons, Inc., 1999.

[33] D. T. Greenwood, *Principles of Dynamics*. Upper Saddle River, EUA, U.S.A.: Prentice Hall, 2nd ed., 1988.

[34] E. T. Hall, "The use of the proton magnetometer in underwater archaeology," *Archaeometry*, vol. 9, no. 1, pp. 32–43, June 1966.

[35] Y. Yang, "Magnetometer and differential carrier phase GPS-aided INS for advanced vehicle control," *IEEE Transactions on Robotics and Automation*, vol. 19, no. 2, pp. 269–282, April 2003.

[36] F. Martel, P. K. Pal and M. L. Psiaki, "Three-axis attitude determination via Kalman filtering of magnetometer data," in *Flight Mechanics/Estimation Theory Symposium*, (NASA Goddard Space Flight Center, MD, U.S.A.), pp. 344–367, May 1988.

[37] IEEE, *IEEE Std. 952-1997 Standard Specification Format Guide and Test Procedure for Single-Axis Interferometric Fiber Optic Gyros*. 2008.

[38] VectorNav Technologies, "Inertial navigation." `http://www.vectornav.com/index.php?option=com_content&view=article&id=21&Itemid=11`.

[39] Microstrain, "Microstrain Little Sensors, Big Ideas."

[40] Xsens, "XSens: 3D Motion Tracking." `http://www.xsens.com`.

[41] Xsens, "XSens Xbus Kit: Measurement of human motion." `http://www.xsens.com/en/general/xbus-kit`.

[42] 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, pp. 1897–1907, July 2007.

[43] P. Aggarwal, Z. Syed, X. Niu and N. El-Sheimy, "Cost-effective testing and calibration of low cost MEMS sensors for integrated positioning, navigation and mapping systems," in *XXIII FIG Congress*, (Munich, Germany), October 2006.

[44] H. Zhang, Y. Wu, W. Wu, M. Wu and X. Hu, "Improved multi-position calibration for inertial measurement units," *Measurement Science and Technology*, vol. 21, no. 1, pp. 1–11, January 2010.

[45] I. Skog and P. Handel, "Calibration of a MEMS inertial measurement unit," in *XVII. Imeko World Congress*, (Rio de Janeiro, Brazil), September 2006.

[46] J. K. Bekkeng, "Calibration of a novel MEMS inertial reference unit," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 6, pp. 1967–1974, June 2009.

[47] S. Levy, A. E. McPherson and E. V. Hobbs, "Calibration of accelerometers," *Journal of Research of the National Bureau of Standards*, vol. 41, pp. 359–369, November 1948.

[48] A. Kourepenis, "Performance of MEMS inertial sensors," in *IEEE Position Location and Navigation Symposium*, (Palm Springs, CA, U.S.A.), pp. 1–8, April 1998.

[49] L. Ojeda, H. Chung and J. Borenstein, "Precision-calibration of fiber-optics gyroscopes for mobile robot navigation," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3, (San Francisco, CA , U.S.A.), pp. 2064–2069, April 2000.

[50] V. Renauding, M. H. Afzal and G. Lachapelle, "Complete triaxis magnetometer calibration in the magnetic domain," *Journal of Sensors*, vol. 2010, pp. 359–369, November 2010.

[51] J. F. Vasconcelos, G. Elkaim, C. Silvestre, P. Oliveira and B. Cardeira, "Geometric approach to strapdown magnetometer calibration in sensor frame," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 47, no. 2, pp. 1293–1306, April 2011.

[52] D. Gebre-egziabher, G. H. Elkaim, J. D. Powell and B. W. Parkinson, "A non-linear, two-step estimation algorithm for calibrating solid-state strapdown magnetometers," in *Proceedings of Eighth International St. Petersburg Conference on Navigation Systems (IEEE/AIAA)*, (St. Petersburg, Russia), pp. 28–37, May 2001.

[53] C. C. Foster and G. H. Elkaim, "Extension of a two-step calibration methodology to include nonorthogonal sensor axes," *IEEE Transactions*

on Aerospace and Electronic Systems, vol. 44, no. 3, pp. 1070–1078, July 2008.

[54] J. C. Lotters, J. Schipper, P. H. Veltink, W. Olthuis and P. Bergveld, "Procedure for in-use calibration of triaxial accelerometers in medical applications," *Sensors and Actuators A: Physical*, vol. 68, no. 1–3, pp. 221–228, June 1998.

[55] A. Saxena, G Gupta, V. Gerasimov and S. Ourselin, "In use parameter estimation of inertial sensors by detecting multilevel quasi-static states," in *Proceedings of the 9th International Conference on Knowledge Based Intelligent Information and Engineering Systems*, vol. 4, (Melbourne, AU), pp. 595–601, 2005.

[56] 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, pp. 2980–2988, August 2008.

[57] E. M. Nebot and H. F. Durrant-Whyte, "Initial calibration and alignment of low-cost inertial navigation units for land vehicle applications," *Journal of Robotic Systems*, vol. 16, no. 2, pp. 81–92, February 1999.

[58] K. P. Schwarz, M. A. Chapman, M. W. Cannon and P. Gong, "An integrated INS/GPS approach to the georeferencing of remotely sensed data," *Photogrammetric engineering and remote sensing*, vol. 59, no .11, pp. 1667–1674, November 1993.

[59] P. S. Maybeck, *Stochastic Models, Estimation and Control*. New York, U.S.A.: Academic Press, 1979.

[60] W. Li, "Simultaneous registration and fusion of multiple dissimilar sensors for cooperative driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 5, no .2, pp. 84–98, June 2004.

[61] M. D. Cecco, "Sensor fusion of inertial-odometric navigation as a function of the actual manoeuvres of autonomous guided vehicles," *Measurement Science and Technology*, vol. 14, no. 5, May 2003.

[62] S. Alban, D. M. Akos, S. M. Rock, and D. Gebre-Egziabher, "Performance analysis and architectures for INS-aided GPS tracking loops," in *Institute of Navigation's National Technical Meeting*, (Anaheim, CA, U.S.A.), pp. 611–622, January 2003.

[63] B. M. Scherzinger, "Precise robust positioning with inertial/GPS RTK," in *Proceedings of the 13th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GPS 2000)*, (Alexandria, VA, U.S.A.), pp. 155–162, September 2000.

[64] S. Sukkarieh, E. M. Nebot and H. F. Durrant-Whyte, "A high integrity IMU/GPS navigation loop for autonomous land vehicle applications," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 3, pp. 572–578, June 1999.

[65] B. Vik and T. I. Vossen, "A nonlinear observer for GPS and INS integration," in *Proceedings of the 40th IEEE Conference on Decision and Control*, vol. 3, (Orlando, FL, U.S.A.), pp. 2956–2961, December 2001.

[66] A. H. Mohamed and K. P. Schwarz, "Adaptive Kalman filtering for INS/GPS," *Journal of Geodesy*, vol. 73, no. 4, pp. 199–203, December 1998.

[67] D. W. Allan, "Statistics of atomic frequency standards," in *Proceedings of the IEEE*, vol. 54, pp. 221–230, February 1966.

[68] M. Tehrani, "Ring laser gyro data-analysis with cluster sampling technique," in *Proceedings of the SPIE the International Society for Optical Engineering*, vol. 412, (Arlington, VA, U.S.A.), pp. 207–220, 1983.

[69] H. Hou, "Modeling of inertial sensors using Allan variance," Master's thesis, Department of Geomatics Engineering, University of Calgary, Alberta, Canada, 2004.

[70] C. R. Kochakian, "Time-domain uncertainty charts (green charts): A tool for validating the design of IMU/instrument interfaces," in *Proceedings of AIAA Guidance and Control Conference*, (Danvers, MA, U.S.A.), August 1980.

[71] A. J. Van Dierendonck, J. B. McGraw and R. G. Brown, "Relationship between Allan variances and Kalman Filter parameters," in *Proceedings of the 16th Annual Precise Time and Time Interval Systems and Applications Meeting*, (NASA Goddard Space Flight Center, MD, U.S.A.), pp. 273–293, November 1984.

[72] D. W. Allan and J. A. Barnes, "A modified "Allan variance" with increased oscillator characterization ability," in *Thirty Fifth Annual Frequency Control Symposium*, pp. 470–475, May 1981.

