AALBORG UNIVERSITET

Simon Konge Koldbæk, 20061943
Luminita-Cristiana Totu, 20091520

Improving MEMS Gyroscope Performance using Homogeneous Sensor Fusion

Master's Thesis, May 2011

This page intentionally left blank

Simon Konge Koldbæk, 20061943
Luminita-Cristiana Totu, 20091520

Improving MEMS Gyroscope Performance using Homogeneous Sensor Fusion

Master's Thesis, May 2011

This page intentionally left blank

Improving MEMS Gyroscope Performance using Homogeneous Sensor Fusion,

This report was prepared by
Simon Konge Koldbæk, 20061943
Luminita-Cristiana Totu, 20091520

Supervisors
Associate Professor Anders La Cour-Harbo
Assistant Professor Morten Bisgaard

Release date:	Date published
Category:	1(public)
Edition:	First
Comments:	This report is part of the requirements to achieve the Master of Science in Engineering (M.Sc.Eng.) at Aalborg University. This report represents 58 ECTS points.

Department of Electronic Systems
Section of Automation and Control
Aalborg University
Fredrik Bajers Vej 7
DK - 9220 Aalborg Øst
Denmark

www.es.aau.dk
Tel: (+45) 96 35 87 02
Fax: (+45) 98 15 17 39

This page intentionally left blank

Abstract

The scope of the project is to investigate the possibilities of using noise correlations and Kalman filtering to improve the performance of a sensor array containing multiple MEMS gyroscope. The project is based on the work of Bayard and Ploen whom have showed, trough simulation, that the performance of MEMS gyroscopes can be improved by combining measurements from favorably correlated gyroscopes. In addition, the project also investigates the possibility of identifying noise correlations by using Expectation-Maximization. The project has been proposed and carried out in collaboration with CDL Scotland, which is developer and provider of subsea inertial navigation sensors and solutions. A custom sensor board containing eight medium grade gyroscopes and additional interface hardware has been designed for the project by CDL.

Based on Allan Variance analysis and classical signal analysis methods, a simple stochastic model for the random bias component in the gyroscope output signal has been developed and implemented in MATLAB®. The model has been validated through comparative analysis of the Root Allan Variance. The Expectation-Maximization algorithm has been implemented in and tested in MATLAB®. Several methods of improving the performance of a sensor array containing multiple MEMS gyroscope has been investigated.

Using Kalman filter based estimation strategies and benchmarking against a simple signal averaging filter, the group has shown through simulations that performance improvement, especially in the angle drift estimation, is possible assuming that favorable correlations exist between the noise processes of the gyroscope signals. The Kalman filter algorithm is atypically applied to a state space model that is not observable nor detectable, and an interesting analysis of this usage is discussed. The implemented Expectation-Maximization algorithm was not able to identify all the relevant noise correlations with sufficient accuracy during simulations. The main problem is related to the ratio between the measurement noise and the system noise, as the system noise is two orders of magnitude smaller than the measurement noise. As such, the problem of identifying the noise correlations remains open, and the improvement potential of the gyroscope board can not be assessed in a satisfactory manner at the present time.

This page intentionally left blank

Preface

This Master thesis documents the final two semester project of the Electronics and IT Master degree at Aalborg University, Department of Control Engineering, specialization in Intelligent Autonomous Systems. The thesis has been prepared by Luminita-Cristiana Totu and Simon Konge Koldbæk during the period September 2^{th} 2010 to May 31^{rd} 2011. The thesis investigates the feasibility of using linear modeling and Kalman filtering to improve the performance of multiple Micro Electro Mechanical Sensors (MEMS) gyroscopes by assuming presence of noise correlations.

The project was proposed by associate professor Anders la Cour-Harbo and CDL Scotland. The project has been supervised by Anders Cour-Harbo and assistant professor Morten Bisgaard, while CDL has been a close collaborative partner throughout the project. In addition to providing knowhow and hardware, CDL also invited the group to visit the company production and development office in Aberdeen Scotland during early spring 2011, where a number of practical project issues were resolved.

MATLAB® has been used during the project for data processing and algorithm implementation. The version used is 7.9.0 or equivalent release R2009b. The CD accompanying the thesis contains scripts, functions and algorithms for processing, simulating and filtering, and sensor data files. The CD also contains software for the project hardware.

This page intentionally left blank

Acknowledgments

We, the authors, would like to acknowledge and thank a number of people who have contributed to the completion of this project.

First and foremost we would like to express our appreciation to our supervisors Anders La Cour-Harbo and Morten Bisgaard for their support, guidance, and constructive suggestions during the project. Morten Bisgaard has been a knowledgeable sparing partner for the Kalman filtering problems and Anders La Cour-Harbo has provided clear answers for a wide range of problems.

We would like to express our sincere appreciation to the engineers at CDL who have provided extensive support in relation to the project hardware. Without their technical support and advice the usage of hardware would have been impossible, and the project would have had to depend solely on simulations. We especially thank Nikolaj Fogh for providing hours of technical support and answering questions which have been invaluable to the project and Mads Fogh for sharing his knowledge of MEMS gyroscopes.

Early Spring 2011 we were invited to visit the CDL headquarters in Aberdeen Scotland. The visit was a rewarding experience as we where able to both resolve critical technical issues and perform a series of test on the project hardware. We where also able to enjoy the beautiful city of Aberdeen and engage in more informal discussions with the engineers at CDL.

We would also like to thank David S. Bayard for providing the group with indispensable information regarding the methods presented in the U.S patent [5]. The patent has been found to contain a typographical error in the key equations, and it has been of significant importance to the project that Mr Bayard was able to provide a copy of the original report [4] on which the patent is based.

This page intentionally left blank

Contents

List of Acronyms

ACF Auto Correlation Function . . . 129
ALS Autocovariance Least Squares . . . 96
ARW Angular Random Walk . . . 24
AV Allan Variance . . . 22
CD Compact Disk
DFT Discrete Fourier Transform . . . 131
EM Expectation Maximization . . . 34
FFT Fast Fourier Transform . . . 128
GPS Global Positioning System . . . 2
INS Inertial Navigation System . . . 2
MEMS Micro Electro Mechanical Sensors . . . iii
PC Personal Computer
PSD Power Spectral Density . . . 119
RMS Root Mean Square . . . 127
RRW Rate Random Walk . . . 24
RTS Rauch-Tung-Striebel . . . 33
SPI Serial Peripheral Interface
UAV Unmanned Aerial Vehicle

This page intentionally left blank

Chapter 1

Introduction

In this chapter the project framework and background is addressed to introduce the main elements and issues of the project. After the project preliminaries have been clearly defined, the project proposal is presented as a logical answer to the issues raised in the project preliminaries.

1.1 Background

Navigation has played an important role in the history of mankind and will continue to do so in the future. The tools and methods of navigation have continuously improved and evolved, enabling increasing levels of accuracy. According to Grewal et al. [11], navigation can be divided into five basic types which may be used alone or in unison.

- *Pilotage*, relies on the navigator to identify landmarks and to infer from these the position and the orientation.
- *Dead reckoning*, relies on the navigator to know the vessels initial position and orientation. The current estimated position and orientation is thereafter inferred from the motion of the vessel i.e. heading and speed.
- *Celestial navigation*, relies on the navigators ability to use known celestial objects (e.g. sun, moon, planets, stars) and knowledge of the movements of the Earth to estimate the current position, latitude and longitude, and orientation.
- *Radio navigation*, relies on radio frequency sources with known locations (e.g. global navigation satellite systems or radio beacons).
- *Inertial navigation*, relies on the navigator to know the vessels initial position, velocity and attitude. The estimated position and orientation is then inferred from measuring the attitude rates and accelerations.

Most of the above mentioned navigational techniques make use of the observations and measurements of known locations, patterns or other external references. The identifiable references provide a reference frame which can be considered fixed in space, and to which the navigators position can be related. One type of navigation that does not require any external references after the initial initialization is inertial navigation. The concept of inertial navigation will be addressed in in the following section.

1.1.1 Inertial Navigation

Unlike most navigational techniques, an Inertial Navigation System (INS) requires only to initially be provided with the position and velocity from another navigational system. The INS hereafter measures the linear acceleration and angular velocity of the system and infers an estimated position relative to the initial position. The estimated position is inferred from measurements of the linear acceleration as the second integral of acceleration is position [20]. By using the initial position and double integrating the measured linear acceleration along the three principal axis of the system, the estimated position can be calculated.

The above approach is only valid if the system maintains its initial orientation during the movement. If the system changes orientation during the movement, the estimate will deteriorate rapidly. This problem is accommodated by including the orientation of the system. The orientation of the system is inferred from angular velocity by taking the first integral [20]. Combining the two sets of information enable the INS to provide the navigator with a more accurate position estimate . It is important to notice that the position provided by the INS is only an estimated position as its quality depends on how accurate the linear acceleration and angular velocity can be measured. Methods of measuring linear acceleration and angular velocity and the sensors through which it may be accomplished are addressed in the following section.

Inertial Navigation Sensors

The INS feature different combinations of sensors, but common to all systems are accelerometers and gyroscopes. The accelerometers measure the linear acceleration of the system and are limited in the sense that they can only measure relative to the moving system itself. This means that the accelerometers are not aware of their orientation. The problem of missing orientation is corrected by using gyroscopes which measure the angular velocity of the system.

There are however inherent problems related to using gyroscopes and accelerometers in INS. According to Schmidt [19], the dominant error sources in INS are gyroscope and accelerometer imperfections, insufficient system initialization and imperfections in the models used in the position calculation. One of the main sources of error in the estimated position is integration drift that occurs as the linear acceleration and angular velocity measurements are integrated to determine the linear and angular position respectively. Any errors in the measurements will thus be cumulative and increase indefinitely over time causing the estimated position to deteriorate. Both gyroscopes and accelerometers experience integration drift, but the problem is most significant for gyroscopes.

As a consequence of the integration drift, most types of INS feature some means of external aiding, where the system is periodically reinitialized with position estimates with bounded error. Today a widely used alternative navigation system is Global Positioning System (GPS), which enables the errors to be kept stable and not drift infinitely, Grewal et al. [11].

There are however several application domains which limit or directly render the use of external aiding impossible. In relation to GPS, satellite based aiding is often unreliable due to blocked line of sight, multi path propagation or intentional jamming. When addressing GPS enabled INS, one of the considerations is the free inertial operation time of the system, that is the duration at which external aiding is non-present. Consequently, distinct types of application domains can be identified based on the free inertial operation time. If the external aiding is absent only for a few minutes at a time, the requirements for sensor drift are not as critical as if the signal is lost for several hours or days. To this extent the requirements for low sensor drift increase with operating time. One may consider an automobile using a GPS based navigation system with an INS fall-back, driving through a tunnel. Here, the free inertial operation time is measured in minutes where a submerged submarine can have a free inertial operation time of several months.

Applications with prolonged free inertial operation time require gyroscopes, that provide high accuracy in order to reduce integration drift. The focus of this project is related to applications with prolonged free inertial operation time. In the following section performance of different gyroscope technologies are addressed.

1.1.2 Gyroscope Technologies and Performance

Through history, different technologies have been developed and used to fulfill the increasing demand for accuracy and precision in relation to measuring angular velocity. The different types of gyroscopes can be divided into three main technologies: spinning mass, optical and vibrating gyros. These technologies use respectively the principle of angular momentum conservation, the Sagnac effect and the Coriolis effect. The technological basis of these three distinct types of gyroscopes, with the exception of vibrating gyros, will not be addressed further in this project.

The performance of the three different technologies can be compared in terms of several parameters such as cost, weight, volume, thermal stability and other. In this introductory chapter focus will be placed on bias stability and scale-factor stability, which are the two performance parameters most often used to classify gyroscope performance. Bias stability is a measure of how stable the output of the gyroscope is, relative to identical input, and is measured in $[^\circ/h]$ or $[^\circ/s]$. Scale-factor stability is a measure of how stable the input-output ratio is relative to different inputs, and is usually measured in parts-per-million $[ppm]$ or percent from linear. In general, the smaller the bias and scale-factor stabilities, the better performance can to be expected from the gyroscope. In figure 1.1, the gyroscope bias and scale-factor stability requirements are illustrated in relation to various applications and gyroscope technologies.

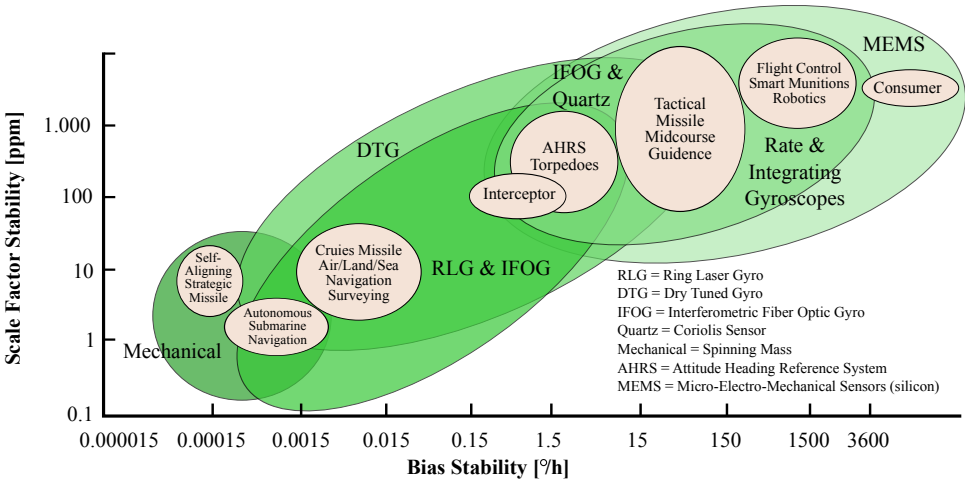

Figure 1.1: Bias stability versus scale factor stability after Schmidt [19]

From figure 1.1 it can be seen that the high-end accuracy region of the graph is dominated by mechanical and optical technologies, that provide extremely low bias stability. There are however also drawbacks related to usage of these high performance gyroscopes. INS that utilize high-end gyroscopes have a large size, which limits their usage to large host vehicles. In addition to being physically large, high-end gyroscopes are expensive, which further restricts the number of feasible applications. In figure 1.2 the costs of INS systems relative to the used gyroscope technology as presented by Schmidt [19] are illustrated. From 1.2, it can be seen that the use of high-end gyroscopes results in expensive INS systems.

From figures 1.1 and 1.2 it can also be observed that cheap INS systems that provide limited accuracy are dominated by Micro Electro Mechanical Sensors (MEMS) technology. MEMS

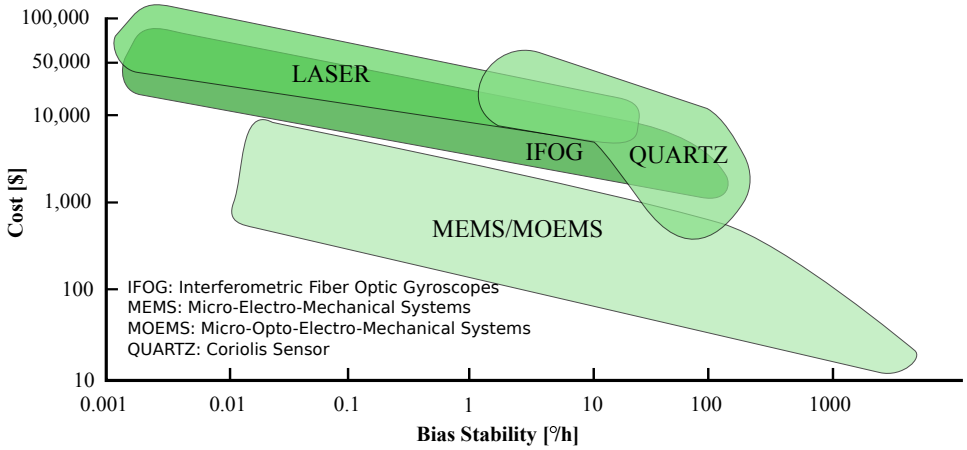

Figure 1.2: System cost versus bias stability after Schmidt [19]

devices have elements ranging in size from one to a hundred microns [*mic*], in average less than the thickness of a human hair. MEMS are made using the basic fabrication techniques and materials of micro electronics. Thin layers of material are deposited onto a base and then selectively etched away leaving a microscopic three dimensional structure. During this process, the mechanical as well as the electrical elements are constructed. The result is a complete system on a single chip which feature both integrated circuitry that provide data processing, active perception and control functions.

The main advantages of MEMS devices are that they can be produced in large quantities, thereby decreasing production cost significantly. The disadvantage is, as illustrated in figure 1.2, that the low production cost and size comes at the price of reduced performance.

1.2 Project Proposal

Part of the the problem-based, project-organized pedagogic model, Aalborg University and the Electronics and IT Master degree program offer students the possibility of working on projects proposed by companies. At the semester start, professors, research staff and companies present different ideas and projects as inspiration for the project that generally accounts for more than half of the study time. One of the projects presented early September 2010 concerned the development of methods for improving the performance of MEMS gyroscopes and developed into a two semesters work, becoming the present master thesis. The project was proposed by associate professor Anders la Cour-Harbo and CDL Scotland. The background for the proposal was a US patent from 2005 [5] that presented a method of improving the performance of MEMS gyroscopes using noise correlations and Kalman filtering.

The group found the project interesting, as it gave the opportunity to work with Kalman filtering which is an essential element in control theory, present in numerous electronic products and used in a very wide range of industries. In addition to working with Kalman filtering, the navigation problem was familiar to both authors from a previous semester project. A final incentive was the fact that the idea of the patent did not appear to be discussed, analyzed and clarified in other easily accessible sources.

The main objective of the project proposal was not only to investigate the patent, but also other methods of improving the performance of MEMS gyroscopes.

The patent [5] is freely available on the Internet and contains the theoretical basis for method to improve the performance of MEMS gyroscopes. The patent is based on and contains references to a more detailed technical report from 2002 [4] which was kindly provided to the group by Mr David S. Bayard. The technical report and patent are important sources

used in this project and can be viewed on the project CD. In the technical report Bayard and Ploen shows through simulation, that significant improvement in MEMS gyroscope drift is achievable using multiple gyroscopes with favorable correlations between gyroscope noise processes. In the following the main concepts of the method presented by Bayard and Ploen is addressed.

The overall method presented by Bayard and Ploen exploits the fact that MEMS sensors are small and can be manufactured in large numbers on a single chip. Separate MEMS sensors manufactured on a single chip will be independent of each other thus providing individual measurements. However due to the size made possible by the MEMS technology, it is possible to manufacture several sensors within a very narrow area. The sensors will thus be both identical in relation to design, working conditions and location. The idea presented by Bayard and Ploen postulates that under these special manufacturing conditions favorable correlation between the gyroscope may exist. These favorable correlation can, if identified can be used to improve performance.

1.3 Concept Evaluation

This section presents a preliminary evaluation of the feasibility of the concept presented by Bayard and Ploen. The evaluation shows how knowledge of the correlations in the noise can be used to improve the performance of the final signal.

A situation is considered where noisy sensor signals are combined using a averaging strategy. A simple model is used for three identical sensors. Because the sensors are identical, averaging the measurements is a good method for approximating the true value. However, if the noises affecting the signals are correlated, it can be shown that averaging is not the optimum technique for combining the signals and that the best solution is a weighted average.