[73] D. A. Howe, "An extension of the Allan variance with increased confidence at long term," in *Proceedings of the 49th IEEE International Frequency Control Symposium*, (San Francisco, CA, U.S.A.), pp. 321–329, May-June 1995.

[74] 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, no. 1, pp. 140–149, January 2008.

[75] J. Z. Sasiadek, "Fuzzy adaptive Kalman filtering for INS/GPS data fusion," in *Proceedings of the 2000 IEEE International Symposium on Intelligent Control*, (Rio Patras, Greece), pp. 181–186, July 2000.

[76] C. Hide, T. Moore and M. Smith, "Adaptive Kalman filtering for low-cost INS/GPS," *The Journal of Navigation*, vol. 56, no .1, pp. 143–152, January 2003.

[77] C. Hu, W. Chen, Y. Chen and D. Liu, "Adaptive Kalman filtering for vehicle navigation," *Journal of Global Positioning Systems*, vol. 2, no. 1, pp. 42–47, 2003.

[78] A. M. Sabatini, "A wavelet-based bootstrap method applied to inertial sensor stochastic error modeling using the Allan variance," *Measurement Science and Technology*, vol. 17, no. 11, pp. 2980–2988, November 2006.

[79] N. El-Sheimy, "Wavelet de-noising for IMU alignment," *IEEE Aerospace and Electronic Systems Magazine*, vol. 19, no. 10, pp. 32–39, October 2004.

[80] P. Minha, *Error Analysis and Stochastic Modeling of MEMS based Inertial Sensors for Land Vehicle Navigation Applications*. PhD thesis, Department of Geomatics Engineering, University of Calgary, Alberta, Canada, 2004.

[81] C. Konvalin, "Compensating for Tilt, Hard-Iron, and Soft-Iron Effects." `http://www.sensorsmag.com/sensors/motion-velocity-displacement/compensating-tilt-hard-iron-and-soft-iron-effects-6475`.

[82] D. Gebre-Egziahber, G. H. Elkaim, J. D. Powell and B. W. Parkinson, "Calibration of strapdown magnetometers in magnetic field domain," *Journal of Aerospace Engineering*, vol. 19, no. 2, pp. 87–102, April 2006.

[83] J. Vcelak, P. Ripka, J. Kubik, A. Platil and P. Kaspar, "AMR navigation systems and methods of their calibration," *Sensors and Actuators A: Physical*, vol. 123, pp. 122–128, September 2005.

[84] Z. C. Wu, Z. F. Wang and Y. Ge, "Gravity based online calibration for monolithic triaxial accelerometers' gain and offset drift," in *Proceedings of the 4th World Congress on Intelligent Control and Automation*, vol. 3, (Shanghai, China), pp. 2171–2175, June 2002.

[85] J. Borenstein, "Experimental evaluation of a fiber optics gyroscope for improving dead reckoning accuracy in mobile robots," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 4, (Leuven, Belgium), pp. 3456–3461, May 1998.

[86] E. Dorveaux, D. Vissiere, A. Martin and N. Petit, "Iterative calibration method for inertial and magnetic sensors," in *Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference*, (Shangai, China), pp. 8296–8303, December 2009.

[87] W. Gander, G. H. Golub and R. Strebel, "Least-squares fitting of circles and ellipses," *Bit Numerical Mathematics*, vol. 34, no. 4, pp. 558–578, 1994.

[88] S. Bonneta, C. Bassompierrea, C. Godina, S. Lesecqa and A. Barraud, "Calibration methods for inertial and magnetic sensors," *Sensors and Actuators A: Physical*, vol. 156, no. 2, pp. 302–311, December 2009.

[89] Wikipedia, "Levenberg-Marquardt algorithm - Wikipedia, the free encyclopedia." http://en.wikipedia.org/wiki/Levenberg%E2%80%93Marquardt_algorithm.

[90] Y. Shi and C. Eberhart, "Parameter selection in particle swarm optimization," *Evolutionary Programming VII Lecture Notes in Computer Science*, vol. 1447, pp. 591–600, 1998.

[91] Y. Shi and C. Eberhart, "Empirical study of particle swarm optimization," vol. 3, (Washington, DC, U.S.A.), July 1999.

[92] National Geophysical Data Center, "World Magnetic Model - Home Page." http://www.ngdc.noaa.gov/geomag/WMM/DoDWMM.shtml.

[93] M. S. Bielas, "Stochastic and dynamic modeling of fiber gyros," in *Proceedings of XII SPIE Fiber Optic and Laser Sensors*, vol. 2292, pp. 240–254, November 1994.

[94] M. S. Keshner, "$1/f$ noise," *Proceedings of the IEEE*, vol. 70, no. 3, pp. 212–218, March 1982.

[95] F. N. Hooge, "$1/f$ noise sources," vol. 41, no. 11, pp. 1926–1935, November 1994.

[96] K. Mathiassen, "Positioning with inertial sensors and aid Sensors when losing GPS Fix," Master's thesis, Engineering Cybernetics, Norwegian University of Science and Technology, Norway, 2010.

[97] S. Han, J. Wang and N. Knight, "Using Allan variance to determine the calibration model of inertial sensors for GPS/INS integration," in *6th International Symposium on Mobile Mapping Technology*, (Presidente Prudente, Sao Paulo, Brazil), July 2009.

[98] N. J. Kasdin, "Discrete simulation of colored noise and stochastic processes and $1/f^{\alpha}$ power law noise generation," in *Proceedings of the IEEE*, vol. 83, pp. 802–827, May 1995.

[99] Wikipedia, "Discretization - Wikipedia, the free encyclopedia." http://en.wikipedia.org/wiki/Discretization.

[100] Wikipedia, "Numerical ordinary differential equations - Wikipedia, the free encyclopedia." http://en.wikipedia.org/wiki/Numerical_ordinary_differential_equations.

[101] S. M. Seong, J. G. Lee and C. G. Park, "Equivalent ARMA model representation for RLG random errors," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 36, no. 1, pp. 286–290, January 2000.

[102] D. R. Osborn, "Exact and approximate maximum likelihood estimators for vector moving average processes," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 114–118, 1977.

[103] Y. Stebler, S. Guerrier, J. Skaloud, and M.-P. V.-Feser, "Constrained expectation-maximization algorithm for stochastic inertial error modeling: study of feasibility," vol. 22, no. 8, August 2011.

[104] M. I. A. Lourakis, "A Brief Description of the Levenberg-Marquardt Algorithm implemented by levmar," technical report, Institute of Computer Science, Foundation for Research and Technology - Hellas (FORTH), Crete, Greece, February 2005.

[105] K. Madsen, H. B. Nielsen and O. Tingleff, "Methods for non-linear least squares problems," lecture note, Department of Informatics and Mathematical Modelling, Technical University of Denmark, April 2004.

[106] M. I. A. Lourakis and A. A. Argyros, "Is Levenberg-Marquardt the most efficient optimization algorithm for implementing bundle adjustment," in *Proceedings of the 10th IEEE International Conference on Computer Vision*, vol. 2, (Beijing, China), pp. 1526–1531, October 2005.

[107] J. Kennedy and R Eberhart, "Particle swarm optimization," in *IEEE International Conference on Neural Networks*, vol. 4, (Washington, DC, U.S.A.), pp. 1942–1948, November 1995.

[108] Wikipedia, "Particle swarm optimization - Wikipedia, the free encyclopedia." http://en.wikipedia.org/wiki/Particle_swarm_optimization.

[109] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Proceedings of the 2000 Congress on Evolutionary Computation*, vol. 1, (La Jolla, CA , U.S.A.), pp. 84–88, July 2000.

[110] Wikipedia, "Gradient descent - Wikipedia, the free encyclopedia." `http://en.wikipedia.org/wiki/Gradient_descent`.

[111] E. Polak, *Optimization: Algorithms and Consistent Approximations*. New York, U.S.A.: Springer, 1997.

[112] R. D. Leone, M. Gaudioso and L. Grippo, "Stopping criteria for line search methods without derivatives," *Mathematical Programming*, vol. 30, no. 3, pp. 285–300, 1984.

[113] L. Armijo, "Minimization of functions having Lipschitz continuous first partial derivatives," *Pacific Journal of Mathematics*, vol. 16, no. 1, 1966.

[114] Wikipedia, "Wolfe conditions - Wikipedia, the free encyclopedia." `http://en.wikipedia.org/wiki/Wolfe_conditions`.

[115] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions: with Formulas, Graphs, and Mathematical Tables*. Mineola, NY, U.S.A.: Courier Dover Publications, 9th ed., 1965.

Appendix A

Levenberg-Marquardt Algorithm (LMA)

LMA is a nonlinear optimization technique commonly used in a broad range of applications. It tries to estimate the optimal parameter set of a function, denoted by $\vec{\theta}^*$, iteratively by minimizing the sum of the squared distance between the actual and estimated outputs of that function, given $\vec{\theta}$, as shown in Equation (A.1). That function intends to model the relation between the true output and the input corresponding to it. The notation used in this subsection can be explained as follows: We use $f(.)$ and $\vec{y}$ to denote the function, whose parameters are unidentified, and the true output vector of the function, respectively. The operator $f(.)$ generates the estimated output vector, given the unknown parameter set $\vec{\theta}$.

$$\vec{\theta}^* = \arg\min_{\vec{\theta}} \left|\left|\vec{y} - f(\vec{\theta})\right|\right| \tag{A.1}$$

LMA is considered to be a hybrid of the steepest descent and Gauss-Newton methods [104]. It converges to the minima by updating the current solution vector, $\vec{\theta}$, through a series of augmented least-squares problems formed after linearization of $f(.)$ in the neighborhood of $\vec{\theta}$. This process is detailed below.

- $f\vec{\theta})$ is linearized around $\vec{\theta}$ by ignoring the second- and higher-order terms

of its Taylor series expansion as

$$f(\vec{\theta}+\delta\vec{\theta}) \approx f(\vec{\theta}) + \frac{\partial f(\vec{\theta})}{\partial \vec{\theta}}\delta\vec{\theta} \tag{A.2}$$

- Normal equations related to Equation (A.2) can be formed by

$$\boldsymbol{J^T J}\ \delta\vec{\theta} = \boldsymbol{J^T}\left(\vec{y} - f(\vec{\theta})\right), \tag{A.3}$$

  where $\boldsymbol{J} = \dfrac{\partial f(\vec{\theta})}{\partial \vec{\theta}}$.

- LMA solves a slight variation of the conventional normal equations, given in Equation (A.3), which is called the augmented normal equations. They are expressed in Equation (A.5) and are used to update $\vec{\theta}$ according to

$$\vec{\theta} = \vec{\theta} + \delta\vec{\theta} \tag{A.4}$$

$$\delta\vec{\theta} = (\boldsymbol{J^T J} + \mu\boldsymbol{I})^{-1}\boldsymbol{J^T}\left(\vec{y} - f(\vec{\theta})\right) \tag{A.5}$$

  Here, the parameter μ denotes the scalar damping term related to LMA and is updated at each iteration according to the internal logic of LMA as well.

- LMA stops when any of the following occurs:
  - the preset upper bound for the number of iterations is reached,
  - the solution cannot be improved sufficiently,
  - the fitting error drops below a certain level.

Pseudocode of the LMA algorithm is given below [104]. The input parameters ϖ, ε_1, and ε_2 are related to the configuration settings used to provide a proper LMA operation. The parameter ϖ is about the initialization of the algorithm whereas ε_1 and ε_2 are the termination condition parameters. The choice of ϖ depends on the quality of the initial parameter guess $\vec{\theta}_0$. As a rule of thumb, a small value (e.g., 10^{-6}) is used if $\vec{\theta}_0$ is believed to be a fair initial guess. Otherwise, ϖ should be set to a value around one. Furthermore, ε_1 and ε_2 are generally set to very small values like 10^{-10} to guarantee that the algorithm has converged. More detailed discussion pertaining to LMA can be found in [104, 105, 106].

Algorithm 1 $\vec{\theta}^*$=LMA($\vec{\theta}_0, \varpi, \varepsilon_1, \varepsilon_2$)

$k \leftarrow 0$
$\vec{\theta} \leftarrow \vec{\theta}_0$
$\mu \leftarrow \varpi \max \boldsymbol{J}^T\boldsymbol{J}$
$\boldsymbol{J} \leftarrow \partial f(\vec{\theta})/\partial\vec{\theta}$
while $k \geq k_{max}$ **or** $stop = 0$ **do**
  $k \leftarrow k + 1$
  $\delta\vec{\theta} \leftarrow (\boldsymbol{J}^T\boldsymbol{J} + \mu I)^{-1}\boldsymbol{J}^T(\vec{y} - f(X, \vec{\theta}))$
  if $\| \delta\vec{\theta} \| \leq \varepsilon_2(\| \vec{\theta} \| + \varepsilon_2)$ **then**
    $stop \leftarrow 1$
  else
    $\vec{\theta}_{new} \leftarrow \vec{\theta} + \delta\vec{\theta}$
    $\varrho \leftarrow (\| \vec{y} - f(\vec{\theta}) \|^2 - \| \vec{y} - f(\vec{\theta}_{new}) \|^2)/(\delta\vec{\theta}^T(\mu\delta\vec{\theta} + \delta\vec{\theta}))$
    if $\varrho > 0$ **then**
      $\vec{\theta} \leftarrow \vec{\theta}_{new}$
      $\boldsymbol{J} \leftarrow \partial f(\vec{\theta})/\partial\vec{\theta}$
      if $\| \boldsymbol{J}^T(\vec{y} - f(\vec{\theta})) \|_\infty \leq \varepsilon_1$ **then**
        $stop \leftarrow 1$
      end if
      $\mu \leftarrow \max(\frac{1}{3}, 1 - (2\varrho - 1)^3)$
    else
      $\mu \leftarrow 2\mu$
    end if
  end if
end while
$\vec{\theta}^* \leftarrow \vec{\theta}$

When adopting the LMA for the deterministic calibration of inertial sensors and magnetometers, $f(.)$ is selected according to one of the sensor model equations (Equations (2.5) and (2.8)). Then, $\vec{\theta}$ is the set of unknown calibration parameters involved in the model (bias error, scale factor error, misalignment, etc.).

Appendix B

Particle Swarm Optimization (PSO)

Though PSO is originally proposed to simulate movements of bird flocks [107], it has become very popular in many research fields (e.g., signal processing) because of its simplicity and the success rate in global convergence for optimization problems. It is generally adopted as an offline optimization tool since it requires a huge amount of computational power. The workflow of the algorithm can be described roughly as the search of particles in the parameter space towards the optimal solution. This optimality is measured according to a quality index which can be described as the cost function $\mathcal{H}(.)$ to be either minimized or maximized. The general PSO problem is portrayed below.

Let S be the number of particles in the swarm, ith of which has a position $\vec{\theta}_i$ and a velocity $\vec{v}_i$ in the search-space $\mathbb{R}^d$. Let $\vec{p}_i$ be the best known position of that particle and $\vec{\theta}^*$ be the best known position of the entire swarm. Pseudo-code of the basic PSO algorithm can be found below [108].

There are several parameters that are of great importance for the proper operation of PSO:

- The parameter m denotes the inertia of the particles having an impact

Algorithm 2 $\vec{\theta}^*$=PSO$(S, \vec{b}_l, \vec{b}_u, \omega, \phi_p, \phi_g, k_{max}, h_{max}, \Delta_{max})$

for all $i \in S$ **do**
  $k \leftarrow 0$
  $\vec{\theta}_i \sim \mathcal{U}\left(\vec{b}_l, \vec{b}_u\right)$
  $\vec{p}_i \leftarrow \vec{\theta}_i$
  if $\mathcal{H}(\vec{p}_i) > \mathcal{H}(\vec{\theta}^*)$ **then**
    $\vec{\theta}^* \leftarrow \vec{p}_i$
  end if
  $\vec{v}_i \sim \mathcal{U}\left(-\left|\vec{b}_u - \vec{b}_l\right|, \left|\vec{b}_u - \vec{b}_l\right|\right)$
end for
repeat
  for all $i \in S$ **do**
    $r_p \sim \mathcal{U}(0, 1)$
    $r_g \sim \mathcal{U}(0, 1)$
    $\vec{v}_i \leftarrow m\,\vec{v}_i + \phi_p\, r_p \left(\vec{p}_i - \vec{\theta}_i\right) + \phi_g\, r_g \left(\vec{\theta}^* - \vec{\theta}_i\right)$
    $\vec{\theta}_i \leftarrow \vec{\theta}_i + \vec{v}_i$
    if $\mathcal{H}(\vec{\theta}_i) > \mathcal{H}(\vec{p}_i)$ **then**
      $\vec{p}_i \leftarrow \vec{\theta}_i$
      if $\mathcal{H}(\vec{\theta}_i) > \mathcal{H}(\vec{\theta}^*)$ **then**
        $\vec{\theta}^* \leftarrow \vec{\theta}_i$
        $k^* \leftarrow k$
      end if
    end if
  end for
  $k \leftarrow k + 1$
until $k < k_{max}$ **or** $k^* - k > \Delta_{max}$

on how much particles tend not to change their current velocities at each iteration.