A single sample measurement with the following model is considered:

$$\begin{cases} z_1 &= \omega + n_1 \\ z_2 &= \omega + n_2 \\ z_3 &= \omega + n_3 \end{cases} \tag{1.1}$$

All variables are scalars, where z_1, z_2 and z_3 are measured output signals, ω is the true signal, n_1, n_2 and n_3 are noise processes with zero mean and known, equal variance $\mathrm{Var}(n_1) = \mathrm{Var}(n_2) = \mathrm{Var}(n_3) = \sigma$. The noises might not necessarily be independent. The noises are expressed as a linear mix of n independent and identical random sources s_i with different coefficients c_{ij}.

$$\begin{cases} n_1 &= c_{11}s_1 + c_{12}s_2 + ... + c_{1n}s_n = \sum_{i=1}^{n} c_{1i}s_i \\ n_2 &= c_{21}s_1 + c_{22}s_2 + ... + c_{2n}s_n = \sum_{i=1}^{n} c_{2i}s_i \\ n_3 &= c_{31}s_1 + c_{32}s_2 + ... + c_{3n}s_n = \sum_{i=1}^{n} c_{3i}s_i \end{cases} \tag{1.2}$$

The objective is to combine the measured values linearly so that the resulting signal will have a minimum variance noise error component. The problem is formulated as follows:

$$\begin{aligned} \hat{\omega} &= x_1 z_1 + x_2 z_2 + x_3 z_3 \\ &= (x_1 + x_2 + x_3)\,\omega + x_1 n_1 + x_2 n_2 + x_3 n_3 \end{aligned} \tag{1.3}$$

where the parameters x_1, x_2, and x_3 are unknown. They must be chosen so two condition are met: the scale of the measured valued should be the same and the variance of the noise minimized.

$$\begin{cases} x_1 + x_2 + x_3 &= 1 \\ [x_1, x_2, x_3] &= \mathrm{argmin}_x \mathrm{Var}_{\hat{\omega}}(x_1, x_2, x_3) \end{cases} \tag{1.4}$$

The minimum variance condition is addressed next:

$$\begin{aligned}\mathrm{Var}(\hat{\omega}) &= \mathrm{Var}(x_1 n_1 + x_2 n_2 + x_3 n_3) \\ &= \mathrm{Var}(x_1 \sum_{i=1}^{n} c_{1i} s_i + x_2 \sum_{i=1}^{n} c_{2i} s_i + x_3 \sum_{i=1}^{n} c_{3i} s_i) \\ &= (x_1 c_{1i} + x_2 c_{2i} + x_3 c_{3i})^2 \mathrm{Var}(s_1) + ... + (x_1 c_{ni} + x_2 c_{ni} + x_3 c_{ni})^2 \mathrm{Var}(s_n) \\ &= \Big((x_1 c_{1i} + x_2 c_{2i} + x_3 c_{3i})^2 + ... + (x_1 c_{ni} + x_2 c_{ni} + x_3 c_{ni})^2 \Big) \sigma_s \end{aligned} \tag{1.5}$$

From the above it can be noted that the variance expression is a sum of quadratic terms, and it is always positive. The minimizing solution (x_1, x_2, x_3) would then be $(0, 0, 0)$, and this is not a valid case for the estimation. Therefore the scaling constraint is used to eliminate the x_1 parameter in the following way:

$$x_1 = 1 - x_2 - x_3 \tag{1.6}$$

The expression for $\hat{\omega}$ is now as follows:

$$\begin{aligned}\mathrm{Var}(\hat{\omega}) = \Big(&[(1 - x_2 - x_3) c_{1i} + x_2 c_{2i} + x_3 c_{3i}]^2 + \dots \\ &\dots + [(1 - x_2 - x_3) c_{ni} + x_2 c_{ni} + x_3 c_{ni}]^2 \Big) \sigma_s^2 \end{aligned} \tag{1.7}$$

By Fermat's theorem of local minima and maxima for stationary points, the condition for minimizing the variance expression is that the partial derivatives in parameters x_2, and x_3 are set to 0.

$$\begin{aligned}\frac{\partial \mathrm{Var}(\hat{\omega})}{\partial x_2} &= 0 \\ \frac{\partial \mathrm{Var}(\hat{\omega})}{\partial x_3} &= 0 \end{aligned} \tag{1.8}$$

To make the calculations easier, only 3 noise sources are considered next, s_1, s_2 and s_3. The system of partial derivatives described above reduces to the following ordinary linear system:

$$(c_{11} - c_{21}) c_{11} + (c_{12} - c_{22}) c_{12} + (c_{13} - c_{23}) c_{13} = \Big[(c_{11} - c_{21})^2 + (c_{12} - c_{22})^2 + (c_{13} - c_{23})^2 \Big] x_2 + \dots$$

$$\dots + \Big[(c_{11} - c_{21})(c_{11} - c_{31}) + (c_{12} - c_{22})(c_{12} - c_{32}) + (c_{13} - c_{23})(c_{13} - c_{33}) \Big] x_3 \tag{1.9}$$

$$(c_{11} - c_{31}) c_{11} + (c_{12} - c_{32}) c_{12} + (c_{13} - c_{33}) c_{13} = \Big[(c_{11} - c_{31})^2 + (c_{12} - c_{32})^2 + (c_{13} - c_{33})^2 \Big] x_3 + \dots$$

$$\dots + \Big[(c_{11} - c_{31})(c_{11} - c_{21}) + (c_{12} - c_{32})(c_{12} - c_{22}) + (c_{13} - c_{33})(c_{13} - c_{23}) \Big] x_2 \tag{1.10}$$

which has the matrix form:

$$\begin{bmatrix} \times & \times \\ \times & \times \end{bmatrix} \begin{bmatrix} x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \times \\ \times \end{bmatrix} \tag{1.11}$$

$$A\mathbf{x} = \mathbf{b}; \;\; \mathbf{x} = A^{-1}\mathbf{b} \tag{1.12}$$

A numerical example is considered next. The standard deviation of the independent noise sources is 1. The mixing coefficients c_{ij} are taken such that the resulting noise components n_1, n_2 and n_3 have equal standard deviation, also 1.

$$\sigma_s = 1, \qquad \sigma_n = 1, \qquad C = \begin{bmatrix} +0.8 & +0.2 & +0.5657 \\ +0.8 & +0.4 & +0.4472 \\ +0.2 & +0.7 & -0.6856 \end{bmatrix} \tag{1.13}$$

In this case, the coefficients of the linear weighted estimator can be identified to be:

$$X = \begin{bmatrix} 3.8458 & -3.7767 & 0.9310 \end{bmatrix}^T \tag{1.14}$$

The standard deviation of the estimate computed with the optimized coefficients is 0.3 and smaller than the variance of the averaged estimator which is $\frac{1}{\sqrt{3}} = 0.577$. The conclusion is that the weighted average provides a better estimate compared to simple signal averaging.

To better illustrate that the weighted average provides a better estimate than averaging the following illustrations have been generated. Two cases are considered, one where the actual property ω has a constant value and one where it is a sinusoidal. The performance of the estimate is illustrated figure 1.3 and 1.4. The actual property ω is draw in black and the sensor measurements are represented by colored points. The averaged estimation is depicted with red, and the optimized estimation with blue. The measurement is simulated with multiple samples, each sample is optimized individually using only the linear combination with the computed weights. From figure 1.3 and 1.4 it can be seen that even though the sensors are

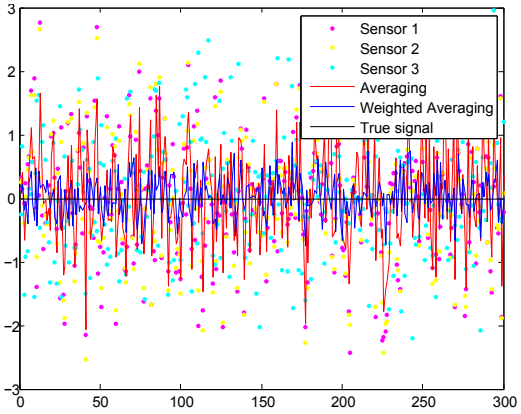

Figure 1.3: Estimation of constant value

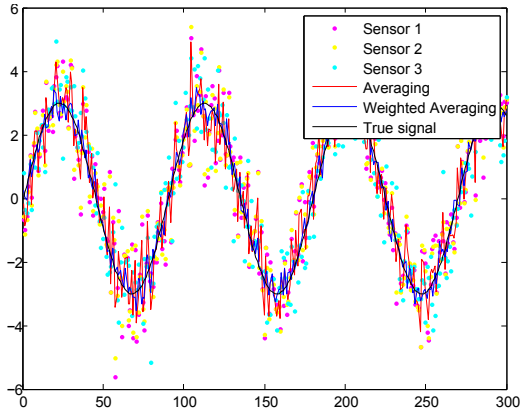

Figure 1.4: Estimation of sinusoidal signal

identical in noise levels and no measurement is "better" than the other, if the noises are correlated then the weighted average is better than simple averaging.

Chapter Conclusion

As presented in this section, high grade INS have historically been dominated by large and expensive solutions, that employ high grade gyroscope technologies. A critical problem of all gyroscopes, but especially in relation to MEMS, is the concern of integration drift, where small errors in the measurement accumulate over time. For most consumer applications with limited free inertial operation time, the problem of integration drift is less crucial as the system can be reinitialized, using external aiding, before the accumulated error grows out of bound. For systems with prolonged free inertial operation time gyroscopes technologies that provide extremely low integration drift are used. The main issues of these systems are their size and cost, which both limit their usage to large host vessels and render the systems expensive.

MEMS is a rapidly growing technology which has enabled increasingly smaller and cheaper gyroscopes. The current use of MEMS in relation to prolonged free inertial operation time systems is however limited due to low accuracy. One method of improving the accuracy of MEMS gyroscopes may be to use multiple gyroscopes and fuse the measurements using Kalman filtering. The method assume that homogeneous MEMS gyroscopes, manufactured on the same chip, will have favorable correlations in relation to sensor noise. The method presented by [4] has been shown to provide promising results under simulation. A crucial

observation in relation to the method is that no investigation, that this group is aware of, have shown that such favorable correlation indeed exist between same chip MEMS gyroscopes.

In the project, the the perspective of using such noise correlations in optimized estimation strategies are investigated. In addition, the existence and identification of noise correlations are also addressed. In the following chapter the project scope and objectives are presented in detail

Chapter 2

Project Scope and Objectives

The results presented in the former chapter indicate that the performance of MEMS can be improved by fusing measurements from multiple homogeneous gyroscopes. In the following the project scope and objective is presented along with the project plan of action.

The project main objective is to answer the following question:

"Can the accuracy of MEMS gyroscopes be improved by combining multiple homogeneous units into a single sensor?"

The project approach is to analyze the characteristics of gyroscopes of same manufacturing, develop an adequate linear model to account for the main error components, and design a Kalman based sensor fusion method to optimally combine the multiple signals into a single improved output. The block elements of 2.1 represent both specific problems to be solved,

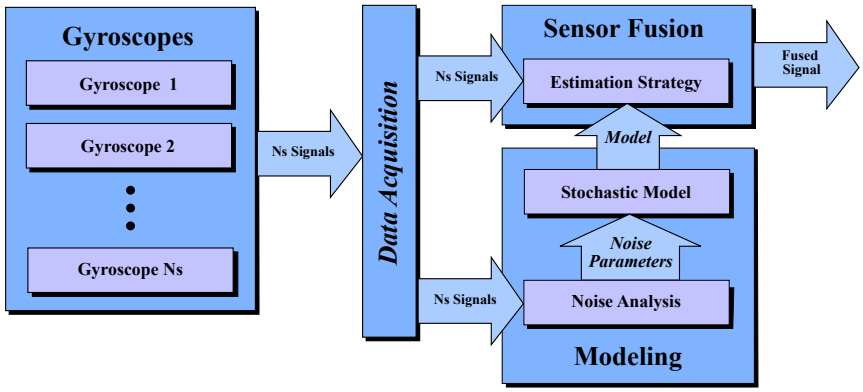

Figure 2.1: Project approach

areas where special knowledge must be acquired and questions that must be resolved. In the following the content and reasoning for the specific elements are elaborated.

Modeling

The filtering strategies used in this project are to be based on the Kalman method and thus require that the system is modeled. As the main focus of the project are the exploitation of favorable noise correlations, the noise processes related to the gyroscopes must be modeled.

Sensor Fusion

The main project element is to evaluate different filtering strategies in relation to homogeneous gyroscope sensor fusion and estimation. The overall strategy is to apply different implementations of the Kalman filter, focusing on utilizing favorable noise correlations between the gyroscopes. The main objective of this element is to provide a transparent evaluation of the performance improvement made possible due to favorable noise correlations. A simple averaging strategy is considered a suitable benchmark for this evaluation.

Correlation Identification

The sensor fusion strategy used in this project require the existence of noise correlations between the used MEMS gyroscopes. In order to evaluate the feasibility of implementing the developed sensor fusion on actual gyroscopes the existence of such noise correlations must be investigated.

In this chapter the project scope and objectives have been presented. The project approach and the main elements of the project have been presented along with considerations related to the project objective. In the following chapter the methods used in the process of solving the project main objective are addressed in detail.

Chapter 3

Gyro-board Equipment

The main focus of the project is to investigate performance improvement in MEMS gyroscopes by using multiple homogeneous units and combining their output signals using sensor fusion methods. In relation to this objective, an array containing several MEMS gyroscope units build by CDL is used in the project. This chapter contains a thorough description of the gyroscope board and presents considerations related to sensor settings and interface.

3.1 Gyro-board Hardware

As presented in the introduction, Bayard and Ploen [4] postulates that favorable correlations between gyroscopes are more likely to exist if several MEMS gyroscopes manufactured on the same chip. The development of such a chip is however considered beyond the scope of this project. An alternative approach is used instead where multiple commercial MEMS gyroscopes are mounted on a custom circuit board within a confined area. The gyro-board has been developed by CDL Scotland, and it is a custom design. The circuit board serves both as a physical framework for the sensors as well as management of the sensory output.

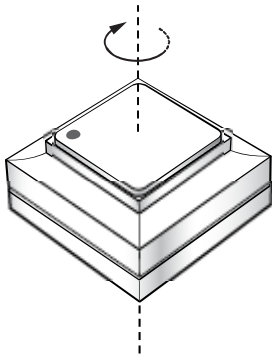

Figure 3.1: ADIS16265 [3]

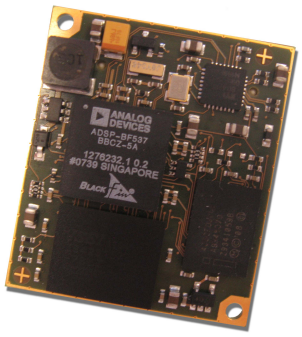

Figure 3.2: CM-BF537E Blackfin [7]

The gyro-board features a total of eight medium-grade programmable yaw rate MEMS gyroscopes. The used gyroscopes are ADIS16265 [3] as the one illustrated in 3.1. The board also contains a CM-BF537E Blackfin processor module [7], figure 3.2 that manages communication and synchronization of the eight gyroscopes.

The Blackfin board streams data in a packet structure containing a time stamp and the eight sensor values through a serial interface that can be easily accessed by a desktop application. The software components are addressed in more detail in section 3.5.

3.2 Gyroscope Operational Principle

Like most other MEMS gyroscopes, the gyroscopes used in this project are vibratory gyroscopes. Vibratory gyroscopes come in a variety of types, but common to all of them is that they use high-frequency mechanical resonators to induce and detect the Coriolis effect. The use of resonators as opposed to rotating components is common as resonators do not require bearings and also eliminate friction and wear. In the following the operational principle of vibratory gyroscopes is described.

The Coriolis effect used in vibratory gyroscope arise from the Coriolis force. The Coriolis force is an fictitious force, that is used to describe the motion of objects observed in a rotating non-inertial reference frame. If a object is traveling in a straight line in an inertial reference frame, and the motion of the object is observed from a rotating reference frame, the object will seam to travel in a curved path [2].

A classical implementation of a vibratory gyroscopes consists of a single vibrating proof mass. The proof mass is suspended above the sensor subtract using flexible flanges that act as suspension to isolate the mass from the gyroscope support structure and allow it to vibrate freely. The proof mass is driven into resonance back and forth along the drive axis using drive electrodes. When the sensor housing is rotated, a Coriolis force perpendicular to both the drive axis and the angular rotation axis is induced on the the proof mass. The displacement of the proof mass is detected using sense electrodes. The direction of the Coriolis force is called the sensitive axis. The concept is illustrated in figure 3.3 where $\vec{\Omega}$ is the angular rate vector, $\vec{v}$ is the vibratory velocity vector and $\vec{c}$ is the Coriolis vector.

$$\vec{c} \sim \vec{\Omega} \times \vec{v} \tag{3.1}$$

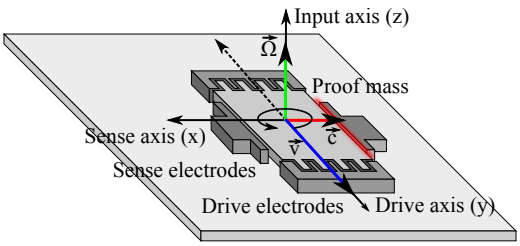

Figure 3.3: Vibratory gyroscope

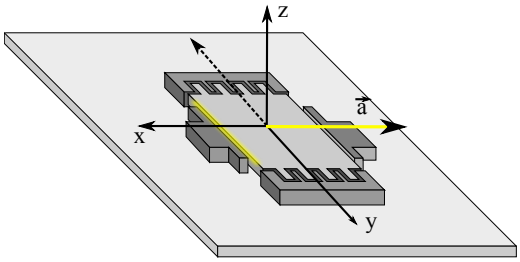

Figure 3.4: Disturbance along the sensitive axis

Figure 3.4 shows one problem with this gyroscope design. Any acceleration $\vec{a}$ along the sensitive axis is mistaken for angular rotation. In the ADIS16265 [3] this problem is countered by using a two mass setup as illustrated in figure 3.5. The main aspect is that the two mases are vibrating in anti-phase and their velocity vectors will therefore point in opposite direction. From figure 3.5 it can be seen that angular input will generate Coriolis vectors with different directions that effect the sense electrodes differently. This setup minimizes the response to linear acceleration (gravity, vibrations) along the sensitive axis. This concept is presented in greater detail in [8].

The signal detected by the sense electrodes is then amplified and processed to produce the analog electrical rate output.

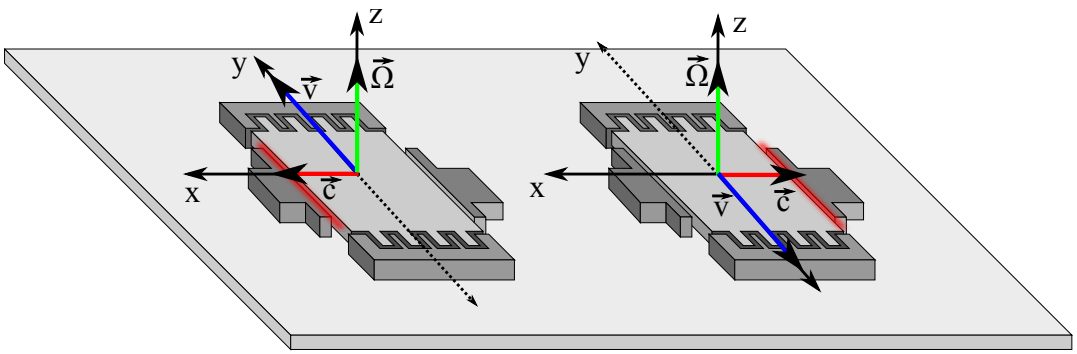

Figure 3.5: Conceptual illustration common-mode rejection in the ADIS16265 [3]

3.3 Design Considerations

The eight MEMS units are arranged on the gyro-board and effectively form a gyroscope cluster with a minimum of separation between the sensors. Each gyroscope in the cluster operates independently and measures yaw rate around the z-axis as illustrated in figures 3.7 and 3.6. The underlying idea behind the gyro-board design has been to mount the gyroscopes as close as possible while using identical control and sampling circuitry.

The sensor orientation is also considered, and the mounting configuration is described in figure 3.7. The input axis of all the units must be parallel so as to measure the same rotational rate. The sensors are organized in two layers of four sensors with flipped orientations of the input axis. Within each layer, the sensors are rotated such that both the derived axis and the sensitive axis have opposite directions in pair of twos. This can be seen in the figure 3.7 by the dot marker on the sensors.

It is hoped that gyroscope array designed with all these considerations will improve the possibility of the gyroscopes being favorably correlated. However, it is important to notice that the existence of such correlations has not been proven, the presence of favorable correlation between the gyroscopes is still thought to be mainly coincidental.

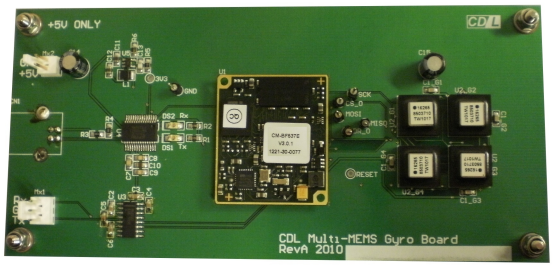

Figure 3.6: Gyro-board - Top view

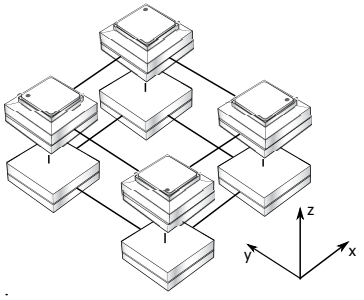

Figure 3.7: Gyroscope configuration

3.4 Sensor Settings

The ADIS16265 gyroscope [3] has several settings accessible to the system designer in order to tailor the performance of the sensor to applications specific requirements. These include the possibility to change the dynamic range, sampling rate, a bandwidth switch and the window size parameter of a digital FIR filter. In the following, considerations related to these parameters are discussed and the chosen setting configuration presented.

The dynamic range of the gyroscope specifies the reliably measurable angular velocity input in

term of magnitude. That is the maximum positive and negative input that can be measured without distortion. The gyroscope can operate in three different dynamic range settings $\pm 80°/s$, $\pm 160°/s$ and $\pm 320°/s$. The sensor features a 14 bit register which represents the measured angular velocity. Depending on the chosen dynamic range each bit in the register represents a different increment in angular speed. In this way higher resolution is possible if it is know that the dynamics of the system will be within one of the given ranges. In relation to this project the dynamic range is chosen to $\pm 320°/s$. The reasons for this is that the two other settings force usage of the FIR filter. This is considered to be a critical issue as the basis of the project is the existence of noise correlations and it is assumed that the investigation of these noise correlations will be most representable if the unfiltered signal is used.

The ADIS sensors default and maximum sampling frequency is 2048Hz. The preliminary evaluation of the raw signal from the gyro-board at 2048Hz revealed that coloration and time correlations. The framework under which the project was designed required the measurement noise to be uncorrelated, in order to ease the implementation of a Kalman filtering scheme. The problem can be bypassed by designing and including a shaping filter in the filter, but another solution was chosen. Since the sampling frequency is not considered important to the main theoretical elements in the project and no externally driven requirements are present, it was chosen to undersample the gyro-board gyroscopes signal by saving only every 20^{th} sample. This signal has a convenient white characteristic. The resulting sampling frequency can be calculated as follows:

$$F_s = \frac{2048}{20} = 102.4 \tag{3.2}$$

Thus the signal frequency of the board data is 102.4Hz and used as such in the methods working directly with the board signal such as the Allan variance. In the simulation algorithms that test the estimation methods the 100Hz value was used as an approximation.

The gyroscope bandwidth parameter switches between 50Hz and 330Hz. This parameter is related to a analog low-pass filter with a programmable primary pole. The largest bandwidth setting is used, and one of the reasons is to maintain the white characteristic of the undersampled 102.4Hz signal. For the same reason, the digital FIR filter is deactivated by setting its window size parameter to 1.

3.5 Gyro-board Interface Software

In this section the gyro-board software components are described.

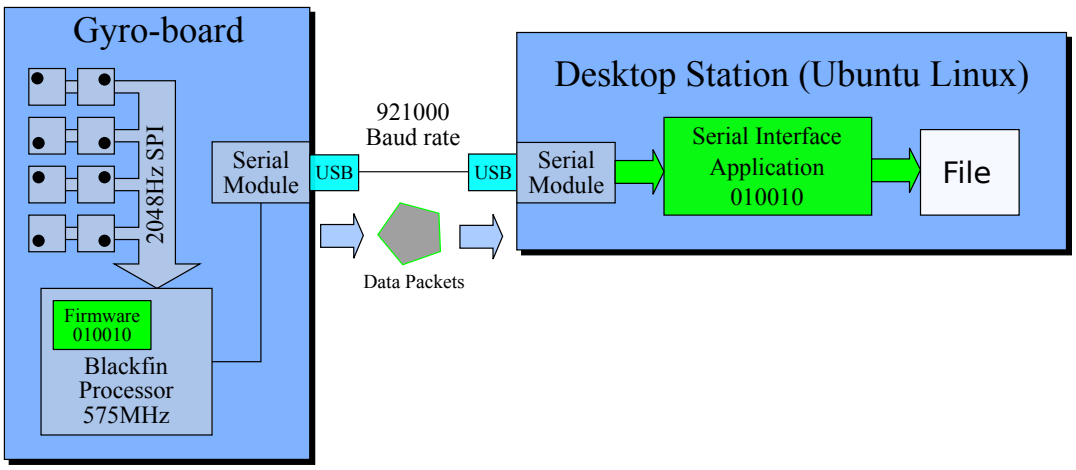

Figure 3.8: Conceptual overview of the hardware and software components

The gyroscope board streams data packets on a serial line at a baud rate of 921000. A desktop computer running Linux Ubuntu was used to interface with the board. The packets contain a time-stamp taken by the Blackfin processor, the eight angular rate data values from the sensors, and a hash signature. A serial interface C++ application has been developed to acquire and process the data packets and write them to a file on disk. The application is cross-platform, Linux and Windows, and the code is included on the project CD.

The gyroscope board is powered by a Blackfin microcontroller from Analog Devices, running at a fast 575MHz clock. A firmware code is uploaded on the board and it performs communication over the SPI bus to send initialization commands to the sensors and receive data values, synchronizes the data from the eight units and build the data packet, setup and communicates with the serial module to stream the data. The firmware code is written in C and was developed by CDL, however small changed were made to change the sensor settings at initialization.

As such, a Blackfin development environment was setup by the group with help from the CDL developers. The Blackfin development toolchain is available as open source, and the ADIS datasheet contains description about the setup procedure of the sensor settings registers. The code is modified and recompiled in the toolchain environment and is uploaded on the Blackfin by accessing the boot sequence of the the microcontroller through the same serial line connection by sending boot command in the first seconds of the start-up.

Chapter Conclusion

In this section the project main piece of equipment has been presented. The gyro-board is a custom piece of hardware which feature eight ADIS16265 [3] programmable gyroscopes. The gyro-board is a custom piece of hardware and has been devolved by CDL, Scotland specifically for this project. The ADIS16265 feature multiple different settings that include varying levels of filtering. In this project however, where the sensor noise is part of the main focus, the gyroscope is used with a minimum of filtering. The used sampling frequency for the gyro-board is set to 102.4Hz to ensure that the gyroscopes measurement noise is approximately white. The physical design of the gyro-board along with the used sampling software has also been presented.

In the following section the methods and accompanying theory used in relation to the project is presented.

This page intentionally left blank

Chapter 4

Methods

In this chapter an overview of the methods used in the thesis is given. Often in literature the methods chapter contain in depth description of all the methods and theory. We have however striven to condense the material to it's essential elements. In this way we hope to equip the reader with a clear understanding of the why and how the methods are applied in the project. The objective of the chapter is not to provide extensive theory and derivations of the methods but to give the reader a clear overview of the methods and their validity in relation to the project.

The following topics will be addressed as logical steps related to the thesis plan of action.

- **Noise Analysis**
  Used in the process of characterizing the project gyroscopes
- **Kalman Filter**
  Used in estimation filter and the process of estimating sensor noise correlations
- **State Smoother**
  Used in the process of estimating sensor noise correlations
- **Noise Correlation Identification**
  Used in the process of estimating sensor noise correlations

As illustrated above is is clear that project can be divided into several components of varying complexity. In the following thee components are addressed to provide the reader with a clear understanding of the project plan of action and relevant methods.

4.1 Random Processes

Datasheets of MEMS gyroscopes contain parameters such as Angular Random Walk or Rate Random Walk coefficients to describe and quantify different noise components present in the output signal. In this section introduces preliminary information needed in clarifying the meaning and usage of these parameters.

4.1.1 Random Walk process

We consider a signal $B(t), \forall t \geq 0$ that is measured every T_s seconds, but in such a way that only the change from one measurement to another, ΔB is recorded. The following recursive

expression b_k can be used to partially reconstruct the process $B(t)$,

$$b_k = b_{k-1} + \Delta B_k \tag{4.1}$$

and to recover directly the value at sample N,

$$b_N = \sum_{k}^{N} \Delta B_k \tag{4.2}$$

If such a measurement is performed for a long period of time, the recorded values ΔB will have the properties of a white random process. The average mean of the ΔB_k sequence is close to 0, containing both positive and negative values, and the measurements ΔB_k are not correlated with each other. Furthermore, the absolute value of the ΔB measurements will depend of the sampling period, such that if the sampling period is small then ΔB tend to be somewhat bigger in average, than for measurements taken with a larger sampling period.

The above original signal $B(t)$ has the characteristics of a Weiner process, also known as a Brownian motion, which is a continuous time stochastic process that is defined by three main properties. The value at initial time is zero, a process realization can be said to be a continuous function in t, and its increments between two time values t and $t + \Delta t$ are independent and normally distributed:

$$\begin{cases} B(t=0) = 0 \\ B(t) \text{ is continuous in } t \\ B(t+\Delta t) - B(t) \sim \mathcal{N}\big(0, (RW_c)^2 \Delta t\big) \end{cases} \tag{4.3}$$

where RW_c is a constant coefficient that determines the scale of the process.

In an applied mathematical context such as signal processing and control, the Weiner process can be used as the integral of white noise.

$$\dot{B}(t) = W(t) \Leftrightarrow B(t) = \int W(t) \tag{4.4}$$

When a Weiner process is sampled, the discrete random walk process is a random walk. The discrete process $b(k)$ obtained by sampling $B(t)$ with period T_s is described by either of the two equivalent forms:

$$b(k) = b(k-1) + T_s w(k) \Leftrightarrow b(N) = \sum_{k=0}^{N} w(k) T_s \tag{4.5}$$

The sampled random walk process is the main focus of this section, because the majority of the MEMS gyroscope units on the market, including the one used in this project, have a digital output.

In the definition above, $\{w(1), w(2), \dots w(N)\}$ is the increment white noise sequence that generates the random walk. Each $w(k)$ element is Gaussian random variable with mean 0 and variance $\sigma_w^2(k) = \sigma_w^2$ that is constant across the time index k. These random variables are independent of each other, and as such, the correlation matrix of the entire sequence will be diagonal.

In the random walk sequence $\{b(1), b(2), \dots b(N)\}$, each variable $b(k)$ has also mean 0 and this can easily be seen by applying the expectation operator $\mathrm{E}\langle\cdot\rangle$ operator to equations (4.4) or (4.5). The variance of $b(k)$ is growing with the time-index k, and can be calculated explicitly from the measurement model using the property of variance being distributive over addition for independent random variables.

$$\mathrm{Var}\big(b(N)\big) = \mathrm{Var}\left(\sum_{k=0}^{N} T_s w(k)\right) = T_s^2 \sum_{k=0}^{N} \mathrm{Var}\big(w(k)\big) = N T_s^2 \sigma_w^2 \tag{4.6}$$

$$\sigma_{b_N} = \sqrt{N} T_s \sigma_w \tag{4.7}$$

The figure 4.1 below shows a white noise sequence with $\sigma_w = 1$ and 100 elements, and the corresponding random walk obtained by integration. The sampling period is considered 1 second. The second figure 4.2 shows a cone of 50 random walks generated by different realizations of the same white noise process. It can be seen how the variance of the process is increasing in time.

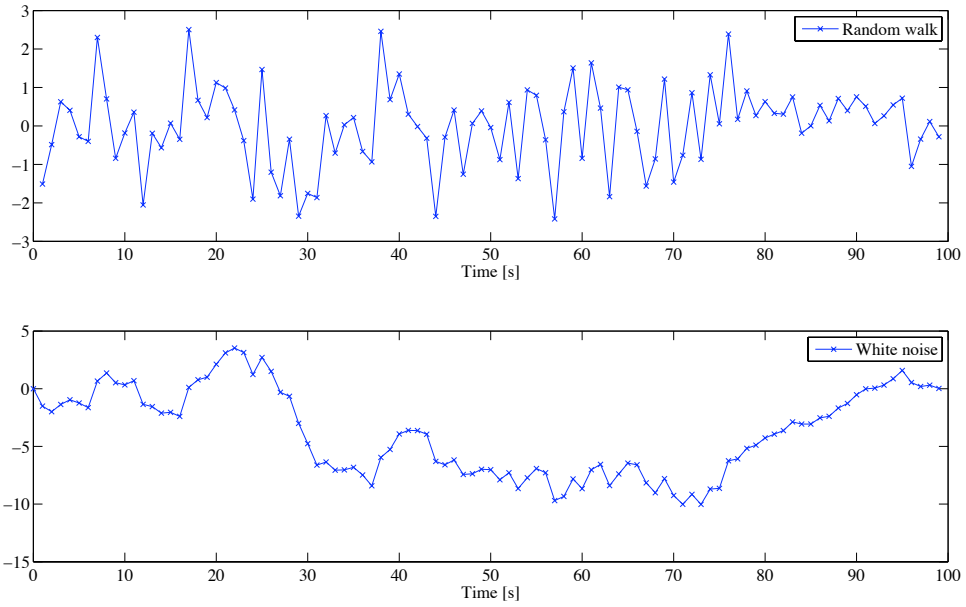

Figure 4.1: Random walk

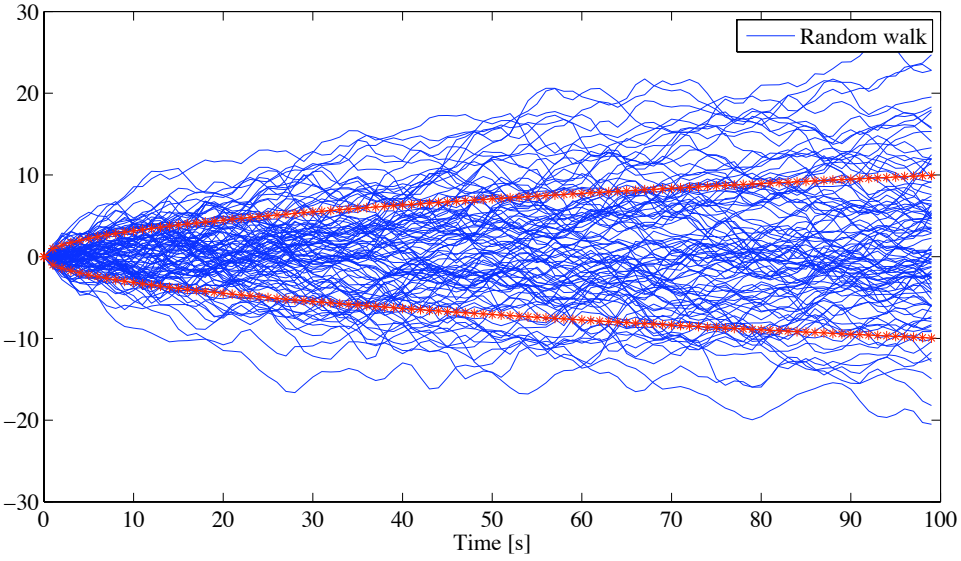

Figure 4.2: Random walk cone

Properties of the sampled random walk

A Weiner signal $B(t)$ can be sampled and measured with different frequencies. The characteristics of the associated generator white noise sequences at different frequencies are discussed next.

We consider a continuous random walk process $B(t)$ and a discrete sequences $b(k)$ that has been sampled with period T_s from $B(t)$. Both signals are equivalent in the sense that the average walk, or the standard deviation of the processes is the same after time $t = NT_s$.

$$\mathrm{Var}\big(B(t)\big) = \mathrm{Var}\big(b(N)\big) \tag{4.8}$$

By expanding the variance of the discrete random walk with the equivalent in relation in (4.6), it can be written that:

$$\mathrm{Var}\big(B(t = NT_s)\big) = NT_s^2\sigma_w^2 = tT_s\sigma_w^2 \tag{4.9}$$

From the definition of the Weiner process, $\mathrm{Var}\big(B(t)\big) = \mathrm{RW}_c^2 t$, resulting that:

$$\sigma_w = \frac{\mathrm{RW}_c}{\sqrt{T_s}} \tag{4.10}$$

This relation is important because it related the property of the increment noise to the coefficient of the random walk, and make it is easy to convert one of these parameters into another.

A property of a sampled random walk is that the variance its increment noise increases as the sampling frequency is higher, since:

$$\sigma_{w_1}\sqrt{T_{s1}} = \sigma_{w_2}\sqrt{T_{s2}} \tag{4.11}$$

It also noted here that the unit of the constant RW_c is $[u/\sqrt{s}]$, where u is the corresponding unit of the signals $B(t)$ or $b(k)$. For example, a random walk process present in an angle signal will be characterized by a random walk coefficient RW_c given in units $[^\circ/\sqrt{s}]$ or $[\mathrm{rad}/\sqrt{s}]$, and a random walk process present in an angular rate signal will be characterized by a random walk coefficient given in units $[^\circ/s\sqrt{s}]$ or $[\mathrm{rad}/s\sqrt{s}]$.

4.1.2 Integrated Random Walk

A look is taken next at a process obtained by integrating the sampled random walk process $b(k)$.

$$r(N) = \sum_{k=1}^{N} b(k)T_s \tag{4.12}$$

The r process is an integrated random walk, or a double integrated white noise process. We show next that this process increases much faster than the random walk. A noise that passes through an integrator filter will become more relevant in the final signal, and a noise that passes though a double integrator even more so.

The variance of the integrated random walk process at time step N can be calculated by taking special care of the fact that processes $b(k_1)$ and $b(k_2)$ are correlated, and not independent,

and using properties of the covariance operator.

$$
\begin{aligned}
\mathrm{Var}\big(r(N)\big) &= \mathrm{Var}\left(\sum_{k=1}^{N} b(k)T_s\right) = \\
&= T_s^2 \sum_{k=1}^{N} \mathrm{Var}\big(b(k)\big) + \sum_{k_1=1}^{N}\sum_{k_2=1}^{N} |_{k_1\neq k_2} \mathrm{Cov}(b(k_1), b(k_2)) = \\
&= T_s^2 \sum_{k=1}^{N} kT_2^2\sigma_w^2 + \sum_{k_1=1}^{N}\sum_{k_2=1}^{N} |_{k_1\neq k_2} \mathrm{Cov}\left(\sum_{i=1}^{k_1} w(i)T_s, \sum_{j=1}^{k_2} w(j)T_s\right) = \\
&= \frac{N(N+1)}{2} T_s^4\sigma_w^2 + \sum_{k_1=1}^{N}\sum_{k_2=1}^{N} |_{k_1\neq k_2} \sum_{i=1}^{k_1}\sum_{j=1}^{k_2} T_s^2 \mathrm{Cov}\big(w(i), w(j)\big) = \\
&= \frac{N(N+1)}{2} T_s^4\sigma_w^2 + T_s^2 \sum_{k_1=1}^{N}\sum_{k_2=1}^{N} |_{k_1\neq k_2} \sum_{k=1}^{\min(k_1,k_2)} \sigma_w^2 = \\
&= \frac{N(N+1)}{2} T_s^4\sigma_w^2 + T_s^2\sigma_w^2 \sum_{k_1=1}^{N}\sum_{k_2=1}^{N} |_{k_1\neq k_2} \min(k_1, k_2) = \\
&= \frac{N(N+1)}{2} T_s^4\sigma_w^2 + 2T_s^2\sigma_w^2 \sum_{k=1}^{N-1} k(N-k)
\end{aligned}
\tag{4.13}
$$

Equation can be used to analyze the variance of the integrated random walk process over time.

The following figure shows the amplification of the white noise in through the double integrator. It can be seen that the size of the noise increases approximately 100 time compared to the random walk, the single integrator.

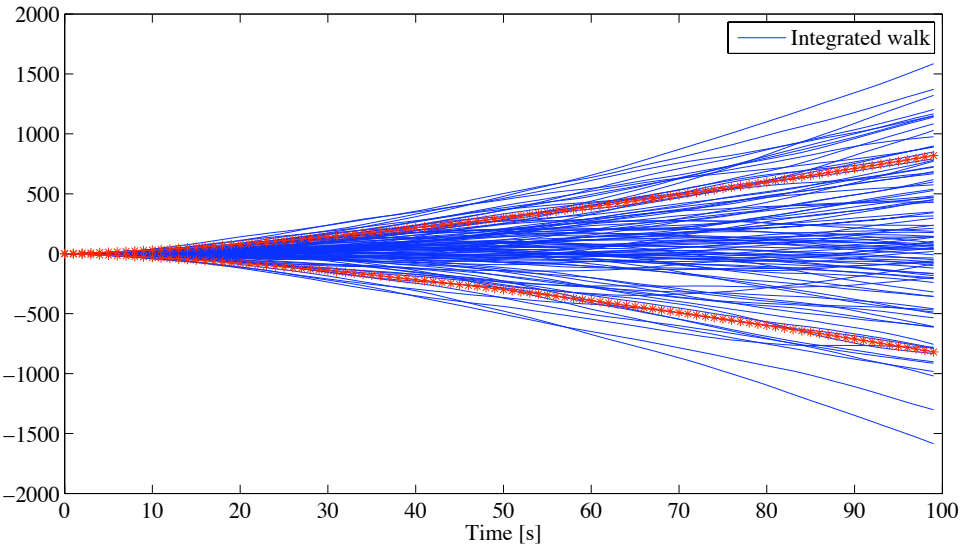

Figure 4.3: Integrated Random walk cone

»»»> .r296

4.2 Allan Variance

The Allan Variance (AV) method, named after David W. Allan is a time domain analysis method developed in the mid-1960ies. The method was originally developed for analysis of high precision clocks and oscillators but it can be applied to various types of precision sensors and measurement devices. The AV analysis is an IEEE accepted standard for characterizing MEMS gyroscopes. In the following overview of the method and related considerations are presented. Unless other is specified the following overview is based on [13].

The AV method, as applied to a sequence of data points, can be divided into tree steps as illustrated below

1. Perform clusters operations on the data and acquire the Root Allan Variance
2. Plot the Root Allan Variance
3. Based on the Root Allan Variance plot identify the type and magnitude of the noise processes in the data

In the following the above steps are elaborated on using examples and figures.

4.2.1 Cluster Analysis

In this section the actual computations related to the AV analysis are described.

Consider the output z from a single sensor sampled in discrete time given as $t = k\tau_0$, $k = 1, 2, 3, \ldots N$. The finite dataset from the sensor thus consists of N consecutive data points with a sample period of τ_0. The cluster operation is illustrated in figure 4.4 where it is applied to simulated data which resemble the output from a MEMS gyroscope.