- The parameter S denotes the population size of the swarm.
- The parameters ϕ_p and ϕ_g determine the amount of the contributions of the entire swarm population's and each particle's own knowledge, respectively while computing the particles' new velocities. They are also called social and global cognitive parameters, respectively.
- The parameters $\vec{b}_l$ and $\vec{b}_u$ represent the lower and upper bounds of the search space. They are constraints of the problem rather than user-set parameters.

- The parameters Δ_{max} and k_{max}, which are the maximum solution update rate and the iteration number limit, define the stopping condition of the algorithm.

The selection of these parameters has been studied extensively in the literature since PSO's performance immensely depends on them. References [90], [91], and [109] are important works and provide insights on choosing the parameter values in practice. It is recommended to set $0.75 < m < 1$ in order to introduce a real dynamic system feature that bird flocks have, into the flow of the optimization. A correlation between the population size parameter S and the dimension of the unknown parameter vector $\vec{\theta}$ is shown for the successful operation. The parameters ϕ_p and ϕ_g are generally set to a value greater than 1. The selection of Δ_{max} and k_{max} is generally based on a trial and error approach.

Appendix C

Gradient-Ascent Optimization (GAO)

Gradient-ascent is a first-order optimization technique that is used to find a local maximum of a given function. It approaches the maximum by updating the current solution in the direction of the gradient of the cost function at each iteration [110]. How much the current solution is updated in the direction of the gradient at each iteration step is usually determined by a line search algorithm [105]. An outline of GAO for the maximization of any $f(\vec{\theta}) : \mathbb{R}^n \rightarrow \mathbb{R}$ is shown below.

The stopping criteria of the algorithm compares the progress of the solution and checks if the number of iterations has reached the k_{max}, which is the maximum allowed iteration number. The parameters ϵ_1 and ϵ_2, which are used in the stopping criteria as well, are considered to be the minimum acceptable step length and the cost function improvement. Furthermore, the *steplength* subfunction in the pseudo-code of GAO determines the amount of change of $\vec{\theta}$ in the direction of the gradient at each iteration step. The best α value is to provide the maximum amount of increase in the function's value. However, the solution of the optimal α value requires the use of optimization techniques and is usually considered to be out of GAO's scope. Instead, line search methods are adopted [111, 112]. In

Algorithm 3 $\vec{\theta}^*$=GAO($\vec{\theta}_0, k_{max}, \epsilon_1, \epsilon_2$)

$k \leftarrow 0$
$stop \leftarrow 0$
$\vec{\theta} \leftarrow \vec{\theta}_0$
while $k \leq k_{max}$ **or** $stop = 0$ **do**
  $\vec{\nabla} f \leftarrow \partial f(\vec{\theta})/\partial \vec{\theta}$
  $\hat{\vec{\nabla}} f \leftarrow \vec{\nabla} f / \parallel \vec{\nabla} f \parallel$
  $\alpha \leftarrow steplength(\vec{\theta}, \hat{\vec{\nabla}} f)$
  if $\alpha \leq \epsilon_1$ **and** $f(\vec{\theta} + \alpha \hat{\vec{\nabla}} f) - f(\vec{\theta}) \leq \epsilon_2 f(\vec{\theta})$ **then**
    $stop \leftarrow 1$
  end if
end while
$\vec{\theta}^* \leftarrow \vec{\theta}$

this study, backtracking line search technique based on Armijo condition [113] is used. The final form of the algorithm's work flow is as shown below.

The κ, α_0, and c are configuration parameters of the backtracking line search algorithm and the corresponding Armijo condition, respectively. The parameter κ is associated with the decrease rate in the step length after a step length is rejected. The parameter α_0 is about the initial step length in the beginning of each iteration and c is used in Armijo rule test. References [113, 114] suggest to set $0 < \kappa < 1$ (e.g., $\kappa = 0.8$), $\alpha_0 > 1$, and $c \ll 1$ (e.g., $c = 0.0001$), respectively. Furthermore, ϵ_1 and ϵ_2 are parameters controlling the termination of the algorithm. They are generally set to a small value like 10^{-3} to guarantee that the algorithm has converged.

When adopting the GAO for stochastic identification, $f(.)$ is selected as one of the likelihood functions given in Equations (3.27) and (3.33). Hence, it is obvious that GAO needs the derivative of the likelihood function with respect to $\vec{\theta}$. We decided to utilize a numerical differentiation technique, five-point stensil [115], rather than deriving the complicated differentials of $L\left(\bar{\vec{v}}_{tot}, \vec{\theta}\right)$. Accordingly, the gradient vector $\vec{\nabla} f$ is calculated by

$$\vec{\nabla} f = \frac{L\left(\bar{\vec{v}}_{tot}, \vec{\theta} - 2\vec{\delta}_\theta\right) + 8L\left(\bar{\vec{v}}_{tot}, \vec{\theta} + \vec{\delta}_\theta\right) - L\left(\bar{\vec{v}}_{tot}, \vec{\theta} + 2\vec{\delta}_\theta\right) - 8L\left(\bar{\vec{v}}_{tot}, \vec{\theta} - \vec{\delta}_\theta\right)}{12\vec{\delta}_\theta} \tag{C.1}$$

Algorithm 4 $\vec{\theta}^*$=GAO($\vec{\theta}_0, \alpha_0, \kappa, c, k_{max}, \epsilon_1, \epsilon_2$)

$k \leftarrow 0$
$stop \leftarrow 0$
$\vec{\theta} \leftarrow \vec{\theta}_0$
while $k \leq k_{max}$ **or** $stop = 0$ **do**
  $\alpha \leftarrow \alpha_0$
  $\vec{\nabla} f \leftarrow \partial f(\vec{\theta})/\partial \vec{\theta}$
  $\hat{\vec{\nabla}} f \leftarrow \vec{\nabla} f / \parallel \vec{\nabla} f \parallel$
  if $f(\vec{\theta} + \alpha \hat{\vec{\nabla}} f) \geq f(\vec{\theta}) + \alpha c \, \vec{\nabla} f^T \, \hat{\vec{\nabla}} f$ **then**
    $\alpha = \alpha \kappa$
  end if
  if $\alpha \leq \epsilon_1$ **and** $f(\vec{\theta} + \alpha \hat{\vec{\nabla}} f) - f(\vec{\theta}) \leq \epsilon_2$ **then**
    $stop \leftarrow 1$
  else
    $\vec{\theta} = \vec{\theta} + \alpha \hat{\vec{\nabla}} f$
  end if
end while
$\vec{\theta}^* \leftarrow \vec{\theta}$

where $\vec{\delta}_\theta$ is the spacing between the grid points. As it decreases, accuracy of the numerical differentiation improves.

Appendix D

Sensor Specifications

Datasheets [39, 40] and calibration sheets specific to the units that we procured are provided in the following pages.

Technical Product Overview

3DM-GX2™

Gyro Enhanced Orientation Sensor

Introduction

3DM-GX2™ is a high-performance gyro enhanced orientation sensor which utilizes miniature MEMS sensor technology. It combines a triaxial accelerometer, triaxial gyro, triaxial magnetometer, temperature sensors, and an on-board processor running a sophisticated sensor fusion algorithm.

3DM-GX2™ offers a range of output data quantities from fully calibrated inertial measurements (acceleration, angular rate and magnetic field or deltaAngle & deltaVelocity vectors) to computed orientation estimates (pitch & roll or rotation matrix). All quantities are fully temperature compensated and corrected for sensor misalignment. The angular rate quantities are further corrected for G-sensitivity and scale factor non-linearity to third order.

3DM-GX2's communications interface hardware is contained in a separable module, and can therefore be easily customized. Currently available interface modules include a wireless transceiver, USB 2.0, RS232 and RS422. An OEM version is available without the communications interface enabling the sensor to be integrated directly into a host system's circuitboard, providing a very compact sensing solution.