Having acquired the data set, the next step is to divided the data into clusters with specific length τ_0, $2\tau_0$, $\ldots$ $M\tau_0$, with $M < N/2$. Where M denotes the number of data points within individual cluster. Associated with each cluster is the correlation time τ which is equal to $M\tau_0$. Associated with each cluster k, containing M data points, is a cluster average $\bar{z}$ which is calculated as follows.

$$\bar{z}_k(\tau_M) = \frac{1}{M}\sum_{i=1}^{M} z_{(k+1)M+i} \tag{4.14}$$

$$\underbrace{z_1, z_2, \ldots z_M}_{k=1}, \underbrace{z_{M+1}, \ldots z_{2\cdot M}}_{k=2}, \ldots \underbrace{z_{N-M}, \ldots z_N}_{k=K}, \tag{4.15}$$

where $k = 1, 2, \ldots, K$ and $K = \frac{N}{M}$. The process of dividing the data into clusters with different correlation time τ and performing the cluster averaging operation can be illustrated as in figure 4.4 where $\tau = 0.1$s, $\tau = 0.2$s and $\tau = 1$s respectively. From the figure it is clear how the variance of the cluster averages changes relative to the used τ.

For each set of clusters with length τ, that can be formed from the data set, the successive averages as defined above make up a set of random variables. The object of interest, the Allan Variance σ_A^2, is the variance over all the clusters of length M. The AV $\sigma_A^2(\tau)$ evaluated for cluster with length M is defied as

$$\sigma_A^2(\tau) = \frac{1}{2}\mathrm{E}\left\langle [\bar{z}_{k+1}(M) - \bar{z}_k(M)]^2 \right\rangle \tag{4.16}$$

In this project the standard estimator, (4.16) is used in which the mean is simply estimated using the sample mean. This approach was chosen as this estimator is known to provides sufficient results with acceptable processing time. Other AV estimators exist for example the overlapping estimator. The overlapping estimator is addressed in appendix D on page 115.

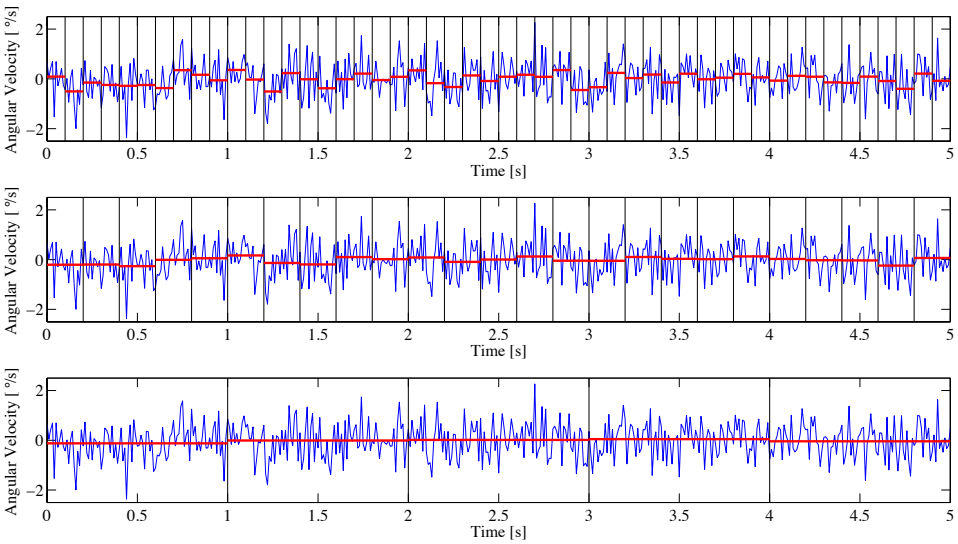

Figure 4.4: Cluster averages for different τ - AV method

The calculation of the standard AV estimator can be easily implemented in MATLAB® as illustrated in the following.

```
function [sig,tau] = allan(data,Fs)
% function [sig,tau] = allan(data,Fs)
% INPUTS:       data:     N x 1 data array                        [°/s]
%               Fs:       Sampling Frequency Fs                   [Hz]
% OUTPUTS:      sig:      Standard Root Allan Variance σA         [°/s]
%               tau:      Cluster time τ                          [s]

tau0 = 1/Fs; N=length(data);            % Sampling period and Length of data set
max_clusternumber = floor( log((N-1)/3)/log(2) );
tau = zeros(1,max_clusternumber+1);
sig = zeros(1,max_clusternumber+1);
for j=0:max_clusternumber
    M = 2^j;                            % Cluster-size being evaluated
    tau(j+1)=M*tau0;                    % Cluster time τ
    index = 0;                          % Index is the d vector index
    d=zeros(1,floor(N/M));              % Cluster difference vector
    for i=1:M:N-M+1
        index = index + 1;
        d(index)=sum(data(i:i+M-1))/M;  % The average for a cluster, size m
    end
    sig(j+1)=sqrt(0.5*mean((diff(d(1:length(d))).^2)));
end
end
```

The function is designed to take a vector of data along with the used sampling frequency as input arguments. The function supplies the values of τ as a vector and the Root Allan Variance vector $\sqrt{\sigma_A^2(\tau)} = \sigma_A(\tau)$ as output. The `allan.m` file can be viewed on the project CD.

Root Allan Variance plot curve

Having presented the method of implementing the AV calculation the next step is to use the Root Allan Variance plot to identify the magnitude and nature of the noise processes contained in the captured data. This is accomplished by plotting the Root Allan Variance, $\sigma_A(\tau)$ as a function of correlation time τ in a loglog plot. The Root Allan Variance plot is thus defined as a plot whose x axis is $x = \log(\tau)$ and whose y axis is $y = \log(\sigma_A(\tau))$. An

often used spacing of theτ values is $M = 2^j$ where j = 0,1,2,3... .

By examining the slope of the Root Allan Variance plot curve, it is possible to identify and determine the magnitude of the different noise processes present in the captured data. In figure 4.5 a standardized Root Allan Variance plot, containing several different noise processes and their appurtenant curve slope is illustrated. The figure is a standardized representation of how processes are represented in the Root Allan Variance plot and should be interpreted as such.

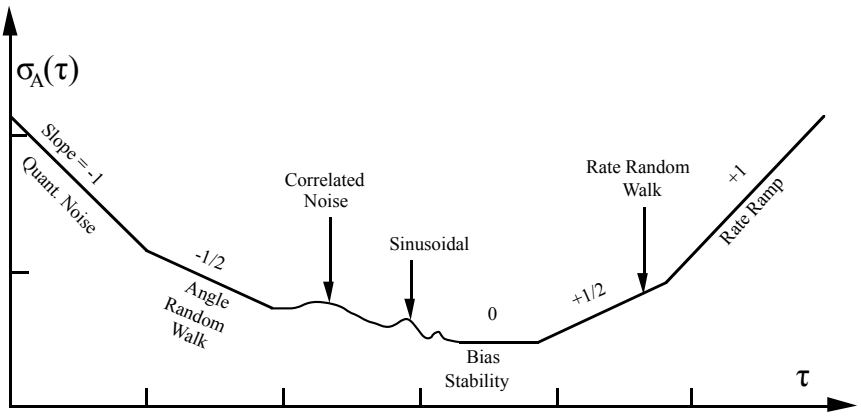

Figure 4.5: Standardized Root Allan Variance plot after [13, p. 71] - AV method

ARW and RRW coefficients

Two types of random processes which are know to be main contributors in MEMS gyroscopes signal noise are Angular Random Walk (ARW) and Rate Random Walk (RRW). Equations for calculating the AV related to these two noise processes are presented in the following without derivation.

$$\sigma^2_{A,ARW}(\tau) = \frac{ARW_c^2}{\tau} \qquad \text{and} \qquad \sigma^2_{A,RRW}(\tau) = \frac{RRW_c^2 \tau}{3} \tag{4.17}$$

where ARW_c is the variance coefficient at $\tau = 1$ which intersects the $-1/2$ slope appurtenant to the ARW noise process. RRW_c is the variance coefficient at $\tau = 3$ which intersects the $+1/2$ slope appurtenant to the RRW noise process. The same approach, that has been presented here, will be used in the process of identifying the noise processes contained in the sensors used in the project.

The ARW and RRW process noises and their meaning is clarified next. The RRW coefficient refers to random walk noise component in the gyroscope rate output signal. This random walk is caused by accumulated errors in the mechanical dynamics of the sensor. This component has a very small coefficient and therefore a dynamic in the low frequency band, and it cannot be removed by classical filtering on board of the sensor. The gyroscope output rate signal thus will drift due to the RRW process, in average by a quantity equal to $\text{RRW}_c \cdot \sqrt{\Delta t}$ where Δt is the considered time duration.

The ARW component refers to a random walk noise in the angle signal. The angle is not measured directly by the sensor, but it is obtained from the rate signal by integration. As such, by saying the ARW noise process we are in fact referring to the white noise present in the rate signal of the gyroscope. Given the ARW_c coefficient of the gyroscope, we know to find the standard deviation of the white noise in the rate signal also by using equation (4.10). It is noted here that white noise at the rate signal, and correspondingly the ARW_c coefficient, will change when the signal is processed by low-pass or FIR filters.

In this section a overview of the AV method has been provided. Using simulated data the computational methods and and the process of identifying noise processes in captured data

has been presented. In appendix D on page 115 the theoretical basis for the method is addressed in greater detail. The appendix also includes explanation of the connection of between the different noise processes and the Root Allan Variance plot as well as considerations related to the accuracy of the method and test design.

4.3 Kalman Filter

The Kalman filter is a discrete algorithm that uses noisy measurements and an approximate knowledge of dynamics to estimate the true internal behavior of the system in-run. The measurements are coming from sensors, and the knowledge about the internal dynamics is described with a linear and discrete state space model. The state-space model contains explicit uncertainty information in the form of random variables. For a linear system, with Gaussian and time-uncorrelated uncertainties and measurement noises, the Kalman filter gives, in a theoretical context, the best possible estimation.

The Kalman algorithm is widely used in most fields of modern technology, embedded in electronics products. The filter, in its most known form, is named after Rudolf E. Kalman, a Hungarian-American electrical engineer, mathematical theorist and college professor who first presented the method in 1960 in the article *A New Approach to Linear Filtering and Prediction Problems* [15]. There is a large body of scientific work that both precedes and follows the publication of this paper. For a short overview of the estimation problem and history we can recommend the *"Introduction"* and *"Scope and Objectives"* sections from the book *"Stochastic Processes and Filtering Theory"* by Jazwinski [14].

The Kalman algorithm is the solution for an estimation problem that can be described as follows. A linear system whose properties of interest are changing over time is observed with discrete-timed sensors. The measurements are corrupted by random noise. Using the available measurements, both the current and the sequence of past ones, it is desired to compute numerical values for a set of system properties that are of interest. These properties are called the system states, and may be the same as the one measured by sensors, or/and hidden properties that are not directly observed or measured. Besides the available measurements, the Kalman filter uses generic knowledge about evolution in time (or dynamics) of the system states, which needs to be given as a mathematical description with both deterministic and stochastic components.

Since the Kalman algorithm combines data from different sensors to make the estimation, it is a natural choice for sensor fusion problems. A system would typically be observed with a set of heterogeneous sensors to measure different properties. For the multi-MEMS problem studied in the project, the Kalman filter is used to combine measurements from a homogeneous sensor configuration.

As a recursive algorithm, the Kalman filter uses the states estimation obtained at the previous step, $k-1$, and the dynamic model, to make a prediction for the state at the time step k. This prediction is combined with the new measurement at time k by a factor that reflects the ratio between the uncertainty in the modeling and the sensor noise. All the operations are linear with respect to the entire set of measurements: matrix and vector additions and scalings with independent matrix factors. The estimate is better than both the prediction and the measurement, and importantly, its uncertainty can be explicitly calculated.

4.3.1 Framework of the method

The knowledge about the evolution in time of the states $\mathbf{x}_k \in \mathbb{R}^n$ needs to be described as a discrete time process and in state-space form. The state space form is a first order linear

matrix difference equation.

$$\mathbf{x}_{k+1} = \Phi \mathbf{x}_k + \Gamma \mathbf{w}_k \tag{4.18}$$

with discrete measurements $\mathbf{z} \in \mathbb{R}^m$ given by a liner algebraic dependence:

$$\mathbf{z}_k = H\mathbf{x}_k + \mathbf{v}_k \tag{4.19}$$

The terms $\mathbf{w}_k \in \mathbb{R}^n$ and $\mathbf{v}_k \in \mathbb{R}^m$ are random vectors. Because of the $\mathbf{w}_k$ and $\mathbf{v}_k$ components, state $\mathbf{x}_k$ and measurement $\mathbf{z}_k$ are also random vectors. Matrices Φ, Γ and H are given and of appropriate sizes to be comfortable in the equations. They represent respectively the dynamics matrix, the noise input matrix and the measurement matrix.

$\mathbf{w}_k$ includes the contributions of modeling uncertainties and of uncontrollable disturbances that affect the evolution in time of the state vector. $\mathbf{v}_k$ is the additive measurement noise that affects the sensor data. Both $\mathbf{w}_k$ and $\mathbf{v}_k$ are vector random variables, with a Gaussian distribution. Their mean is $\mathbf{0}_n$ and $\mathbf{0}_m$ respectively.

$$\begin{cases} p(\mathbf{w}_k) \sim \mathcal{N}(0, Q_k) \\ p(\mathbf{v}_k) \sim \mathcal{N}(0, R_k) \end{cases} \tag{4.20}$$

Q_k and R_k are the covariance matrices corresponding to the $\mathbf{w}_k$ and $\mathbf{v}_k$ random vector variables respectively, and will be described in more detail in section 6.5. In the general case Q_k and R_k can be time-varying, but in the following are considered constant.

The time sequences $[\mathbf{w}_0, \mathbf{w}_1, ...]$ and respectively $[\mathbf{v}_0, \mathbf{v}_1, ...]$ are white, meaning independent in time. Taking noise $\mathbf{w}$ as example, the probabilities in the sequence are not conditioned of each other

$$p(\mathbf{w}_{k_1} | \mathbf{w}_{k_2}) = p(\mathbf{w}_{k_1}), \ \forall k_1 \neq k_2 \tag{4.21}$$

and implicitly, not correlated:

$$\mathrm{Cov}(\mathbf{w}_{k_1}, \mathbf{w}_{k_2}) = 0, \ \forall k_1 \neq k_2 \tag{4.22}$$

When this condition is not met, for example the measurement noise is correlated in time, a shaping filter can be designed [10] and the state space model is augmented to include the states of the shaping filter. The new model will now fulfill the whiteness condition.

Furthermore, $\mathbf{w}_k$ and $\mathbf{v}_k$ are assumed to be independent of each other, $\mathrm{Cov}(\mathbf{w}_k, \mathbf{v}_k) = 0$. Derivation and results of the Kalman filter with correlated process and measurement noise can be found in literature, for example [14], but will not be addressed in this thesis.

Before start, the filter must know the initial state, given as a Gaussian random variable with known mean $\mathbf{x}_0$ and covariance matrix $\mathbf{P}_0$.

If the initial state, the system or measurement noises are not Gaussian, the Kalman filter can still be applied to the linear system and will be the best linear estimator based on the model and measurement, however it is not guaranteed to be the optimum. If the noises are not Gaussian, the best estimation may be a design that is non-linear in the measurements.

The following specific properties of the Gaussian distribution guarantee that a linear combination of the measurements is at any time the best estimate for the true state of the system.

- In general, to fully describe a random variable, the density or distribution function needs to be known. In the case of a Gaussian random variable, either scalar or vector, the probability function can be written explicitly and it depends only on two parameters: the mean and the variance.
- A linear combination of Gaussian random variables is itself Gaussian. Thus, the state of the system, $\mathbf{x}_k$, a random variable, is Gaussian and also the measurement $\mathbf{z}_k$.

- The conditional probability of two Gaussian random variables is also Gaussian, and the conditional expectation is linear in the observed variable,

$$\mathrm{E}\langle X|Y\rangle = f(Y) = a + BY \tag{4.23}$$

  where a is vector of the same size as the random variable X and B is matrix of corresponding size to be multiplied with random vector Y and yield a result in the size of X. Both a and B are deterministic values.

The filter problem is to find a good approximation of the system vector state $\mathbf{x}_k$, given that all the required elements of the state-space model and of the noises are known, and also knowing the sequence of measurements $\mathbf{Z}_p = \{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_p\}$. In the estimation case $p = k$. When $p < k$ the problem is called prediction, and when $p > k$ the problem is called smoothing. The smoother problem appears later in the project and an algorithm is discussed in the next section.

Finally, it is noted that the performance of the Kalman filter depends in a great measure on the validity of the system model. The successful implementation of the Kalman filter requires that the model is a sufficiently accurate representation for the real system.

4.3.2 On the derivation

There are different methods for deriving the recursive equations Kalman filter. In the thesis appendix chapters two derivations are presented in more detail.

The Kalman equations can be given an especially consistent interpretation when they are derived in a probabilistic approach. In the stochastic filtering theory, the solution of the estimation problem is a conditional probability function. This conditional probability function is also the solution for the more general estimation problem of a non-linear system and non-Gaussian random distributions.

Because the state $\mathbf{x_k}$ is a random variable, the probability density function is a complete characterization for it. Knowledge of the measurements is included by introducing conditioning. The filter is completely know if the conditional probability density function,

$$\begin{aligned} p(\mathbf{x_k}|z_1, \ldots, z_k) &= p(\mathbf{x_k}|\mathbf{Z}_k) \\ &= \mathbf{f}(\text{model}, \mathbf{x}_0, \mathbf{Z}_k) \end{aligned} \tag{4.24}$$

can be determined or calculated. This is done starting with the known distribution of the initial state and propagating the conditional density function through the system model.

Next, when the conditional probability density function of $\mathbf{x}_k$ is known, using a goodness criteria, a point-value can be chosen from the distribution as the estimate $\hat{\mathbf{x}}_k$, and its quality as an estimate can be evaluated based of confidence intervals. In the linear Gaussian case, the estimate is always taken to be the mean of the conditional distribution, and the numerical measure for the quality of the estimate is given by the conditional covariance matrix. The covariance matrix of the estimate contains on the diagonal the variances of the states.

Since the random variables in the system are Gaussian, the probabilities are completely determined by the mean and the covariance matrix. However propagating the conditional covariance is complicated and a more detailed discussion is not considered, an interesting material on the subject is [14].

Another approach is to start with a linear recursive formulation and express the estimate at time step k as a sum of two components: a state prediction that is a propagation of the previous estimate $\hat{\mathbf{x}}_{k-1}$ through the model and a weighted contribution from the current

measurement vector.

$$\begin{cases} \hat{\mathbf{x}}_k^- & = \Phi \mathbf{x}_{k-1} \\ \hat{\mathbf{x}}_k^+ & = \hat{\mathbf{x}}_k^- + K_k \mathbf{z}_k \end{cases} \tag{4.25}$$

The matrix K is then determined by searching a minimum for the estimate error in the probability space with the help of the Orthogonality principle\Orthogonal projections theorem. The theorem can be proved by defining a Hilbert space over the random variables with finite variance, where the scalar product between two elements is the expectation operation. As such, the theorem has also an intuitive geometric interpretation.

The recursive equations of the Kalman filter can also be determined using a least-square approach.

4.3.3 Mean and Covariance propagation

Given the initial state vector $\mathbf{x}_0$ known as a random Gaussian variable with mean $\bar{\mathrm{x}}_0$ and covariance P_0, it is of interest to consider the evolution of the state as a random variable through the model, in the absence of measurements. This provides general information about the behavior of the model, showing if an unknown initial uncertainty grows or decreases though propagation. It is also needed in the prediction step of the filter.

The evolution of the state mean, a deterministic value, is governed by a linear algebraic equation in matrix Φ. If the matrix Φ would be strictly stable, with poles inside the unit circle, the mean of the random state would converge to 0 over time.

$$\begin{aligned} \mathbf{x}_{k+1} = \Phi \mathbf{x}_k + \Gamma \mathrm{w}_k \quad \Rightarrow \\ \mathrm{E}\langle \mathbf{x}_{k+1} \rangle = \Phi \mathrm{E}\langle \mathbf{x}_k \rangle + 0 \end{aligned} \tag{4.26}$$

The evolution law for the covariance, $\mathrm{Cov}(\mathbf{x}_{k+1})$, can be obtained by applying definitions and the linearity property of the expected operator.

$$\begin{aligned} \mathbf{x}_{k+1} &= \Phi \mathbf{x}_k + \Gamma w_k \qquad |\mathrm{Cov} \Rightarrow \\ \mathrm{Cov}(\mathbf{x}_{k+1}) &= \mathrm{E}\langle [\Phi \mathbf{x}_k + \Gamma w_k - \Phi \mathrm{E}\langle \mathbf{x}_k \rangle][\Phi \mathbf{x}_k + \Gamma w(k-1) - \Phi \mathrm{E}\langle \mathbf{x}_k \rangle]^T \rangle \\ &= \mathrm{E}\langle [\Phi \mathbf{x}_k + \Gamma w_k - \Phi \mathrm{E}\langle \mathbf{x}_k \rangle][\mathbf{x}_k^T \Phi^T + w_k^T \Gamma^T - \mathrm{E}\langle \mathbf{x}_k \rangle^T \Phi^T] \rangle \\ &= \mathrm{E}\langle [\Phi \bar{\mathbf{x}}_k + \Gamma w_k][\bar{\mathbf{x}}_k^T \Phi^T + w_k^T \Gamma^T] \rangle \\ &= \mathrm{E}\langle \Phi \bar{\mathbf{x}}_k \bar{\mathbf{x}}_k^T \Phi^T + \Gamma w_k \bar{\mathbf{x}}_k^T \Phi^T + \Phi \bar{\mathbf{x}}_k w_k^T \Gamma^T + \Gamma w_k w_k^T \Gamma^T \rangle \\ &= \mathrm{E}\langle \Phi \bar{\mathbf{x}}_k \bar{\mathbf{x}}_k^T \Phi^T \rangle + \mathrm{E}\langle \Gamma w_k \bar{\mathbf{x}}_k^T \Phi^T \rangle + \mathrm{E}\langle \Phi \bar{\mathbf{x}}_k w_k^T \Gamma^T \rangle + \mathrm{E}\langle \Gamma w_k w_k^T \Gamma \rangle \\ &= \Phi \mathrm{Cov}(x_k) \Phi^T + 0 + 0 + \Gamma \mathrm{Cov}(w_k) \Gamma^T \\ &= \Phi \mathrm{Cov}(x_k) \Phi^T + \Gamma Q \Gamma^T \end{aligned} \tag{4.27}$$

Using the notation $P_k = \mathrm{Cov}(\mathbf{x}_k)$, it can be written:

$$P_{k+1} = \Phi P_k \Phi^T + \Gamma Q \Gamma^T \tag{4.28}$$

The evolution of the covariance is determined also by a linear algebraic equation that depends on the matrices Φ and Γ and Q.

4.3.4 Kalman filter equations

Each recursive step k of the Kalman filter has two parts: a prediction and an update. In the predict step, the estimate from the previous iteration $k-1$ is propagated though the model

in (4.30) to obtain $\hat{\mathbf{x}}_k^-$, the *a priori* estimate. Based on this *a priori* estimate, a measurement prediction $\hat{\mathbf{z}}_k$ is made using the observation equation of the model in (4.29). The uncertainty of the estimate at $k-1$ is known by the covariance matrix P_{k-1}^+. By propagating this matrix thorough the state-space model based on relation (4.28) derived in the previous section, the covariance of the *a priori* estimate, P_k^- is calculated in (4.31).

$$\text{Predict step:} \begin{cases} \hat{\mathbf{z}}_k = H\hat{\mathbf{x}}_k^- & (4.29) \\ \hat{\mathbf{x}}_k^- = \Phi\hat{\mathbf{x}}_{k-1}^+ & (4.30) \\ P_k^- = \Phi P_{k-1}^+ \Phi^T + \Gamma Q \Gamma^T & (4.31) \end{cases}$$

In the update step, a new measurement $\mathbf{z}_k$ becomes available and the *a priori* estimate must be combined with this new information. The optimum linear solution is to take the residual between the actual measurement and the predicted one $\hat{\mathbf{z}_k}$, also called the innovation, scale it with a matrix K and add it as a correction to the *a priori* estimate. The result is the *a posteriori* estimate $\hat{\mathbf{x}}_k^+$ in (4.32). The matrix K is called the Kalman gain and is the key of the optimization. It is constructed to actively balance between the the noise levels of the sensor data and the uncertainty of the *a priori* estimate, weighing the uncertainty of the sensors versus the uncertainty of the prediction to make the best combination of the two. It also transforms the measurement residual vector into a compatible state vector contribution.

$$\text{Update step:} \begin{cases} \hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k(\mathbf{z}_k - \hat{\mathbf{z}}_k) & (4.32) \\ K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} & (4.33) \\ P_k^+ = (I - K_k H) P_k^- & (4.34) \end{cases}$$

Finally, the covariance of the *a posteriori* estimator is P_k^+. Equation (4.34) updates the covariance to reflect the modification (decrease) in the uncertainty after the information from the new measurement has been used to update the *a priori* estimate.

The recursive Kalman filter is the optimum solution for the theoretical estimation problem with a linear system and the Gaussian noises. This is guaranteed by the fact that the filter keeps complete information of the random state of the system: the mean and the variance, and it can explicitly propagate the Gaussian probability density through the system and compute the measurement conditioning. The estimate is chosen as the peak/mean of the Gaussian probability density, which is the optimum point for both minimum variance and maximum likelihood or any other reasonable criteria of interest.

Joseph form for the covariance update

The general equation for the covariance update is the Joseph form:

$$P_k^+ = (I - K_k H) P_k^- (I - K_k H)^T + K_k R K_k^T \tag{4.35}$$

Equation (4.34) is a particular form, obtained by considering optimum gain K_k expression. In the Kalman implementation, the Joseph's form is used to help compensate numerical round-off errors and other approximations that would make the computed K_k not to be the theoretical optimum.

The Kalman equations can be derived and re-written in different equivalent ways, the above implementation is an often used choice that provides a good balance between equation readability and adequate numerical performance. More numerically stable algorithms are for example the square-root form [10].

Generic implementation

The following MATLAB® code is a generic implementation of the Kalman filter.

```
function [ x_est, r ] = kalman_generic( meas, model_d, x0, P0)
N = length(meas);
Phi = model_d.Phi;
Gamma = model_d.Gamma;
H   = model_d.H;
Q   = model_d.Q;
R   = model_d.R;
n = size(Phi,1);
m = size(H,1);
Q_= Gamma*Q*(Gamma');
% MATLAB preallocation
x_est = zeros(n,N+1);
r = zeros(m,N);                    % Sequence of residuals

x_est(:,1) = x0;                   % Initial state estimation
P = P0;                            % Initial covariance

for k=1:1:N                        % Kalman filter recursive equations
    x = Phi*x_est(:,k);            % Predict the state x̂_k^-
    z = H*x;                       % Measurement prediction ẑ_k
    P = Phi*P*(Phi.') + Q_;        % A priori error covariance P_k^-
    K = P*H'/(H*P*H'+R);           % Compute the Kalman gain K_k
    r(:,k)= meas(k,:) - z;         % Residual (or innovation)
    x_est(:,k+1) = x + K*r(:,k);   % Update estimate with measurement x̂_k^+
    %P =(eye(Ns+1)-K*H)*P;         % Update error covariance P_k^+ Standard Form
    P=(eye(n)-K*H)*...             % A posteriori error covariance P_k^+ Joseph Form
      P*(eye(n)-K*H)'+ K*R*(K');
end
```

4.3.5 Riccati equations

The uncertainty of the Kalman estimate can be further analyzed by studying the covariance equations. The uncertainties of the *a priori* estimates are given by the covariance matrix sequence P_k^- and those of the *a posteriori* estimate are given by the the covariance sequence P_k^+.

It can be noticed that the covariances and the Kalman gain computations do not depend on the particular measurement values, they are intrinsic properties of the system and depend only on model parameters. Since the covariance expresses the uncertainty of the estimate, the filter performance can be calculated and analyzed in the design phase.

Furthermore, the *a posteriori* estimate covariance equation (4.34) can be rewritten in a standalone recursive form in variable P^+ by replacing the expression of P_k^- and K_k. The following notation is used next for simplification:

$$\bar{Q} = \Gamma Q \Gamma^T \tag{4.36}$$

$$\begin{aligned} P_k^+ &= (I - K_k H) P_k^- \\ &= \left(I - (\Phi P_{k-1}^+ \Phi^T + \bar{Q}) H^T [H(\Phi P_{k-1}^+ \Phi^T + \bar{Q}) H^T + R]^{-1} H \right) (\Phi P_{k-1}^+ \Phi^T + \bar{Q}) \\ &= \Phi P_{k-1}^+ \Phi^T - (\Phi P_{k-1}^+ \Phi^T H^T + \bar{Q} H^T)[H \Phi P_{k-1}^+ (H\Phi)^T + \dots \\ &\quad \dots + HQH^T + R]^{-1} (H \Phi P_{k-1}^+ \Phi^T + H\bar{Q}) + \bar{Q} \end{aligned} \tag{4.37}$$

Similarly, the *a priori* estimate covariance equation (4.34) can be rewritten in a standalone recursive form in variable P^- by replacing the expression of P_{k-1}^+ and K_k.

$$\begin{aligned} P_k^- &= \Phi P_{k-1}^+ \Phi^T + \bar{Q} \\ &= \Phi (I - K_{k-1} H) P_{k-1}^- \Phi^T + \bar{Q} \\ &= \Phi P_{k-1}^- \Phi^T - \Phi P_{k-1}^- H^T (H P_{k-1}^- H^T + R)^{-1} H P_{k-1}^- \Phi^T + \bar{Q} \end{aligned} \tag{4.38}$$

The evolution in time of the covariance matrices is of interest, and a special case is when the the case when they converge over time to a bounded value. By replacing both the terms P_k and P_{k+1} with a single unknown matrix $X = P_\infty$, an algebraic equation is obtained from the dynamic one. The general form of the discrete algebraic Riccati equation is:

$$E^T X E = A^T X A - (A^T X B + S)(B^T X B + V)^{-1}(B^T X A + S^T) + W \tag{4.39}$$

The *a posteriori* equation is a particular case of discrete Riccati algebraic equation where $E = I; A = \Phi^T, B = \Phi^T H^T, S = \bar{Q} H^T, V = R, W = \bar{Q}$ by noting that matrix $\bar{Q}$ is symmetric by construction since Q is a covariance matrix. Similarly, the *a priori* algebraic equation is a particular case of Riccati equation where $E = I; A = \Phi^T, B = H^T, S = 0, V = R, W = \bar{Q}$.

Under certain conditions, the algebraic Riccati equation is guaranteed to have a relevant solution, P. It is sufficient to look only at the *a priori* equation, since the solution of the *a posteriori* equation can then be obtained by (4.34). A common form and notation of the discrete Riccati algebraic equation is:

$$P = A^T P A - (A^T P B)(B^T P B + R)^{-1}(B^T P A^T) + Q \tag{4.40}$$

P is a relevant solution if it is semi-positive definite, so that it can be a covariance matrix for a real system, and if stabilizes the filter matrix $(I - KH)\Phi$, which is discussed in the next section. There are theorems that state both necessary and sufficient conditions for the existence of relevant solutions for the algebraic Riccati equation, and also for uniqueness. We can recommend NI [18] as a concise summarizing article for the continuous algebraic Riccati equation. The presentation here is not rigorous, but highlights an important point in the context of the project:

A necessary condition for the existence of a relevant solution P is that the matrix pair (Φ, H) is detectable.

The second necessary condition requires that eigen values of the system matrix Φ which are on the margin of stability, the unit circle, must be controllable. If these necessary conditions are met, they are also sufficient to guarantee at least one solution P, and the sequence P_k of uncertainties from the Kalman filter will converge to one of the semi-positive matrix solutions P. For uniqueness, the second necessary condition becomes slightly stronger, requiring that the matrix pair (Φ, Q_c) is stabilizable, where Q_c is the Cholesky decomposition of matrix Q, $Q = Q_c Q_c^T$.

The actual solution P of the Riccati algebraic equation cannot usually be expressed in a closed form except for simpler and more particular cases, however it can be calculated numerically. When it exists, the MATLAB® function `dare()` can compute the unique stabilizable solution of the discrete Riccati algebraic equation.

It is noted here, that the useful algebraic Riccati equation also appears in the study of the Linear-quadratic-Gaussian control problem (LQG), which is has a formulation duality with the linear-quadratic estimation problem, i.e. the Kalman filter.

4.3.6 Convergence of the filter

The Kalman filter can be considered as a system. The state of the filter is the estimation error, and the dynamic equation can be expressed in the following way:

$$\begin{aligned}
\hat{\mathbf{x}}_k^+ - \mathbf{x}_k &= \hat{\mathbf{x}}_k^- + K_k(\mathbf{z}_k - \hat{\mathbf{z}}_k) - \mathbf{x}_k \\
&= \Phi\hat{\mathbf{x}}_{k-1}^+ + K_k(H\mathbf{x}_k + \mathbf{v}_k - H\hat{\mathbf{x}}_k^-) - \Phi\mathbf{x}_{k-1} - \Gamma\mathbf{w}_{k-1} \\
&= \Phi(\hat{\mathbf{x}}_{k-1}^+ - \mathbf{x}_{k-1}) + K_k(H\Phi\mathbf{x}_{k-1} + H\Gamma\mathbf{w}_{k-1} - H\Phi\hat{\mathbf{x}}_{k-1}^+) + K_k\mathbf{v}_k - \Gamma\mathbf{w}_{k-1} \\
&= \Phi(\hat{\mathbf{x}}_{k-1}^+ - \mathbf{x}_{k-1}) + K_k H\Phi(\mathbf{x}_{k-1} - \hat{\mathbf{x}}_{k-1}^+) + K_k H\Gamma\mathbf{w}_{k-1} - \Gamma\mathbf{w}_{k-1} + K_k\mathbf{v}_k \\
&= (I - K_k H)\Phi(\hat{\mathbf{x}}_{k-1}^+ - \mathbf{x}_{k-1}) + (K_k H - I)\Gamma\mathbf{w}_{k-1} + K_k\mathbf{v}_k \\
\epsilon_k &= (I - K_k H)\Phi\epsilon_{k-1} - (I - K_k H)\Gamma\mathbf{w}_{k-1} + K_k\mathbf{v}_k
\end{aligned} \tag{4.41}$$

If the filter system is stable, by definition, the state $\hat{\epsilon}_k$ will tend to 0 over time. Thus the filter estimation will converge to the real state.

As noted in Jazwinski [14], derivation of the Kalman filter uses conditions for optimality that do not necessarily imply stability. Stability must be addressed separately.

If the algebraic Riccati equation has solution, $P_\infty^- = P^-$ can be used to compute a constant Kalman gain:

$$K = P^- H^T (H P^- H^T + R)^{-1} \tag{4.42}$$

The constant expression is helpful for implementation as it reduces the complexity of the calculations performed. The filter equations are greatly simplified:

$$\begin{aligned} \hat{\mathbf{x}}_k^+ &= \Phi \hat{\mathbf{x}}_{k-1}^+ + K(\mathbf{z}_k - H\Phi\hat{\mathbf{x}}_{k-1}^+) \\ &= (I - KH)\Phi\hat{\mathbf{x}}_{k-1}^+ + K\mathbf{z}_k \end{aligned} \tag{4.43}$$

The Kalman constant gain also creates a stable $(I - KH)\Phi$ matrix. The minimum variance gain sequence is K_k and will converge to K. Using Kalman gain at the start of the filter might not be the optimum, however in many cases, the filter can be run directly with the constant K gain matrix, also through the transitory period, without affecting the stability or performance of the estimation. It is also possible to start the filter in the recursive form, and after a determined number of steps k to switch to the constant gain formula.

4.4 State Smoother

In the previous section the optimal *a posteriori* state estimate has been addressed as a key element in the Kalman filter. The *a posteriori* state estimate $\hat{x}_k^+$ at time k, is the state estimate $\hat{x}$ at time k based on all observations z up to and including k i.e.

$$\hat{x}_k^+ = \hat{x}_{k|k} \simeq E[x_k | z_1, \ldots, z_k] \tag{4.44}$$

This in short means that the state estimate generated by the Kalman filter, only depends on past and present observations. An alternate notation regarding the samples to which the estimate is conditioned is introduced as estimates are now conditioned on both past, present, and future observations.

A situation where future observations are accessible can however easily be imagined. Such situations can be imagined only for off-line estimation problems or for on-line problems where delays can be tolerated. Since additional observations are accessible in these situations, there is no reason why these observations should not be incorporate in the estimation of x and thereby obtain a more refined estimate. The problem can be described as follows

$$\hat{x}_{k|j} \simeq E[x_k | \mathbf{Z}_j] \qquad \text{where} \qquad j > k \tag{4.45}$$

Such estimators, which consider both past, current and future observations are called smoothers and can be divided into three general three types [9, sec. 4.13].

- **Fixed point smoothing**
  The state estimate $\hat{x}$ at a fixed time instance k is continuously updated as new observations become available, that is estimate $\hat{x}_{k_0|k}$ for $k = k_0, k_0 + 1, \ldots$. The quality of the estimate increases as more and more observations are considered.
- **Fixed lag smoothing**
  The state estimate $\hat{x}$ at time instance k is calculated based on fixed number of future observations c, where the time index k continuously changes as new observations become available. The lag c is a scalar constant, that describe number of future observations included in the estimate, i.e. the estimate is $\hat{x}_{k|k+c}$ for $k = 0, 1, \ldots$.

- **Fixed interval smoothing**
  A sequence of state estimates $\{\hat{x}_{1|N}, \ldots \hat{x}_{N|N}\}$ is calculated based on a fixed interval of observations $[z_1, \ldots, z_N]$. That is the estimate is $\hat{x}_{k|k_0}$ for $k = 0, \ldots, k_0$.

The reasoning of implementing a smoother in this project is to refine the state estimate, given that a prolonged recording of the gyro board output signal has been acquired. This approach is used in relation to identifying the process and measurement noise covariance matrices Q and R respectively. In the following section we present the smoothing algorithm used in this project.

4.4.1 Rauch-Tung-Striebel Smoother

The smoother implemented in this project is a Rauch-Tung-Striebel (RTS) smoother, which is a special implementation of a fixed interval smoother. The motivation for using this particular smoother type is that it enables re-usage of the Kalman filter as described in 4.3 and that the EM algorithm uses intermediate results from the smoother. The EM algorithm is used in the process of investigating the existence of sensor noise correlations in chapter 11 on page 85. process and measurement noise covariances.

The RTS smoother works in two distinct steeps as presented by [9].

• *Forward Pass:* Based on the observation sequence $\{z_1, z_2 \ldots, z_N\}$, with a standard Kalman filter compute the standard *a posteriori* estimates $\hat{x}_{k|k}$ and $P_{k|k}$.

• *Backward Pass:* Using the intermediate results from the forward pass $\{\hat{x}_{1|1} \ldots, \hat{x}_{N|N}\}$ and $\{P_{1|1} \ldots, P_{N|N}\}$, calculate $\hat{x}_{k|N}$ as

$$\hat{x}_{k|N} = \hat{x}_{k|k} + L_k \left(\hat{x}_{k+1|N} + \Phi \hat{x}_{k|k}\right) \tag{4.46}$$

$$P_{k+1|k} = \Phi P_{k|k} \Phi^T + Q \tag{4.47}$$

$$L_k = P_{k|k} \Phi^T \left(P_{k+1|k}\right)^{-1} \tag{4.48}$$

$$P_{k|N} = P_{k|K} + L_k \left(P_{k+1|N} - P_{k+1|k}\right) L_k^T \tag{4.49}$$

$\hat{x}_{k|N}$ and $P_{k|N}$ are the smoothed state vector and state covariance at time step k. L_k is the smoother gain at time step k, which specify to what degree the smoothed estimate should be corrected at time step k. The recursion is started from from the last time instance N with $\hat{x}_{k|N} = \hat{x}_{N|N}$ and $P_{k|N} = P_{N|N}$.

The RTS smoother is easily implemented in MATLAB®, as illustrated in the following. The function `rts.m` only implements the backwards pass. The forward pass is implemented by the standard Kalman filter.

```
1  function [Sx,SPpost,L] = rts(x,P,Phi,Q)
2  % function [xest,P,L] = rts(xest,P,Phi,Qest)
3
4  % INPUTS:   xest:   N_s × N matrix of N state estimate from Kalman filter, x̂_{k|k}.
5  %           P:      N_s × N matrix of N noise covariance from Kalman Filter, P_{k|k}.
6  %           Phi:    N_s × N_s state transition matrix, Φ.
7  %           Q:      N_s × N_s process noise covariance matrix, Q.
8  % OUTPUTS:  xest:   Smoothed state sequence, x_{k|N}
9  %           P:      moothed covariance sequence, P_{k|N}
10 %           L:      Smoother gain sequence, L_k
11
12 Ns = size(x,1);                                    % Size of data set N_s
13 N = size(x,2);                                     % Length of data set N
14 L = zeros(Ns,Ns,N);                                % Empty Smoother gain L
15 SPpost = zeros(Ns,Ns,N);                           % Smoothed covariance P_{k|N}
16 SPpost(:,:,N) = P(:,:,N);                          % Initeal condition P_{k|N} = P_{N|N}
17 Sx = zeros(Ns,N);                                  % Smoothed state vector x_{k|N}
```

```
18 Sx(:,N) = x(:,N);                                % Initeal condition x̂_{k|N} = x̂_{N|N}
19 for k=(size(x,2)-1):-1:1                         % Decreasing index
20     Ppri = Phi * P(:,:,k) * Phi' + Q;            % P_{k+1|k}
21     L(:,:,k) = P(:,:,k) * Phi' / Ppri;           % Smoother Gain L_k
22     Sx(:,k) = x(:,k) + L(:,:,k)*...              % x̂_{k|N}
23         (Sx(:,k+1) - Phi*x(:,k));
24     SPpost(:,:,k) = P(:,:,k) + L(:,:,k)*...      % P_{k|N}
25         (SPpost(:,:,k+1) - Ppri)*L(:,:,k).';
26 end
27 end
```

The function file `rts.m` is designed to take four inputs, the main being the estimated state vector $\hat{x}_{k|k}$ and the estimated noise covariance matrix $P_{k|k}$ for the collective sequence of estimates. The function also takes discrete state transition matrix Φ and the process noise covariance matrix Q. The function provides the smoothed state vector $\hat{x}_{k|N}$ and covariance matrix $P_{k|N}$. The MATLAB® function file `rts.m` can be viewed on the project CD.