Features & Benefits

- small, light-weight, low-power design ideal for size-sensitive applications including wearable devices
- fully temperature compensated over entire operational range
- calibrated for sensor misalignment, gyro G-sensitivity, and gyro scale factor non-linearity
- simultaneous sampling for improved time integration performance
- available with wireless and USB communication interfaces
- user adjustable data rate (1 to 250Hz) and sensor bandwith (1 to 100Hz)
- outputs include Euler angles, rotation matrix, deltaAngle & deltaVelocity, acceleration and angular rate vectors

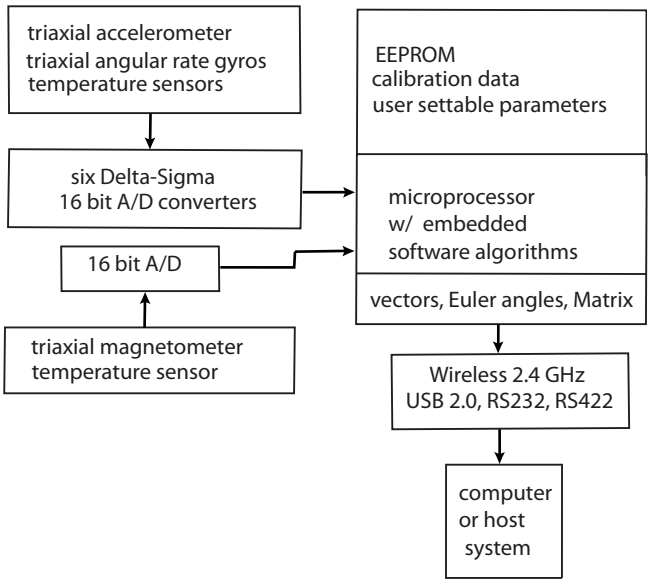
Applications

- inertial aiding INS and GPS, location tracking
- unmanned vehicles, robotics – navigation, artificial horizon
- computer science, biomedical – animation, linkage free tracking/control
- platform stabilization
- antenna and camera pointing

BEST OF sensors expo CHICAGO 2005 GOLD

The system architecture has been carefully designed to substantially eliminate common sources of error such as hysteresis induced by temperature changes and sensitivity to supply voltage variations. The use of six independent Delta-Sigma A/D converters (one for each sensor) ensures that all sensors are sampled simultaneously, and that the best possible time integration results are achieved. On-board coning and sculling compensation allows for use of lower data output rates while maintaining performance of a fast internal sampling rate.

3DM-GX2 incorporates an integral triaxial magnetometer; optionally, the magnetometer can be located remotely to reduce hard and soft iron interference.



Specifications

Orientation range (pitch, roll, yaw)	360° about all axes
Accelerometer range	accelerometers: ± 5 *g* standard ± 10 *g* and ± 2 *g* also available
Accelerometer bias stability	± 0.010 *g* for ± 10 *g* range ± 0.005 *g* for ± 5 *g* range ± 0.003 *g* for ± 2 *g* range
Accelerometer nonlinearity	0.2%
Gyro range	gyros: ± 300°/sec standard, ± 1200°/sec, ± 600°/sec, ± 150°/sec, ± 75°/sec also available
Gyro bias stability	± 0.2°/sec for ± 300°/sec
Gyro nonlinearity	0.2%
Magnetometer range	± 1.2 Gauss
Magnetometer nonlinearity	0.4%
Magnetometer bias stability	0.01 Gauss
A/D resolution	16 bits
Orientation Accuracy	± 0.5° typical for static test conditions ± 2.0° typical for dynamic (cyclic) test conditions & for arbitrary orientation angles
Orientation resolution	<0.1° minimum
Repeatability	0.20°
Output modes	acceleration and angular rate, deltaAngle and deltaVelocity, Euler angles, rotation matrix
Interface options	RS232, RS422, USB 2.0 and wireless - 2.45 GHz IEEE 802.15.4 direct sequence spread spectrum, license free worldwide (2.450 to 2.490 GHz) - 16 channels
Wireless communication range	70 m
Digital output rates	1 to 250 Hz with USB interface 1 to 100 Hz with wireless interface
Serial data rate	115200 bps
Supply voltage	5.2 to 9.0 volts
Power consumption	90 mA
Connectors	micro DB9
Operating temp.	-40 to +70°C with enclosure -40 to +85°C without enclosure
Dimensions	41 mm x 63 mm x 32 mm with enclosure 32 mm x 36 mm x 24 mm without enclosure
Weight	39 grams with enclosure, 16 grams without enclosure
Shock limit	1000 *g* (unpowered), 500*g* (powered)

MicroStrain®

MicroStrain Inc.
310 Hurricane Lane, Unit 4
Williston, VT 05495 USA
www.microstrain.com

ph: 800-449-3878
fax : 802-863-4093
sales@microstrain.com

Patent Pending

Updated July 13, 2007

MicroStrain®

www.microstrain.com

Certificate of Calibration

This document certifies that the equipment referenced below meets published specifications

Model Name: 3DM-GX2

Model Number: 4200
Serial Number: 3559
Device Options: 4200_3559 Cal Results

Firmware Version: 2113
Calibration Date: 12/22/2009

Calibration Technician: JTH

MicroStrain, Inc.
310 Hurricane Lane
Suite 4
Williston, VT 05495-3211
USA

Phone (802) 862-6629
fax (802) 863-4093

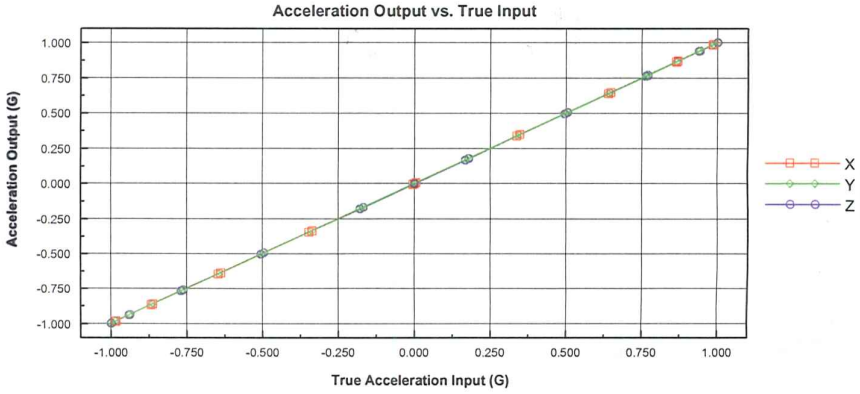
support@microstrain.com

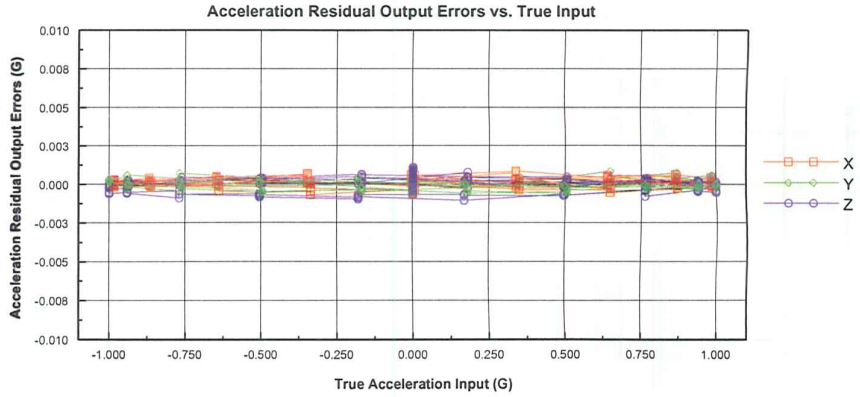
MicroStrain®

Reference Calibration - Acceleration Output Results

Model Name: 3DM-GX2
Model Number: 4200
Serial Number: 3559

Firmware Version: 2113
Calibration Date: 12/22/2009





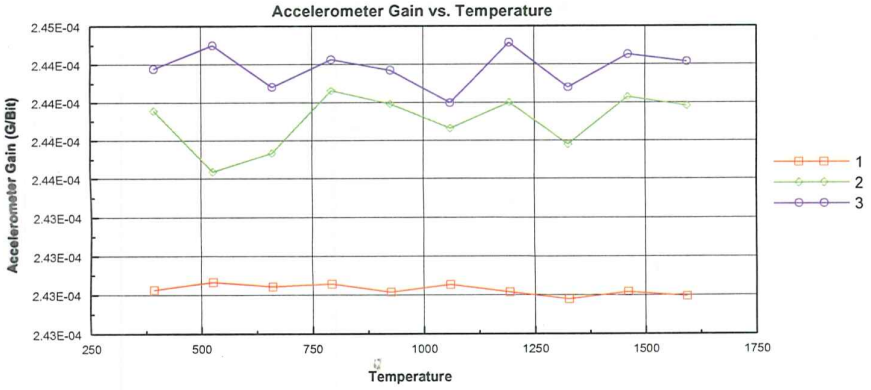
Notes:
1. Reference Calibration carried out at constant temperature (23 deg C nominal).
2. Data collected during rotation of sensor about three orthogonal axes.
3. Residual Output Error gives the difference between the actual system output and the ideal system output.