In this section an overview of the RTS smoother has been provided. The smoother has been presented as a addition to the Kalman filter which refines the state estimate by considering the collective observation set given a finite data set.

4.5 Noise Correlations

In section 4.3 on page 25 it has been discussed how to obtain optimal *a posteriori* state estimate using the Kalman filter. When implementing the Kalman filter the quality of this estimate, to a large degree depends on the following filter parameters.

- Initial state x
- Initial error covariance matrix P
- System noise covariance matrix Q
- Process noise covariance matrix R
- System model

As the initial state x and initial error covariance matrix P are updated as the filter runs, the tuning of these parameters are less important to the filer performance. If the system model sufficiently models the dynamics of the system, the only tuning parameters are the system and process noise covariance. In most systems the value of these parameters are found trough hand tuning with acceptable results.

In this project this approach is however not a feasible solution, as the underlying assumptions of the project entail, that the process and measurement noise covariance matrices are known prior to filter implementation. The underlying assumption in the project is, that the process and measurement noise for the individual sensors are correlated, i.e. the value of the off-diagonal elements in the covariance matrices are nonzero.

To the groups knowledge no dedicated analysis on the existence of such correlations is available in literature. A method of estimating the off-diagonal elements in the process and measurement noise covariance matrices must be designed and implemented, to validate the existence of such correlations. The method utilized in this project is the Expectation Maximization (EM) method. The EM has been used by [1] and [16] in relation to estimating process and measurement noise covariance matrices with success. In the following an overview of the EM method and the framework under which it is applied, is presented.

4.5.1 Expectation Maximization

The EM method is a iterative parameter estimation algorithm that, under certain conditions, will converge to parameter values at a local maximum of the likelihood function. As such it falls into the framework of maximum-likelihood estimation and is used in situations where part of the data can be considered to be incomplete or hidden.

Consider a finite sequence of d-dimensional vectors $Z = \{z_1, z_2, \dots z_N\}$ which are independent and drawn from the distribution p. The density function $p(z|\Theta)$ is governed by a set of parameters Θ i.e. p might be a set of Gaussians and Θ might be the means and covariances. The resulting density for the sample will be:

$$p(Z|\Theta) = \prod_{i=1}^{N} p(z_i|\Theta) = \mathcal{L}(\Theta|Z) \tag{4.50}$$

The function $\mathcal{L}(\Theta|Z)$ is the likelihood function i.e. the likelihood of the parameters given the observed data. In this way the likelihood can be thought of as a measure of the likelihood of a set of parameter values given a sequence of observations. In the maximum likelihood problem the objective is to find a set of parameters $\hat{\Theta}$ that maximizes L i.e.

$$\hat{\Theta} = \arg\max_{\Theta} \mathcal{L}(\Theta|Z) \tag{4.51}$$

In many situations it is easier to work with the log-likelihood $\log(L(\Theta|Z))$. The reasoning for using the log-likelihood is that it is easier to maximized than the likelihood, which is a property utilized in the EM method.

Consider a sequence of observation vectors $\{z_1, z_2, \dots z_N\}$, each consisting of N_s rate measurement from the project gyroscopes i.e. $z_i = [z_{1,i}, z_{2,i}, \dots z_{N_s,i}]^T$. The distribution p which the observation vector is drawn from is governed by the process and measurement noise covariance matrices. The parameters of Q and R can, in accordance with the above presented, be estimated by performing the following maximizing:

$$\left(\hat{Q}, \hat{R}\right) = \arg\max_{Q,R} \log \mathcal{L}(z_0, \dots, z_N|Q, R, \text{Model}) \tag{4.52}$$

The EM algorithm uses the smoothed state estimate $\hat{x}_{k|N}$ and covariance $P_{k|N}$ as presented in section 4.4 on page 32. The EM algorithm iteratively arrives at a local maximum of the likelihood function by applying two distinct steps to the RTS smoother output. In the first step, known as the E step, the expectation of the log-likelihood evaluated using the current estimate for the latent variables. In the second step, know as the M step, the parameters maximizing the expected log-likelihood found in the E step are computed. These estimates are then used to determine the distribution of the latent variables in the next E step. Using the newly estimated parameters, the smoothing operation is repeated to generate a new sequence of smoothed state estimates to which the EM algorithm can be applied.

In the following the concept of the EM algorithm are illustrated. The equations used to update the noise covariance matrices are presented without derivation [1]. Notice that the equations condense the E and M step into a single operation.

- Initialize $\hat{Q}$ and $\hat{R}$ with initial guess
- Run Kalman Filter and obtain $\hat{x}_{k|k}$ and $P_{k|k}$
- Run RTS smoother and obtain $\hat{x}_{k|N}$, $P_{k|N}$ and L_k

- Update $\hat{Q}$ and $\hat{R}$ as follows [1]

$$\hat{Q} = \frac{1}{N}\sum_{k=0}^{N-1}\left(\hat{x}_{k+1|N} - \Phi\hat{x}_{k|N}\right)\left(\hat{x}_{k+1|N} - \Phi\hat{x}_{k|N}\right)^T + \Phi P_{k|N}\Phi^T + \dots$$
$$\dots + P_{k+1|N} - P_{k+1|N}L_k^T\Phi^T - \Phi L_k P_{k+1|N} \tag{4.53}$$
$$\hat{R} = \frac{1}{N+1}\sum_{k=0}^{N}\left(y_k - C\hat{x}_{k|N}\right)\left(y_k - C\hat{x}_{k|N}\right)^T + CP_{k|N}C^T \tag{4.54}$$

The iteration is repeated until parameter convergence. One method of detecting convergence is to compute the value of the state estimate log-likelihood after each iteration and stop iterating when performance in not increased significantly from one iteration to the next [21]. The process of calculating the state estimate log-likelihood is addressed in the following section

State Estimate log-likelihood

The state estimate log-likelihood is a standard method of evaluating the performance of Kalman filters. The method is based on calculating the likelihood of the estimates, based on the observation sequence $\{z_1, z_2 \dots, z_N\}$.

The equations for calculating the state estimate log-likelihood are presented here without derivation [16].

$$\mathbb{P}(z_0, z_1, \dots, z_N) = \mathbb{P}(z_0)\mathbb{P}(z_1|z_0)\mathbb{P}(z_2|z_{0:1})\dots\mathbb{P}(z_N|z_{0:N-1}) \tag{4.55}$$
$$= \prod_{k=1}^{M}\frac{1}{(2\pi)^{d/2}|S|^{1/2}}e^{-\frac{1}{2}\left(z_{k+1}-\hat{z}_{k+1|k}\right)^T S^{-1}\left(z_{k+1}-\hat{z}_{k+1|k}\right)} \tag{4.56}$$
$$= \sum_{k=0}^{M}\left\{\log\left(\frac{1}{(2\pi)^{d/2}|S|^{1/2}}\right) - \frac{1}{2}\left(z_{k+1} - \hat{z}_{k+1|k}\right)^T S^{-1}\left(z_{k+1} - \hat{z}_{k+1|k}\right)\right\}$$
$$= \sum_{k=0}^{M}\left\{-\frac{d}{2}\log(2\pi) - \frac{1}{2}\log(|S|) - \frac{1}{2}\left(z_{k+1} - \hat{z}_{k+1|k}\right)^T S^{-1}\left(z_{k+1} - \hat{z}_{k+1|k}\right)\right\}$$

where d is the dimension of the observation vector z, $S = CP_{k+1|k}C^T + R$ and $|S|$ is the determinant of S. The log likelihood is used in the process of evaluating to what degree the state estimation is improved for each iteration of the EM algorithm. This in effect becomes a measure that indicate whether or not the covariance estimates are improved at each iteration. If the performance is not increased from one iteration to the next the estimate is assumed to have converged.

In should be noted that there is no guarantee that the estimate will converge to a global maximum and the algorithm should be run several times using different initialization. In the following implementation of the EM algorithm the initialization of the estimated noise covariances Q and R are generated using the function `sim_rand_correlations.m`. The function provides a random correlation matrix for the process noise $w(t)$ and measurement noise $v(t)$, based on a normal distribution. The function `sim_rand_correlations.m` is addressed in chapter 6 on page 49. The generated correlation matrices are transformed to covariance

matrices using the following transform:

$$\hat{Q} = \begin{bmatrix} \sigma_{w,1} & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & \sigma_{w,N_s} \end{bmatrix} \cdot Q_{corr} \cdot \begin{bmatrix} \sigma_{w,1} & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & \sigma_{w,i} \end{bmatrix} \tag{4.57}$$

$$\hat{R} = \begin{bmatrix} \sigma_{v,1} & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & \sigma_{v,N_s} \end{bmatrix} \cdot R_{corr} \cdot \begin{bmatrix} \sigma_{w,1} & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & \sigma_{w,N_s} \end{bmatrix} \tag{4.58}$$

where N_s is the number of gyroscopes considered and $\sigma_{w,i}$ and $\sigma_{v,i}$ are the standard deviation for the process and measurement noise of the individual gyroscopes. The value of these two parameters can be obtained using the AV method as described in section 4.2 on page 22.

In the following the implementation of the MATLAB® Expectation Maximization (EM) algorithm is presented. The function `corrident.m` is designed to take four inputs the main being the data to be processed, the minimum number of the number of iterations used, and the number of individual initializations. The function also takes the standard deviation for the process and measurement noise as additional input arguments. In addition a plot flag can be provided so that the state log-likelihood is plotted for each iteration. The plot is a good measure of whether or not the estimate has converged. The function provides the estimated process and noise covariances $\hat{Q}$ and $\hat{R}$.The MATLAB® implementation of the EM algorithm described above is used in the process of investigating the existence of noise correlations between the project gyroscopes. The process of applying the function to data captured from the gyroscopes and evaluating the results are presented in chapter 11 on page 85.

In this section an overview of the EM method, as it applied in this project, has been provided. The method equations have been presented without derivation along with the MATLAB® implementation of the EM algorithm. The state estimate log-likelihood has been presented and implemented as a method of evaluating the convergence of the noise covariance estimates.

Chapter Conclusion

In this chapter a overview the methods used in this project have been presented. The theory of the methods has been presented in a condensed fashion while maintaining the main elements and relevances to the project. The objective of the chapter has not been an exhaustive analysis of the used methods, but to provide the reader with a clear understanding of the project plan of action and relevant methods. Method relevant theory and additional considerations, that where omitted in this chapter, can be found in the project appendices.

This page intentionally left blank

Chapter 5

MEMS Gyroscope Modeling

In this chapter the process of modeling the MEMS gyroscopes contained in the gyro board are presented. The chapter focus on modeling the stochastic characteristics of the gyroscopes. The chapter is divided into three main sections that address the different steps related to modeling the gyroscopes. In the first part a widely accepted method of modeling MEMS gyroscopes is presented as a point of reference. In the second part the project gyroscopes are characterized using the AV method and additional methods of signal analysis. The third and final part address the implementation and validation of a stochastic model, that model the characteristics of the project gyroscope.

For an ideal sensor, the sensor output is identical with that of the measured quantity of interest. For MEMS gyroscopes, which are rate measuring devices, the ideal sensor can be described as follows

$$z(t) = c \cdot \omega(t) \tag{5.1}$$

where $z(t)$ is the gyroscope output, often in units of volts and $\omega(t)$ is the true rate, often in units $[^\circ/s]$. The parameter c is a scalar constant, a proportionality term in appropriate units, whose value is given in gyroscope data sheet. In the following c=1 is assumed or equivalently that the measurement $z(t)$ has been already been transformed according to $z(t)/c$, and now has the same unit as the measured property, i.e. $[^\circ/s]$.

The model described above is a poor approximation for most real gyroscopes, where a number of errors cause the sensor output to differ in several ways from the measured quantity of interest. These errors, which can either be deterministic or random in nature, often render the relation between $z(t)$ and $\omega(t)$ significantly more complicated, than a simple proportional term. Deterministic errors, which among others include temperature dependencies, scale factor and constant biases, can normally be damped or removed with internal compensation. In example the project sensor [3] is compensated for different performance at different temperatures. For random errors, internal compensation is not an option as the errors are not deterministic, and as such can not be identified before they occur. The random errors are often related to external disturbances, forces, vibrations, or electric noise affecting the interface circuits [13].

5.1 Generic MEMS Gyroscope Model

One of the main sources of random errors in MEMS gyroscopes is white measurement noise at the gyroscope output. The effects of this error, in terms of limited accuracy at the rate

level, may be acceptable if rate is indeed the state of interest. However most often the angle and not the rate is the state of interest and this renders the effects of the measurement noise much more evident. By integrating the measured rate the effects of the measurement noise is accumulated and manifested as a random walk at the angle level. Due to this close coupling between the measurement noise and the angle, the measurement noise is often refereed to as Angular Random Walk (ARW). Another main sources of error in MEMS gyroscopes are due to small changes in the gyroscope bias, which changes over time as a random walk. As the error is manifested at the rate level, this error is often refereed to as Rate Random Walk (RRW).

In accordance with the above presented error sources, widely accepted MEMS gyroscopes model is presented in the following. The presented model includes both the effects of ARW and RRW, that is measurement noise and a slowly changing bias at the gyroscope rate level. The model is given as follows

$$z(t) = \omega(t) + b(t) + v(t) \tag{5.2}$$

$$\dot{b}(t) = w(t) \tag{5.3}$$

where $z(t)$ is the gyroscope output, $\omega(t)$ is the true rate, $b(t)$ is a slowly changing bias driven by the white noise $w(t)$ and $v(t)$ is white measurement noise. The two white noise processes $w(t)$ and $v(t)$ are assumed to have different variance. The units for the bias $b(t)$ is $[^\circ/s]$ and $[^\circ/s^2]$ for the white noise $w(t)$.

Together, equations (5.2) and (6.3) make up the generic stochastic MEMS gyroscope model, that will be used in simulations throughout the thesis. The purpose of this model is not perfectly characterize a single sensor, but to establish a adequate MEMS gyroscope model, that include the predominant random noise processes. To this extent the generic model is considered a minimum representation of the stochastic characteristics of a MEMS gyroscope.

One of the key aspects of this thesis is to investigate methods of combining measurements from multiple MEMS gyroscopes into a single rate estimate. For this process a representative model of the MEMS gyroscopes used in the project must be developed. The model is developed through examination of the stochastic characteristics of the gyroscopes output signals. In the following the process of characterizing the stochastic properties of the project gyroscopes is decried.

5.2 Allan Variance Analysis

As presented in the methods section 4.2 on page 22, the AV is a IEEE [13] accepted method of characterizing the random noise processes in rate sensors. In the following the method is applied to data captured from the project gyroscopes [3].

A total of two data sets where acquired for the analysis. The two data sets where acquired by sampling the project gyroscopes for 48 hours on two different occasions. In appendix F on page 133 a comprehensive description of the process of both acquiring the necessary data, processing the data and finally identifying the noise processes is given. In the following the main results of the AV analysis are presented.

The main result of the AV analysis are the Root Allan Variance plots by which both the type and magnitude of random noise processes, present in the captured data, can be identified. In figure 5.1 the Root Allan Variance plot generated from the second of the two acquired data sets is illustrated.

From figure 5.1 it can be seen, that the gyroscopes have an almost identical Root Allan Variance. This is in accordance with the expected results, as the sensors are of identical type, manufacture and working under identical operating conditions. From figure 5.1 it can also be seen, that the main noise processes in the gyroscopes are the ARW and RRW

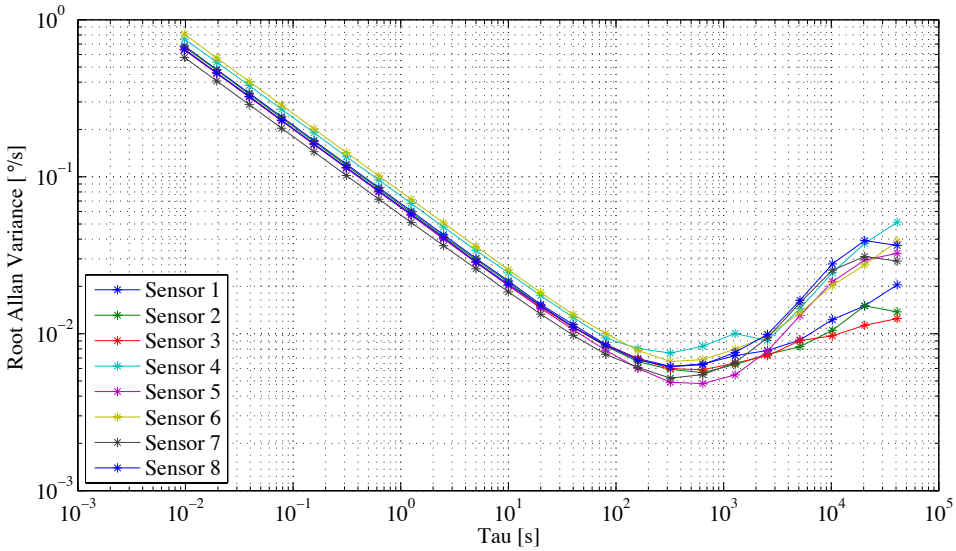

Figure 5.1: Root Allan Variance, gyro board Output 22 April - 25 April - AAU Test

processes. The two processes are identifiable by a slope of $-1/2$ and $1/2$ respectively. The Root Allan Variance plot generated from the first of the two acquired data sets is more or less identical to that presented in figure 5.1. The plot and additional notes on identifying the noise coefficients can be viewed in appendix F on page 133. The collective results of the AV analysis are presented in tables 5.1 and 5.2 which feature the identified noise coefficients.

From tables 5.1 and 5.2 it can be seen that the individual gyroscope noise coefficients

Gyroscope	Bias Stability $^\circ/s$	ARW_C $^\circ/\sqrt{s}$	RRW_C $^\circ/\sqrt{s^3}$
1	0.00674	0.07118	0.00032
2	0.00551	0.06882	0.00018
3	0.00674	0.06434	0.00027
4	0.00746	0.07874	0.00039
5	0.00533	0.06654	0.00028
6	0.00589	0.08422	0.00015
7	0.00631	0.06221	0.00033
8	0.00631	0.06654	0.00029
Mean value	0.00629	0.07032	0.00028

Table 5.1: Noise coefficients, gyro board Output 15 April - 18 April - AAU Test

only vary slightly between the two test sessions. The identified Bias Stability is found to be convincingly close to the typical value, $0.007^\circ/s$ given in the gyroscopes data sheet [3]. Based on this, and the fact that the standard deviation for the estimated coefficients is small, the estimates are considered to be sufficiently good estimates of the actual values. Several places in the thesis, what is considered to be typical values for the noise coefficients ARW_C and RRW_C are used in simulations. Based on the mean value for the two capture sessions, the typical values are set to the following

$$\text{ARW}_{C,TYP} \triangleq 0.068^\circ/\sqrt{s} \qquad \text{and} \qquad \text{RRW}_{C,TYP} \triangleq 0.00029^\circ/\sqrt{s^3} \tag{5.4}$$

In this section the AV method has been used to identify the random noise processes in the project gyroscopes [3]. The method has shown, as expected, that the main noise processes are the ARW and RRW. In relation to the ARW the AV method however suffers from inherent problems as other types of noise, such as xxx may also result in Root Allan Variance plots

Gyroscope	Bias Stability $^\circ/s$	ARW_C $^\circ/\sqrt{s}$	RRW_C $^\circ/\sqrt{s^3}$
1	0.00610	0.06434	0.00020
2	0.00589	0.06434	0.00019
3	0.00610	0.06434	0.00022
4	0.00746	0.07361	0.00043
5	0.00515	0.06015	0.00028
6	0.00674	0.08144	0.00034
7	0.00533	0.05437	0.00032
8	0.00631	0.06434	0.00035
Mean value	0.00613	0.06587	0.00029

Table 5.2: Noise coefficients, gyro board Output 22 April - 25 April - AAU Test

with a slope of $-1/2$. This means that the characteristic slope of $-1/2$ in figure 5.1 may not be caused by ARW noise process, but some other noise process, for instance a Gauss Markov process with short correlation time. To eliminate this source of error, the characteristics of the gyro board measurement noise $v(t)$, that is ARW is investigated further.

In the following section the measurement noise $v(t)$ is isolated by defining the operational conditions under which data is captured from the gyroscopes. Having isolated the measurement noise $v(t)$, the characteristics of the noise is analyzed to investigate to what degree the process can be considered to be white. If the measurement noise is indeed found to be white, it is considered substantiated that measurement noise $v(t)$ is in fact the source of the ARW noise component identified through the AV analysis.

5.3 Measurement Noise Analysis

In this section, the characteristics of the measurement noise $v(t)$ is investigated. The main objective is to substantiate that the measurement noise can be considered white as described by the AV analysis. In addition to investigating the whiteness, the distribution of the noise is also investigated. The objective is to substantiate that the measurement noise is Gaussian, i.e. normally distributed. The Gaussian distribution is an desired property, as it is a required criterion for the Kalman filter to provide the optimal estimate as described in chapter 8 on page 65.

The analysis of the measurement noise $v(t)$ is based on the generic model illustrated in 5.2 on page 40. By specifying the operational conditions under which the data for the analysis is acquired, the model can be drastically simplified. The bias process can be viewed to contain a deterministic component and a stochastic component, that is slowly changing. If relatively short periods time is considered the sensor bias can thus be approximated to consists of only a deterministic component, i.e. a scalar offset. The operational conditions is further specifying by defining the gyroscope to be motionless. By doing so, a stationary measurement where the angular velocity is constant i.e. $\omega(t) = 0$ can be acquired. This allows for further reduction of the generic model.

Under the described operational conditions the project gyroscopes can be modeled in the following model.

$$z(t)_i = b_i + v(t)_i \tag{5.5}$$

Where b_i is the bias for i^{th} sensor, which can be considered a deterministic offset. The value

of the bias is calculated as the sample mean as follows

$$b_i = \frac{1}{N}\sum_{k=1}^{N} z_i(k) \tag{5.6}$$

where i is the sensor number and N is the collective number of samples in the captured data. One question that can be raised is, for what sample duration the bias can be considered constant? To answer this question the theoretical bias drift caused by RRW must be calculated. The typical for the coefficient was found to be $\mathrm{RRW}_{C,TYP} = 0.00029°/\sqrt{s^3}$. The theoretical bias drift in $°/s$ after x seconds is calculated as follows

$$\sigma_w = \underbrace{\mathrm{RRW}_{C,TYP} \cdot \sqrt{\frac{2048}{20}}}_{\text{Scaled noise coefficient}} \cdot\sqrt{x} \tag{5.7}$$

Using equation (5.7) the theoretical after 120 seconds can be calculated as follows

$$\sigma_w = 0.00029 \cdot \sqrt{102.4} \cdot \sqrt{120} \tag{5.8}$$
$$= 0.032147 \tag{5.9}$$

From equation 5.8 it can be seen, that the theoretical bias drift after 120 seconds is approximately $0.032°/s$. This means that if a sample of only 120 seconds is considered sensor bias can be considered constant, as the bias drift caused by RRW will by negligible.

Based on the above considerations the gyroscopes have been sampled for 120 seconds at 102.4Hz. The used test setup and procedure is described in detail in appendix E on page 125. In the following the results and findings of the test is summarized. The results presented are generated by running the MATLAB® script file `signalcharacteristics.m` that can be viewed on the project CD. The gyroscope output signal is characterized using different signal evaluation methods. The main point of interest, that have been investigated, are listed here below.

- **Mean, Standard Deviation, RMS, Minimum and Maximum values**
  Provides amplitude information about the signal
- **Probability distribution**
  Provide information about statistical characteristics in amplitude domain
- **Autocorrelation**
  Provide information about statistical characteristics in the time domain
- **Amplitude Spectrum**
  Provide information about the statistical characteristics in the frequency domain

In table 5.3, the results of the signal sample mean, standard deviation, Minimum, Maximum and RMS values. The data histograms are plotted to investigate the statistical characteristics of the measurement noise. In figure 5.2 the histogram for sensor 6 is illustrated along with a fitted Gaussian density function. As it can be seen from figure 5.2, measurement noise $v(t)$ is approximated quite well by a Gaussian distribution. The same is true for the additional gyroscopes. The collective histograms for the eight project gyroscopes can be viewed in appendix E on page 125. This finding is particular important as is a desirable property in relation to the performance of the Kalman filter, as will be addressed in later chapters.

The main assumption given in the former sections has been that the measurement noise $v(t)$ is white noise. This assumption is investigated by plotting the autocorrelation for the captured data. In figure 5.3 the autocorrelation for sensor 1 is presented. The 95% confidence limits, illustrated in blue, are calculated as $-1/n \pm 2/\sqrt{n} \approx \pm 2/\sqrt{n}$ in accordance with [17].

Gyroscope	Mean	Standard Deviation	Max	Min	RMS
1	-0.56031	0.79799	2.75811	-3.17595	0.79796
2	0.10571	0.81730	3.04447	-3.62219	0.81861
3	-0.46951	0.76365	3.39991	-3.12023	0.81865
4	0.84696	0.91002	3.91494	-3.41106	0.91143
5	-0.90311	0.77111	3.02765	-2.61337	0.91143
6	0.24428	0.94069	3.27220	-3.17468	0.94100
7	-0.52403	0.75907	2.72183	-3.21223	0.94101
8	0.36220	0.79537	2.93450	-3.14608	0.94106
Mean value		0.81940			0.88514

Table 5.3: Results from Measurement Noise Analysis

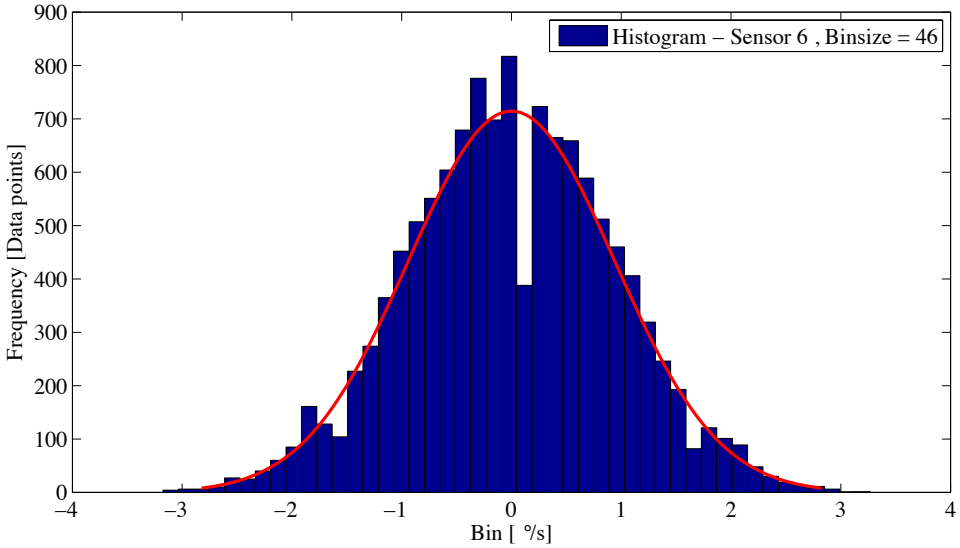

Figure 5.2: 120 Second 100Hz Histogram - AAU Test

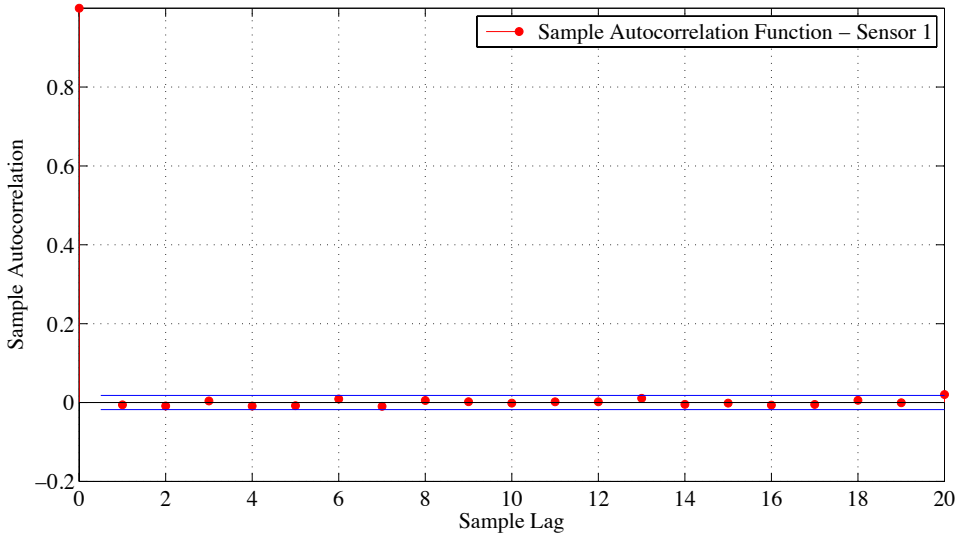

Figure 5.3: 120 Second 100Hz Autocorrelation - AAU Test

The autocorrelation plot 5.3 shows a peak of one at origin, and drops very fast to zero, showing no correlations, and no periodicity. The same is true for the additional gyroscopes as described in appendix E on page 125. As it is know that white noise processes have a flat amplitude spectrum, the amplitude spectrum of the captured data have been plotted. In figure 5.4 the amplitude spectrum for sensor 6 is illustrated.

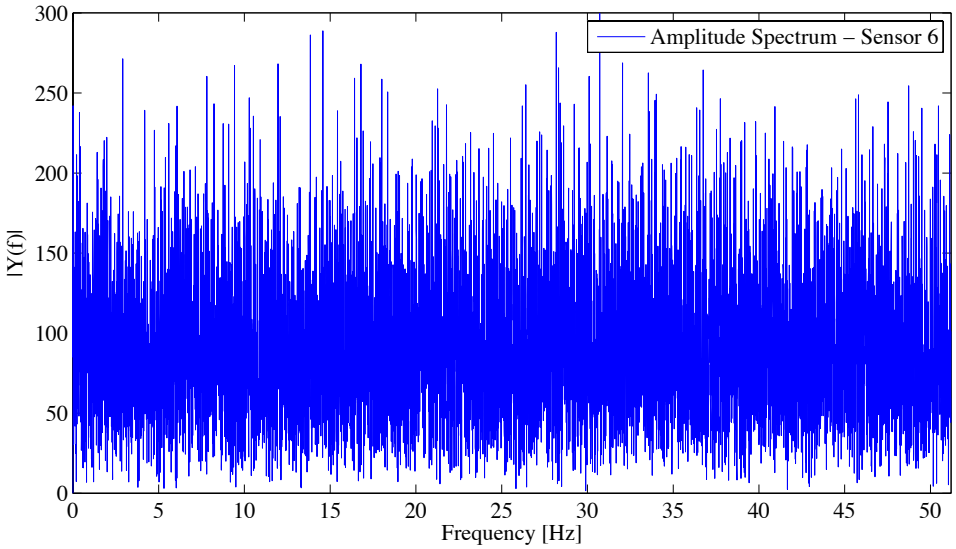

Figure 5.4: 120 Second 100Hz Amplitude Spectrum - AAU Test

The amplitude spectrum in figure 5.4 shows a approximately flat frequency band. The same is true for the additional gyroscopes as described in appendix E on page 125.

Based on the above presented results, as well as the additional results presented in appendix E it is considered proven that the gyroscope measurement noise $v(t)$ can be considered Gaussian white noise. The basis of this conclusion of the fact that the data histogram resemble a Gaussian distribution, the data autocorrelation shows no correlations and the data amplitude spectrum is relatively flat.

5.4 Model Simulation and Validation

This section describes the MATLAB® implementation and validation of the project gyroscope model. The model is designed in accordance with the generic model presented previously and includes only the stochastic characteristics of the project gyroscopes. The modeled noise processes, in accordance with the formerly presented results, are the ARW, RRW. As such this section is mainly a presentation of the MATLAB® implementation of the generic model, using the noise coefficients estimated in section 5.2 on page 40.

The MATLAB® implementation `allan.m` of the generic gyroscope model is illustrated here below.

```
1 function [data] = singlegyrosim(hours,Fs,ARWc,RRWc)
2 % function [data] = singlegyrosim(hours,Fs,ARWc,RRWc)
3 % INPUTS:       hours:      Length of simulation N          [h]
4 %               Fs:         Sample Frequency Fs             [Hz]
5 %               ARWc:       ARWC cofficient ARWC            [°/√s]
6 %               RRWc:       RRWC cofficient RRWC            [°/√s^3]
7 % OUTPUTS:      data:       [Time vector,simulated signal]  [s,°/s]
8
9 N = floor(Fs*60*60*hours);              % Number of data points
```

```
10 std_v = ARWc*sqrt(2048/20);              % σ for white noise v(t)
11 std_w = RRWc*sqrt(Fs);                   % σ for white noise driving w(t)
12 v =  std_v.*randn(N,1);                  % Generate white measurement noise v(t)
13 w =  std_w.*randn(N,1);                  % Generate bias noise w(t)
14 bias = zeros(N,1);                       % Vector containing bias b(t)
15 bias(:)=Integr(N,w,1/Fs);                % Integrate bias noise w(t), generating b(t)
16 data = zeros(N,2);                       % Vector containing simulated signal
17 data(:,1) = (0:1:(N-1))*1/Fs;            % Generate simulation time vector
18 data(:,2) = bias(:) + v(:);              % Generate simulated gyroscope signal
19 end
```

In order to simulate the output from the project gyroscope the function `allan.m` is called using the values for $\text{ARW}_{C,TYP}$ and $\text{RRW}_{C,TYP}$. The file `singlegyrovalidation.m` implements this action and further provides the figures and values used in the following validation of the sensor model. The file can be viewed on the project CD. The simulation provided by `singlegyrovalidation.m` consist of a 48 hour simulation with a sampling frequency of 102.4Hz. To this extent the simulation will generate data which is similar in size and structure, to the data captured from the actual gyroscope during the AV analysis. In figure 5.5 simulated data, along with actual gyroscopes data is illustrated

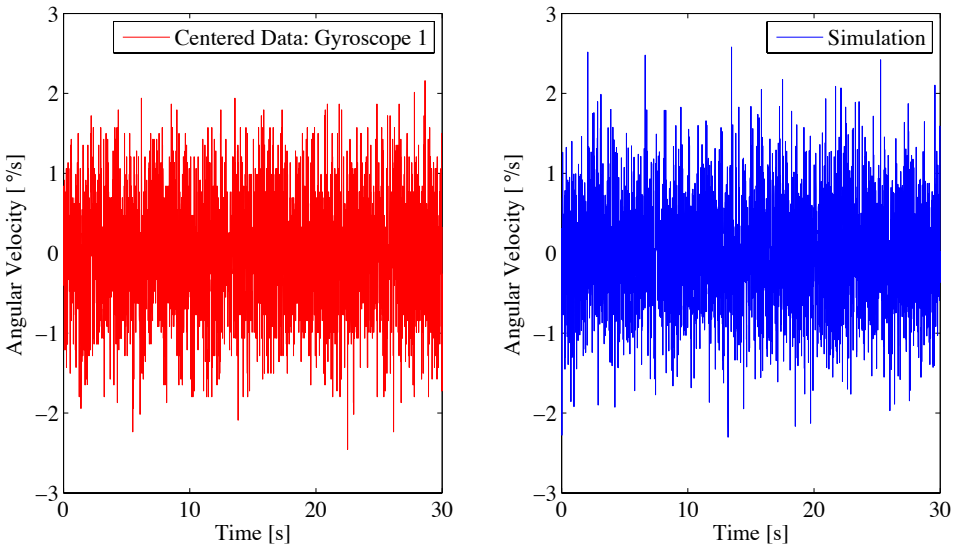

Figure 5.5: Actual gyroscope data and simulated data

From figure 5.5 it can be seen, that the simulated data at first glance is very similar to that of the actual gyroscope. The data from the actual gyroscopes has been centered by subtracting the mean, to better compare the two signals. The standard deviation for the first two 120 seconds data point where found to be $\sigma = 0.69035$. This is identical to the value found for the real gyroscopes. To validate that the simulation model actually model the random noise processes contained in the project gyroscopes, the Root Allan Variance has been calculated and plotted. In figure 5.6, the Root Allan Variance plot is illustrated.

The noise coefficients can be identified from figure 5.6 to be in accordance with the coefficient values given to the simulation model. The Bias Stability has not been modeled. From figure 5.6 it can however been seen that the Root Allan Variance plot shows a plateau at approximately the same level as the actual gyroscopes. To this extent the Bias Stability will not be addressed further.

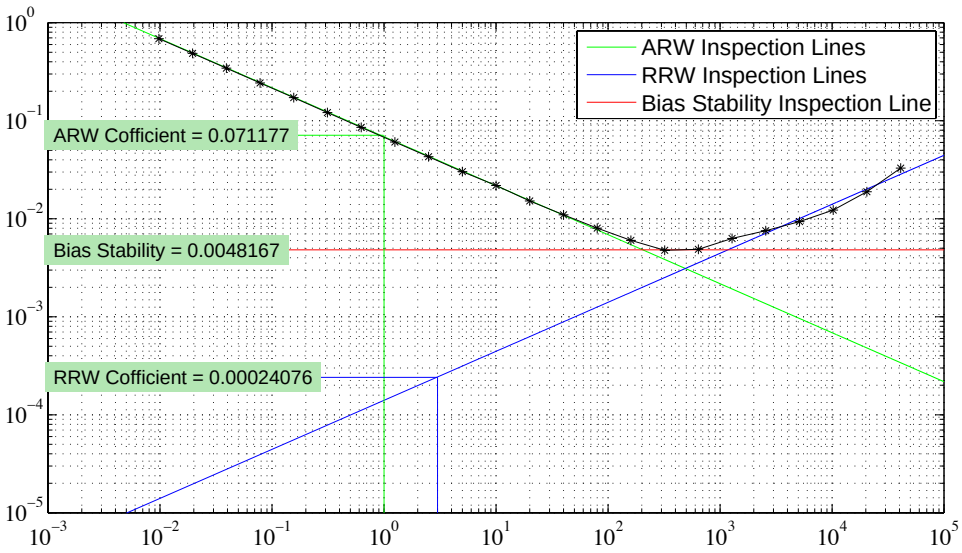

Figure 5.6: Root Allan Variance plot of 48 hour simulated data

Chapter Conclusion

In this chapter a model for the project gyroscopes have been developed. This has been done by analyzing the gyroscopes with the AV method. The main noise processes in the gyroscope data where found to be ARW and RRW. The characteristics of the measurement noise $v(t)$ where further investigated using standard methods of characterizing random signals. Based on this analysis it was concluded that the gyroscope measurement noise can be approximated by white Gaussian noise.

Based on the findings of the gyroscopes analysis a stochastic model for project gyroscopes have been developed. This model has been validated through a comparative analysis where the Root Allan Variance plot for data generated using the model where compared to that of the actual project gyroscopes.

In the following the framework under which the gyroscope model is implemented to form the collective gyro board model. The following chapter also address the concept of correlated gyroscope noise.

This page intentionally left blank

Chapter 6

State-space model

In this section the gyro-board state-space model and its properties are described. The model combines N_s individual sensor models, as developed in the chapter 5 on page 39 in a state space framework. The state space model will be used both in relation to various simulations and in the Kalman filter.

6.1 State-space formulation

A state space formulation is composed of two types of equations: the dynamics of the system states and the measurement (or observation) equation.

The main state in the system is the true angular rate ω that is to estimated. In defining the dynamic equation for the angular rate no assumptions can be made about the underlying forces, moments and accelerations because the sensor system can be attached to any system. In effect, the dynamic of the angular rate can only be assumed to be random.

To reflect this, the one-axis rate acceleration will be modeled by a stochastic term w_ω that represents a random, uncontrollable and unpredictable disturbances. An approximation will be made that this random component has a Gaussian distribution since this will be helpful later in the framework that optimizes the problem. In order not to limit the dynamic range of the sensor model, and also to express that this uncertainty of this modeling is high, the variance σ_ω will be taken to be excessively large.

Thus, the angular rate state is modeled as a random walk generated by a white noise with large variance. For the discrete representation with a sampling period T_s, a zero hold approximation is chosen for its simplicity:

$$\omega_{k+1} = 0 + T_s w_k, \quad \sigma_\omega >> 1 \tag{6.1}$$

In the measurement equation, the model in equation (5.2) is used where the output of each sensor is combination of the true angular motion, the measurement white noise and the bias.

$$z_k = \omega_k + b_k + v_k, \tag{6.2}$$

The white noise v_k is characterized either by the its standard deviation σ_v, or by the ARW coefficient ARW_c, as discussed in section 4.1.

The bias b_k is described as a slowly changing component with a random dynamic. This is a similar model as used for the angular rate, and differs only by the fact that the generating

white noise is limited to a small range. The random walk bias process is characterized either by the standard deviation of the generator white noise, σ_v or equivalently by the RRW coefficient RRW_c. The discrete difference equation is as follows:

$$b_{k+1} = 0 + T_s w_k, \quad \sigma_{w_i} < 1 \tag{6.3}$$

The sensor biases will be estimated by introducing them as states in the state-space model. A more detailed discussion about estimating both measurement and system noise elements by augmenting the system state with new variables can be found in [10].