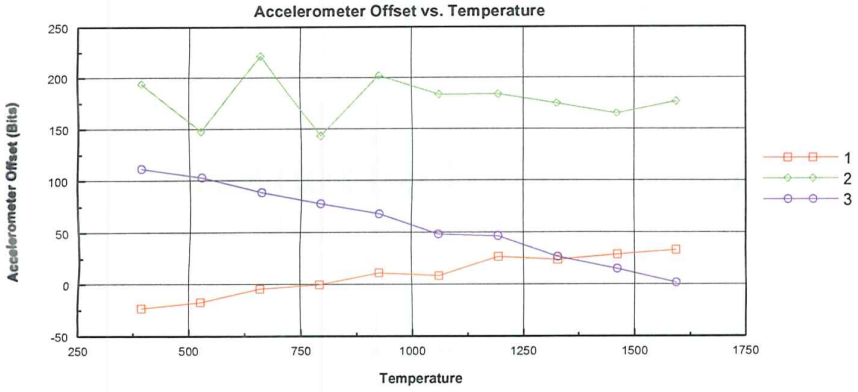
MicroStrain®

Temperature Calibration - Accelerometer Gain & Offset

Model Name: 3DM-GX2
Model Number: 4200
Serial Number: 3559

Firmware Version: 2113
Calibration Date: 12/22/2009





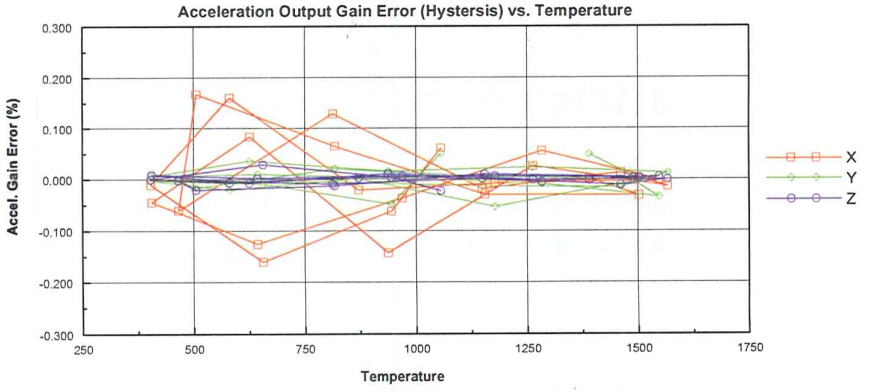
Notes: Temperature Calibration Carried out over -20 to +70 deg C ambient temperature range.

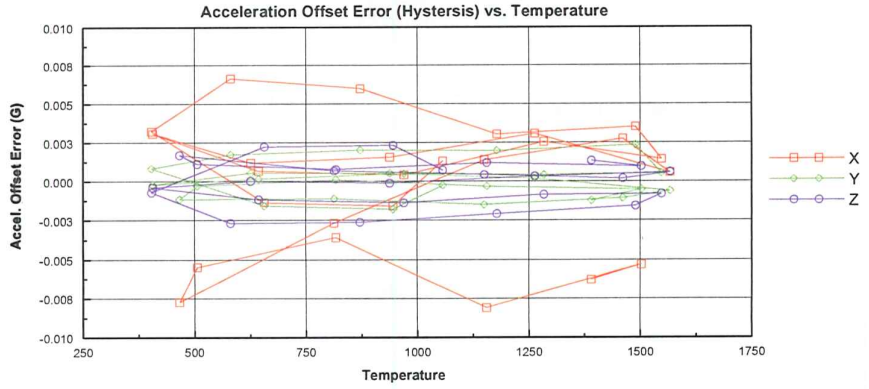
MicroStrain®

Temperature Calibration - Acceleration Gain & Offset Error

Model Name: 3DM-GX2
Model Number: 4200
Serial Number: 3559

Firmware Version: 2113
Calibration Date: 12/22/2009





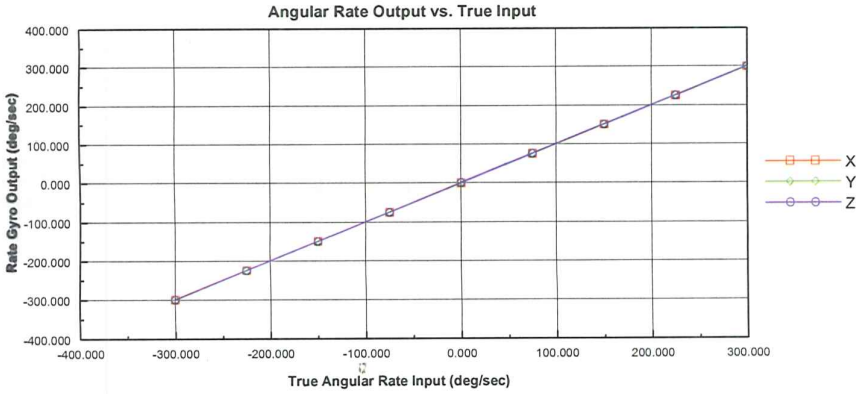
Notes: Temperature Calibration Carried out over -20 to +70 deg C ambient temperature range.

MicroStrain®

Reference Calibration - Rate Gyro Output Results

Model Name: 3DM-GX2
Model Number: 4200
Serial Number: 3559

Firmware Version: 2113
Calibration Date: 12/22/2009



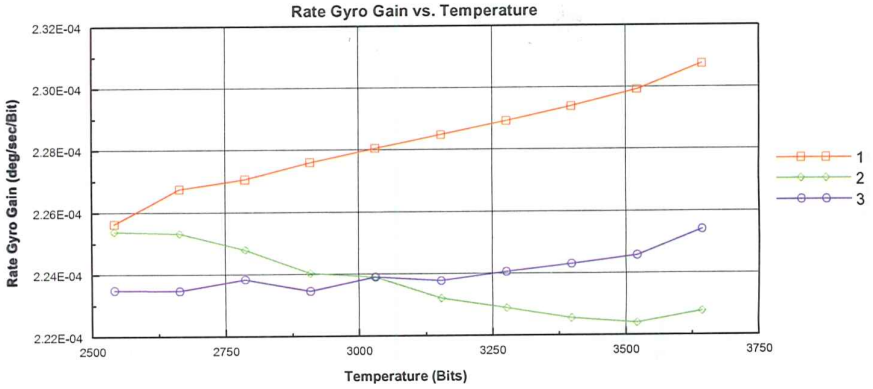
Notes:
1. Reference Calibration carried out at constant temperature (23 deg C nominal).
2. Data collected during rotation of sensor about three orthogonal axes.
3. Residual Output Error gives the difference between the actual system output and the ideal system output.

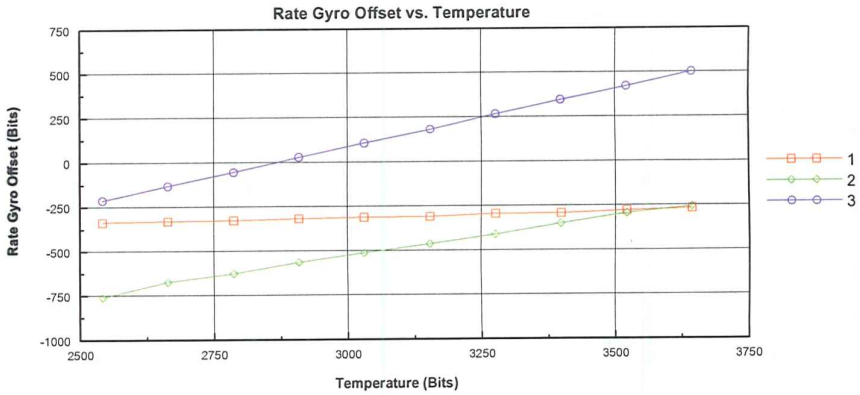
MicroStrain®

Temperature Calibration - Rate Gyro Gain & Offset

Model Name: 3DM-GX2
Model Number: 4200
Serial Number: 3559

Firmware Version: 2113
Calibration Date: 12/22/2009





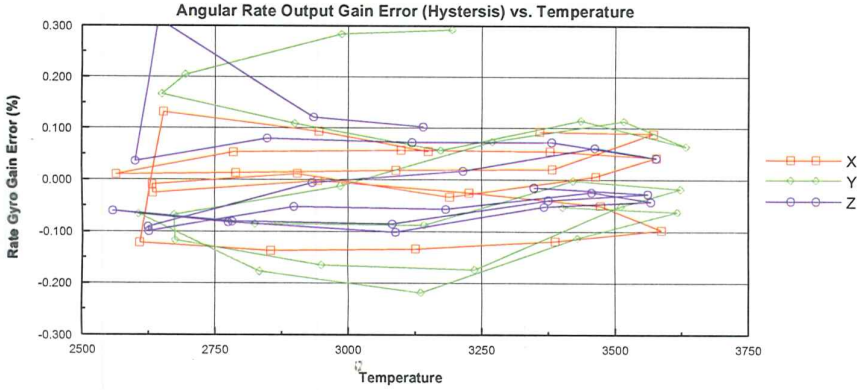
Notes: Temperature Calibration Carried out over -20 to +70 deg C ambient temperature range.

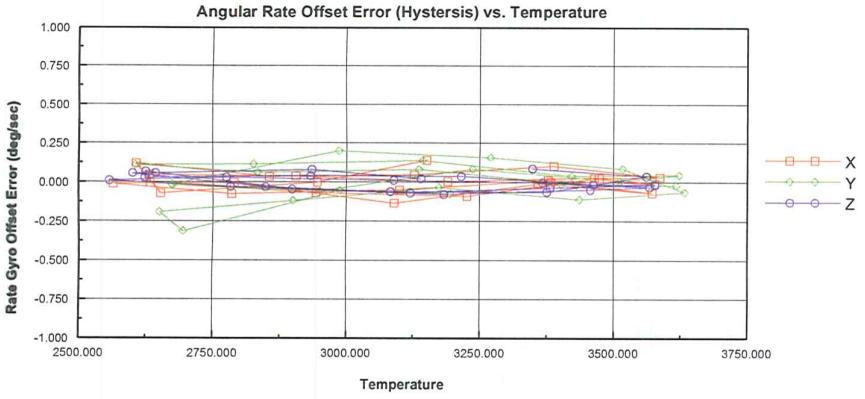
MicroStrain®

Temperature Calibration - Rate Gyro Gain & Offset Error

Model Name: 3DM-GX2
Model Number: 4200
Serial Number: 3559

Firmware Version: 2113
Calibration Date: 12/22/2009





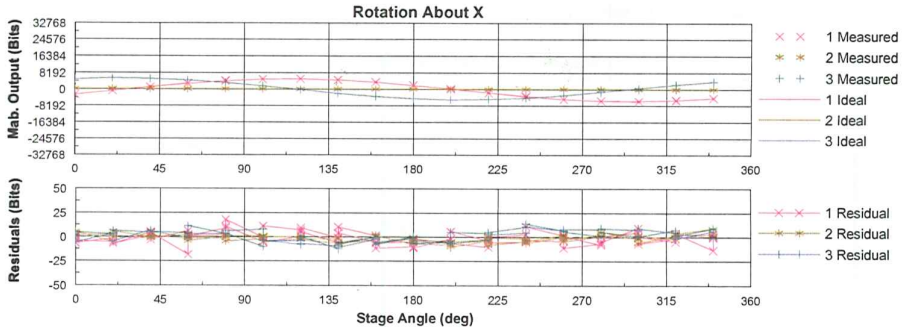
Notes: Temperature Calibration Carried out over -20 to +70 deg C ambient temperature range.

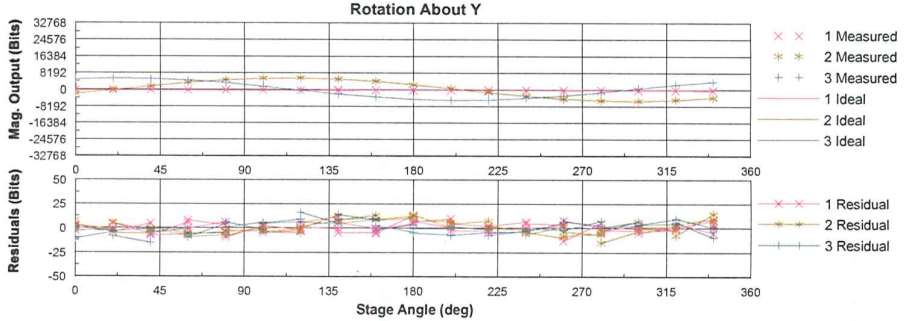
MicroStrain®

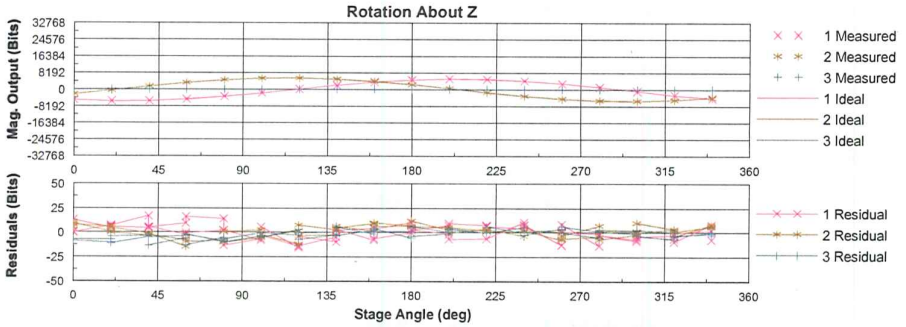
Reference Calibration - Magnetometer Raw Data

Model Name: 3DM-GX2
Model Number: 4200
Serial Number: 3559

Firmware Version: 2113
Calibration Date: 12/22/2009







Notes: 1. Reference Calibration carried out at constant temperature (23 deg C nominal).
2. Data collected during rotation of sensor about three orthogonal axes.
3. Residual Output Error gives the difference between the actual sensor output and the ideal sensor output.

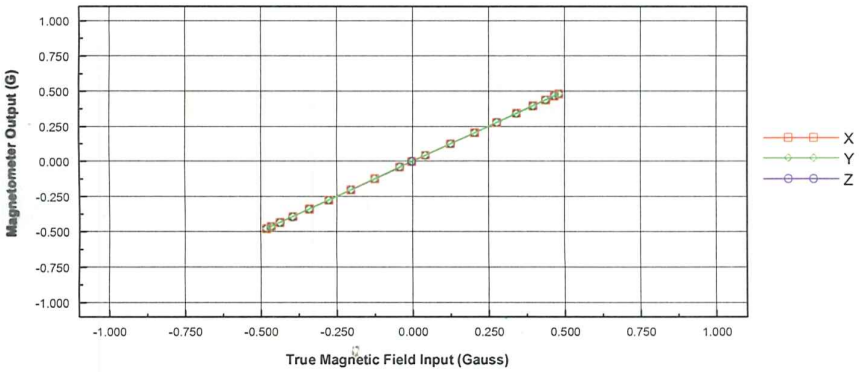
MicroStrain®

Reference Calibration -Magnetometer Output Results

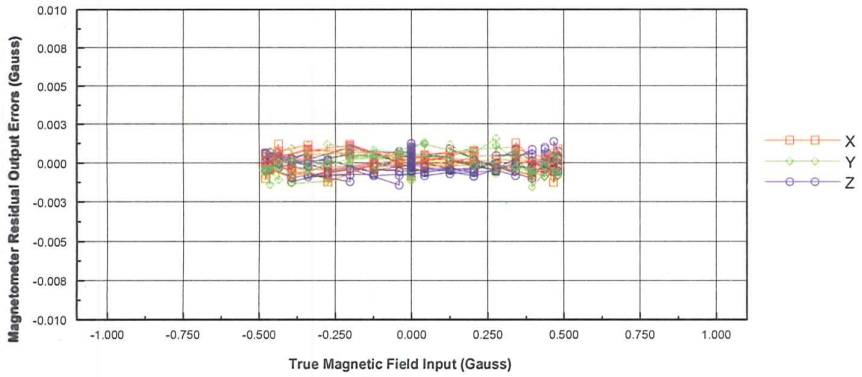
Model Name: 3DM-GX2
Model Number: 4200
Serial Number: 3559

Firmware Version: 2113
Calibration Date: 12/22/2009

Magnetometer Output vs. True Input



Magnetometer Residual Output Errors vs. True Input



Notes:
1. Reference Calibration carried out at constant temperature (23 deg C nominal).
2. Data collected during rotation of sensor about three orthogonal axes.
3. Residual Output Error gives the difference between the actual system output and the ideal system output.

MTx

3DOF ORIENTATION TRACKER

xsens

The MTx is a small and accurate 3DOF Orientation Tracker. It provides drift-free 3D orientation as well as kinematic data: 3D acceleration, 3D rate of turn (rate gyro) and 3D earth-magnetic field. The MTx is an excellent measurement unit for orientation measurement of human body segments and other applications requiring very low profile and light-weight sensor units.

PRODUCT OVERVIEW

Features

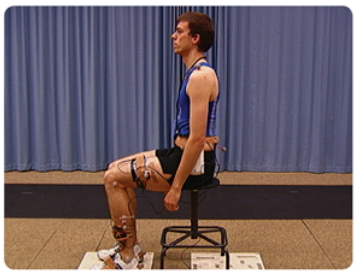
- Accurate full 360 degrees 3D orientation output
- Highly dynamic response combined with long-term stability (no drift)
- 3D acceleration, 3D rate of turn and 3D earth-magnetic field data
- Compact design
- High update rate
- Accepts synchronization pulses
- All solid state miniature MEMS inertial sensors inside
- Individually calibrated for temperature, 3D misalignment and sensor cross-sensitivity

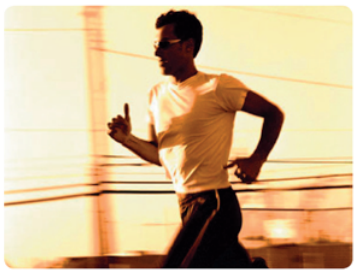
Fields of use

- Biomechanics
- Rehabilitation
- Sports science
- Virtual reality
- Ergonomics
- Animations

The MTx uses 3 rate gyros to track rapidly changing orientations in 3D and it measures the directions of gravity and magnetic north to provide a stable reference. The system's real-time algorithm fuses the sensor information to calculate accurate 3D orientation, with a highly dynamic response which remains stable over prolonged periods. With the MTx Software Development Kit, the MTx can be easily integrated in any system or (OEM) application.

A standalone MTx is available, as well an Xbus version. With the Xbus Master, Xsens' digital data bus, multiple MTx's can easily be used simultaneously, enabling ambulatory and cost-effective measurements of human body motion.





MTx TECHNICAL SPECIFICATIONS

Output
3D orientation (Quaternions/Matrix/Euler angles)
3D acceleration
3D rate-of-turn
3D earth-magnetic field (normalized)
Temperature

Orientation performance

Dynamic Range	all angles in 3D
Angular Resolution[1]	0.05 deg
Static Accuracy (Roll/Pitch)	<0.5 deg
Static Accuracy[2] (Heading)	<1 deg
Dynamic Accuracy[3]	2 deg RMS

Sensor performance	Rate of turn	Acceleration	Magnetic field
Dimensions	3 axes	3 axes	3 axes
Full Scale (standard)	± 1200 deg/s	± 50 m/s²	± 750 mGauss
Linearity	0.1% of FS	0.2% of FS	0.2% of FS
Bias stability[4]	1 deg/s	0.02 m/s²	0.1mGauss
Scale Factor stability[4]	-	0.03%	0.5%
Noise	0.05 deg/s/ √Hz	0.002 m/s²/√Hz	0.5 mGauss
Alignment error	0.1 deg	0.1 deg	0.1 deg
Bandwidth	40 Hz	30 Hz	10 Hz
Max update rate	512 Hz	512 Hz	512 Hz

Interfacing

Max. update rate	512 Hz (calibrated sensor data) 120 Hz (orientation data)
Operating voltage[5]	4.5 - 30 V
Power consumption	360 mW (orientation output)
Digital interface (standard)	RS-232 and USB (external converter) or 'Xbus'

Housing

Dimensions	38x53x21 mm (WxLxH)
Weight	30 g
Ambient temperature operating range[6]	-20... +55 °C
Specified performance operating range[4]	0.. +55 °C

Options and product code

Interface:
RS-232 (RS-232, sync in) 28
RS-485 (RS-485) 48
Xbus 49
(two connectors, only to be used with Xbus Master)

Full Scale Acceleration:
5g (50 m/s²) A53
18g (180 m/s²) A83

Full Scale Rate of Turn:
300 deg/s G35
1200 deg/s G25

Other options on request.
Surcharges may apply.

Product code:	MTx-## A## G##
Standard version:	MTx-28 A53 G25
Standard Xbus version:	MTx-49 A53 G25

[1] 1σ standard deviation of zero-mean angular random walk
[2] in homogenous magnetic environment
[3] may depend on type of motion
[4] deviation over operating temperature range (1σ) specifications subject to change without notice
[5] only valid for MTx's with device ID's > 2000, other units operate on 4.5 - 15 V max
[6] non-condensing environment

xsens

ABOUT XSENS TECHNOLOGIES

Xsens has strong expertise in biomechanics and inertial sensor technology. Thousands of Xsens inertial motion sensors have already been deployed in challenging human and machine motion applications such as motion capture, training & simulation, biomechanics, marine technology and automotive. Xsens' customers include Daimler, PGA, Össur, Roessingh Research and Development, TNO, INAIL, Electronic Arts, Sony Computer Entertainment, and others. The combination of expertise in human motion analysis and innovative inertial motion sensors makes Xsens a leader in inertial human motion capture solutions.

xsens

Xsens Technologies B.V.

phone +31 88 97367 00
fax +31 88 97367 01
e-mail info@xsens.com
internet **www.xsens.com**

Motion Tracker Test & Calibration Certificate

Type name:	MTx
Product ID:	MTx-49A53G25
Device ID:	00327230
Tested on:	18-Mar-2010
Calibrated on:	18-Mar-2010

Niddk Koch

Test Engineer Signature

IMU Specifications	**Accelerometer**		**Rate Gyro**		**Magnetometer**	
Full Scale:	50	[m/s2]	1200	[deg/s]	5	[a.u.]
Bandwidth [Hz]:	30		40		10	
Default Sample Frequency [Hz]:	100					
Default Baudrate [bps]:	115200					

Basic test results	**Accelerometer**		**Rate Gyro**		**Magnetometer**	
Noise:	0.008	[m/s2]	0.006	[rad/s]	0.001	[a.u.]
Static accuracy residual:	0.465	[deg]	Temperature residual:		0.304	[deg]

Calibration Data

Please refer to Technical Documentation for interpretation of values!

Accelerometer

Gains

419	416	421

Offsets

33115	33089	32435

Alignment Matrix

1.00	-0.01	0.00
0.01	1.00	-0.01
-0.01	-0.01	1.00

Temperature Parameters

a	b
0.028	0.002

Rate Gyro

Gains

1114	1124	1135

Offsets

32481	32845	32883

Alignment Matrix

1.00	-0.01	0.00
0.01	1.00	-0.01
0.00	0.00	1.00

Temperature Parameters

a	b
n/a	0.098

Magnetometer

Gains

7301	7518	7445

Offsets

31715	32852	33286

Alignment Matrix

1.00	0.00	-0.06
0.04	1.00	0.04
-0.06	0.00	1.00

Temperature Parameters

a	b
0.353	0.000

Xsens Technologies B.V.

Pantheon 6a
P.O. Box 559
7500 AN Enschede
The Netherlands

phone +31-(0)88-9736700
fax +31-(0)88-9736701
e-mail support@xsens.com
internet www.xsens.com