There are N_s sensors in the configuration, and the gyro-board discrete state-space model is:

$$\begin{cases} \mathbf{x}_{k+1} & = \Phi \mathbf{x}_k + \Gamma \mathbf{w}_k \\ \mathbf{z}_k & = H \mathbf{x}_k + \mathbf{v}_k \end{cases} \tag{6.4}$$

The state space vectors are

$$\mathbf{x} = \begin{bmatrix} \omega \\ b_1 \\ b_2 \\ \vdots \\ b_{N_s} \end{bmatrix} \qquad \mathbf{w} = \begin{bmatrix} w_\omega \\ w_1 \\ w_2 \\ \vdots \\ w_{N_s} \end{bmatrix} \qquad \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{N_s} \end{bmatrix} \qquad \mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_{N_s} \end{bmatrix} \tag{6.5}$$

where $\mathbf{x}$ is the state vector containing the true angular rate and the biases of the sensors. $\mathbf{w}$ is the noise vector of the system dynamics and $\mathbf{v}$ is the noise vector of the measurement equation. $\mathbf{z}$ is the measurement vector containing the N_s outputs of the sensors. The vector sizes are as follows:

$$\mathbf{x}_k \in \mathbb{R}^{(N_s+1)\times 1}; \mathbf{z}_k \in \mathbb{R}^{N_s \times 1}; \mathbf{w}_k \in \mathbb{R}^{(N_s+1)\times 1}; \mathbf{v}_k \in \mathbb{R}^{N_s \times 1}; \tag{6.6}$$

The system, measurement and input-noise matrices are as follows

$$\Phi = \mathbf{I}_{N_s+1} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix} \tag{6.7}$$

$$H = \begin{bmatrix} \mathbf{1}_{N_s} & \mathbf{I}_{N_s} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & \dots & 0 \\ 1 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 1 \end{bmatrix} \tag{6.8}$$

$$\Gamma = T_s \mathbf{I}_{N_s+1} = \begin{bmatrix} T_s & 0 & \dots & 0 \\ 0 & T_s & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & T_s \end{bmatrix} \tag{6.9}$$

This state-space system describes gyro-board model where there are no deterministically known inputs and the dynamic of the measured property is random.

6.1.0.1 Deterministic and stochastic

The state space model presented above contains both deterministic and stochastic components. The stochastic part is represented by the elements $\mathbf{w}$ in the system dynamics equation and by the noise $\mathbf{v}$ in the measurement equation. The state $\mathbf{x}$ can be viewed to consist of

two components, a deterministic component $\mathbf{x}_d$ and a stochastic component, $\mathbf{x}_s$. The measurements vector $\mathbf{z}$ can be viewed in a similar way to contain a deterministic component and a stochastic component.

$$\begin{cases} \mathbf{x} &= \mathbf{x}^d + \mathbf{x}^s \\ \mathbf{z} &= \mathbf{z}^d + \mathbf{z}^s \end{cases} \tag{6.10}$$

The deterministic and stochastic parts of the system can be written separately. The deterministic dynamics and observations are given in the following equation.

$$\begin{cases} \mathbf{x}^d_{k+1} &= \Phi \mathbf{x}^d_k \\ \mathbf{z}^d_k &= H\mathbf{x}^d_k \end{cases} \tag{6.11}$$

Some estimation algorithms that operate only on the deterministic component of the model. The stochastic dynamics and observations are given by

$$\begin{cases} \mathbf{x}^s_{k+1} &= \Phi \mathbf{x}^s_k + \Gamma \mathbf{w}_k \\ \mathbf{z}^s_k &= H\mathbf{x}^s_k + \mathbf{v}_k \end{cases} \tag{6.12}$$

6.2 Lack of observability

Observability is the property of a deterministic system, that assures that all internal states can be known from the measurements and the sensor configuration. The difficulty with the gyro-board state space model (6.4) is that, although it describes the gyroscope behavior well, (Φ, H) is *not an observable* matrix pair.

Observability characterizes the possibility of recovering the initial state from a number of measurements, which in turn assures that any past state can be uniquely and precisely known from a (sufficiently long horizon) set of measurements.

$$\begin{aligned} \mathbf{z}_0 &= H\mathbf{x}_0 \\ \mathbf{z}_1 &= H\mathbf{x}_1 = H\Phi\mathbf{x}_0 \\ \mathbf{z}_2 &= H\mathbf{x}_2 = H\Phi^2\mathbf{x}_0 \\ &\cdots \\ \mathbf{z}_n &= H\mathbf{x}_n = H\Phi^{n-1}\mathbf{x}_0 \end{aligned} \tag{6.13}$$

or equivalently

$$\begin{bmatrix} \mathbf{z}_0 \\ \mathbf{z}_1 \\ \cdots \\ \mathbf{z}_n \end{bmatrix} = \begin{bmatrix} H \\ H\Phi \\ \cdots \\ H\Phi^{n-1} \end{bmatrix} \mathbf{x}_0$$

$$\mathbf{Z}_n = \mathbf{O}\mathbf{x}_0 \tag{6.14}$$

The dimensions are $\mathbf{Z}_n \in \mathbf{R}^{N_s n \times 1}$, $\mathbf{O} \in \mathbb{R}^{N_s n \times (N_s+1)}$, $\mathbf{x}_0 \in \mathbb{R}^{(N_s+1)\times 1}$. Using vector spaces and linear algebra, it can be shown that it is only important to study the case of $n = N_s + 1$, the number of states. To be able to uniquely solve the system and recover the initial state $\mathbf{x}_0$, the observability matrix $\mathbf{O}$ must have rank equal to the number of system states, $rank(\mathbf{O}) = N_s + 1$.

The rank of the observability matrix is as follows:

$$\mathbf{O}(\Phi, H) = \begin{bmatrix} H & H\Phi & H\Phi^2 & \dots & H\Phi^{N_s} \end{bmatrix}^T \tag{6.15}$$

can be shown to be N_s, smaller than the required $N_s + 1$. Thus, the system is not fully observable and it is not possible to recover the initial state from the measurements.

The interpretation is that the angular rate is present in every measurement accompanied each time by a different bias, and because its dynamics of are not known, it is not possible to distinguish if a changes in the measurement is caused by the bias or by the rate. As such, the lack of observability is caused because there are two unknown dynamics in the model, that of the rate and that of the bias states, and it is not possible to separate between them.

It can also be seen that since all the states are characterized by zero deterministic dynamics - all states considered constant by the model if excluding the stochastic component, and having the measurements as sums of the first state and respectively each of the subsequent states, it is not possible to uniquely determine $N_s + 1$ constants.

The lack of observability has negative consequences that affect the performance of state estimators algorithms.

6.2.1 Observable Decomposition

Lack of observability means that the states of the system are not sufficiently transparent in the measurement. Two or more different state values could translate into exactly the same measurement, and more, even knowing a past state, there are more possible transitions that can occur such that they will yield the same set of measurements. However, while it is not possible to know the entire internal state, the measurements still convey partial information. For some unobservable systems, it is possible to recover a number of internal states from the measurements using the observable decomposition. It is shown in this section that an observable decomposition is not useful for the gyro-board system since it is not possible to recover fully any of the states, but only partial information about each of the states.

The observability decomposition is done by use of a similarity transformation. A transformation matrix T that can be applied to the original system $(\mathbf{x}, \Phi, \Gamma, H)$ to obtain a new equivalent system $(\bar{\mathbf{x}}, \bar{\Phi}, \bar{\Gamma}, \bar{H})$ which is ordered to separate a set of n_O fully observable states. n_O is the rank of the observability matrix $\mathbf{O}(\Phi, H)$.

In general, a transformation T matrix, which must be invertible, acts on the system:

$$\bar{\mathbf{x}} = T^{-1}\mathbf{x}, \bar{\Phi} = T^{-1}\Phi T, \bar{H} = HT, \bar{\Gamma} = T^{-1}\Gamma \tag{6.16}$$

A particular transformation T can decompose the gyro-board system in the following way:

$$\bar{\Phi} = \begin{bmatrix} \bar{\Phi}_O & \bar{\Phi}_{12} \\ 0 & \bar{\Phi}_{nO} \end{bmatrix} \quad \bar{H} = \begin{bmatrix} 0 & \bar{H}_O \end{bmatrix} \tag{6.17}$$

where $\bar{\Phi}_O$ is square matrix of dimension N_s, and the pair $(\bar{\Phi}_O, \bar{H}_O)$ is completely observable.

The observability decomposition can be performed with the MATLAB® function `obsvf`.

```
Ns = 8;
Phi = eye(Ns + 1 );
Gamma = zeros(Ns+1,1);
H = [ eye(Ns) ones(Ns,1) ];
[Phi_bar, Gamma_bar, H_bar, Tm1, k ] = obsvf(Phi, Gamma, H);
```

The results are shown below. T is a full matrix, with no lines or columns 0, and matrices $\bar{\Phi}$

and $\bar{H}$ have the following form:

$$T^{-1} = \begin{bmatrix} -0.3333 & -0.3333 & -0.3333 & -0.3333 & -0.3333 & -0.3333 & -0.3333 & -0.3333 & 0.3333 \\ 0 & 0.8451 & 0.2318 & -0.2154 & -0.2154 & -0.2154 & -0.2154 & -0.2154 & 0 \\ 0 & 0.3780 & -0.8963 & 0.1037 & 0.1037 & 0.1037 & 0.1037 & 0.1037 & 0 \\ 0 & 0 & 0 & 0.7182 & -0.0075 & -0.0075 & -0.0075 & -0.6957 & 0 \\ 0 & 0 & 0 & -0.3039 & 0.0461 & 0.8529 & -0.2747 & -0.3204 & 0 \\ 0 & 0 & 0 & -0.4188 & 0.1624 & -0.0750 & 0.7730 & -0.4416 & 0 \\ 0.9354 & -0.1336 & -0.1336 & -0.1336 & -0.1336 & -0.1336 & -0.1336 & -0.1336 & 0 \\ 0 & 0 & 0 & -0.1282 & 0.8783 & -0.2587 & -0.3563 & -0.1352 & 0 \\ -0.1179 & -0.1179 & -0.1179 & -0.1179 & -0.1179 & -0.1179 & -0.1179 & -0.1179 & -0.9428 \end{bmatrix}$$

$$\bar{\Phi} = \begin{bmatrix} I_8 & 0 \\ 0 & I_1 \end{bmatrix}$$

$$\bar{H} = \begin{bmatrix} & 0 & 0 & 0 & 0 & 0 & 0.9354 & 0 & -1.0607 \\ & 0.8452 & 0.3780 & 0 & 0 & 0 & -0.1336 & 0 & -1.0607 \\ & 0.2318 & -0.8963 & 0 & 0 & 0 & -0.1336 & 0 & -1.0607 \\ 0_{8\times 1} & -0.2154 & 0.1037 & 0.7182 & -0.3039 & -0.4188 & -0.1336 & -0.1282 & -1.0607 \\ & -0.2154 & 0.1037 & -0.0075 & 0.0461 & 0.1624 & -0.1336 & 0.8783 & -1.0607 \\ & -0.2154 & 0.1037 & -0.0075 & 0.8529 & -0.0750 & -0.1336 & -0.2587 & -1.0607 \\ & -0.2154 & 0.1037 & -0.0075 & -0.2747 & 0.7730 & -0.1336 & -0.3563 & -1.0607 \\ & -0.2154 & 0.1037 & -0.6957 & -0.3204 & -0.4416 & -0.1336 & -0.1352 & -1.0607 \end{bmatrix}$$

By the observability decomposition, the system is partitioned in two subspaces: an observable one, characterized by the dynamics matrix $\bar{\Phi}_O$, and an unobservable one, characterized by the dynamics matrix $\bar{\Phi}_{nO}$. The states in the observable space,that is $(T^{-1}\mathbf{x})^{1:N_s}$, can always be recovered or estimated from the measurements.

However, the linear combination $(T^{-1}\mathbf{x})^{1:N_s}$ does not isolate any useful information. The observable states are all a linear mix of the biases and the true angular rate, and it is still not possible to isolate any individual state: a pure bias or the angular rate.

It is not possible to perfectly determine the states of the system because they are not transparent enough with the available sensor configuration. This is an inherent problem of the gyro-board configuration when trying to estimate the biases of the sensors.

6.2.2 Lack of Detectability

From the observable decomposition, the detectability property can be easily investigated. Detectability is a condition that assures the convergence of the Riccati equation. A system is called detectable if all its unobservable modes are asymptotically stable. The unobservable modes are the eigenvalues of the matrix $\bar{\Phi}_{nO}$.

The system is not detectable, there is only one unobservable eigenvalue that is however not asymptotically stable, since it is not strictly negative; it is 1. The P_k covariance sequence of the gyro-board Kalman filter might not be bounded, but can grow to infinity.

It is noted here, that another property called reconstructability can be defined in the following form: A discrete system is reconstructable if and only if all the eigenvalues of the unobservable subspace, eig($\bar{\Phi}_{nO}$), are 1. Reconstructability means that it is possible to know the current state $\mathbf{x}_n$ of a deterministic system when the initial state $\mathbf{x}_0$ is known.

The deterministic system (Φ, H) is reconstructible, however this property is not strong enough to be useful for the system. That is due to the stochastic part: since the stochastic noise-input $\mathbf{w}$ cannot be known, it is not possible to have reconstruction.

6.3 Stability

Stability is discussed so as to complete the set of properties of the system. When trying to apply different methods, prerequisites such as stability or observability might need to be checked, and thus it is of interest to have a good knowledge of the overall system properties.

There are two types of stability definitions that are discussed: Lyapunov and Bounded Input Bounded Output or BIBO.

The Lyapunov stability characterizes the trajectory line of the system state in time. The Lyapunov stability uses the notion of equilibrium point. Starting with at initial state $\mathbf{x}_0$

close to an equilibrium point $\mathbf{x}_E$, if the system naturally, in the absence of other inputs or disturbance, remains within a bounded distance to the equilibrium point, then it is Lyapunov stable. Further more, if the trajectory of the state converges and collapses back into the equilibrium point over time, then the system is called asymptotically stable.

In the multi-sensor state-space model, every state point is a equilibrium point. Given an initial condition $\mathbf{x}_0$, the system will naturally remain in this state in the absence of inputs or disturbances, due the lack of dynamics in the system. From the Lyapunov definition, the multi-sensor system is stable, and can even be considered asymptotically stable.

The BIBO definition characterizes a system from the input-output point of view. For any bounded input, a BIBO stable system will have a bounded output response. A practical equivalent condition is that all the eigen values of dynamic matrix of a discrete system must to have the modulus smaller than 1, with at most one value on the unit circle for marginal stability. This condition is not met by the multi-sensor system, as all the $N_s + 1$ eigen value of Φ are 1, and on the unit circle. From the BIBO definition, the system is not stable.

6.4 Variations in the state-space form

The presented states-space (6.4) is a generic form that models the basic properties of the gyro-board. The main aspect of this setup is that there are two types of unknown dynamics that must be considered. The property to be measured, and that of the biases present in each sensor. This combination of unknown dynamics creates the lack of observability and detectability.

The model can to be extended to contain further details and characterize specifics of a particular implementation, but this will not make the representation observable. In the following a few possible variations are considered.

The designed state-space model accommodates any number of sensors, as the block form of the system matrices and does not change. To account for the opposite orientations of the four sensors on the top side of the gyro-board board from the four sensors on the underside of the board, matrix H can take the following form:

$$H = \begin{bmatrix} \mathbf{1}_4 & \mathbf{I}_8 \\ -\mathbf{1}_4 & \end{bmatrix} \tag{6.18}$$

The system description remains unobservable.

Another situation related to the measurement equation can be considered. Although not present in the 100Hz signal acquired from the unit-gyroscope of our board, time-correlation and dynamics might be present at higher frequencies or for other types of sensors. For example, besides the bias, a sensor signal $z(k)$ can have a first order dynamic and a time delay.

$$x_{k+1} = 0.2x_k + \omega_k \tag{6.19}$$
$$y_{k+1} = x_k \tag{6.20}$$
$$b_{k+1} = b_k + w_k \tag{6.21}$$
$$z_k = y_k + b_k + v_k \tag{6.22}$$

In the notation above, the state variable x models the first order dynamics and y the time delay. A system of multiple identical sensors, with the rate $\omega(k)$ defined by a zero acceleration model, will still be unobservable. The state-space matrices for a two sensor system are shown

below:

$$\Phi = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0.2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.23}$$

$$\Gamma = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \tag{6.24}$$

Even more general, the each sensor might have different dynamic descriptions. The main conclusion is that as long every sensor measures the same property where no dynamics is assumed and each has a random walk type bias, the configuration will be 1-order short of observability.

6.5 Stochastic elements

The stochastic components of the gyro-board model, i.e. the noises $\mathbf{w}_k$ and $\mathbf{v}_k$ are modeled as random vector sequences. Each of the noise is an infinitely long ordered set of random variables $\{\mathbf{w}_k\}$ and respectively $\{\mathbf{v}_k\}$, where the k is the time step and the index used for the set, $k = \{0, 1, 2, ...\}$. For each k, $\mathbf{w}_k$ is a $(N_s + 1)$-variate random variable, and respectively $\mathbf{v}_k$ is N_s-variate. The notation $\mathbf{w}_k$ and $\mathbf{v}_k$ is also used for the realization of the random processes in a measurement (or simulation) session.

6.5.1 Expectation and covariance definitions for vector random variables

The definitions of the expectation vector and that of the covariance matrix of a vector random variable are noted below for reference. Using the notations x for a scalar random variable with density probability function $p(x)$, the expectation or the first order of a vector random variable $\mathbf{x}$ is a deterministic vector value:

$$\mathrm{E}\langle \mathbf{x} \rangle = \mathrm{E}\langle \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \rangle = \begin{bmatrix} \mathrm{E}\langle x_1 \rangle \\ \vdots \\ \mathrm{E}\langle x_n \rangle \end{bmatrix} = \begin{bmatrix} \int_{-\infty}^{+\infty} x_1 p(x_1) dx \\ \vdots \\ \int_{-\infty}^{+\infty} x_n p(x_n) dx \end{bmatrix} = \mu_{\mathbf{x}}$$

It is noted here, that although the time moments k are discrete, the variable x_i is continuous. The second order moment, that is the covariance is considered next.

$$\begin{aligned} \mathrm{Cov}(\mathbf{x}) &= \mathrm{E}\langle (\mathbf{x} - \mu_{\mathbf{x}})(\mathbf{x} - \mu_{\mathbf{y}})^T \rangle \\ &= \mathrm{E}\langle \begin{bmatrix} (x_1 - \mu_{x_1})^2 & (x_1 - \mu_{x_1})(x_2 - \mu_{x_2}) & \cdots & (x_1 - \mu_{x_1})(x_n - \mu_{x_n}) \\ (x_2 - \mu_{x_2})(x_1 - \mu_{x_1}) & (x_2 - \mu_{x_2})^2 & \cdots & (x_2 - \mu_{x_2})(x_n - \mu_{x_n}) \\ \vdots & \vdots & \vdots & \vdots \\ (x_n - \mu_{x_n})(x_1 - \mu_{x_1}) & \cdots & \cdots & (x_n - \mu_{x_n})^2 \end{bmatrix} \rangle \\ &= \begin{bmatrix} \mathrm{Var}(x_1) & \mathrm{Cov}(x_1, x_2) & \cdots & \mathrm{Cov}(x_1, x_n) \\ \mathrm{Cov}(x_1, x_2) & \mathrm{Var}(x_2) & \cdots & \mathrm{Cov}(x_2, x_n) \\ \vdots & \vdots & \vdots & \vdots \\ \mathrm{Cov}(x_n, x_1) & \cdots & \cdots & \mathrm{Var}(x_n) \end{bmatrix} \end{aligned}$$

The matrix $\mathrm{Cov}(\mathbf{x})$ is square.

For scalar values, it is also common to use a scaled covariance, the correlation, which can undertake values between -1and 1.

$$\mathrm{Cor}(x_1, x_2) = \frac{\mathrm{Cov}(x_1, x_2)}{\sigma_1 \sigma_2} \tag{6.25}$$

For vector random variables, the correlation matrix is:

$$\mathrm{Cor}(\mathbf{x}) = \begin{bmatrix} \sigma_{x_1} & 0 & \cdots & 0 \\ 0 & \sigma_{x_2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \sigma_{x_n} \end{bmatrix} \cdot \mathrm{Cov}(\mathbf{x}) \cdot \begin{bmatrix} \sigma_{x_1} & 0 & \cdots & 0 \\ 0 & \sigma_{x_2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & \sigma_{x_n} \end{bmatrix} \tag{6.26}$$

The cross-covariance of two random vector variables, $X \in R^m$ and $Y \in R^n$, is defined as:

$$\begin{aligned} \mathrm{Cov}(\mathbf{x}, \mathbf{y}) &= \mathrm{E}\langle (\mathbf{x} - \mu_{\mathbf{x}})(\mathbf{y} - \mu_{\mathbf{y}})^T \rangle \\ &= \mathrm{E}\langle \begin{bmatrix} (x_1 - \mu_{x_1})(y_1 - \mu_{y_1}) & (x_1 - \mu_{x_1})(y_2 - \mu_{y_2}) & \cdots & (x_1 - \mu_{x_1})(y_n - \mu_{y_n}) \\ (x_2 - \mu_{x_2})(y_1 - \mu_{y_1}) & (x_2 - \mu_{x_2})(y_2 - \mu_{y_2}) & \cdots & (x_2 - \mu_{x_2})(y_n - \mu_{y_n}) \\ \vdots & \vdots & \vdots & \vdots \\ (x_m - \mu_{x_m})(y_1 - \mu_{y_1}) & \cdots & \cdots & (x_m - \mu_{x_m})(y_n - \mu_{y_n}) \end{bmatrix} \rangle \\ &= \begin{bmatrix} \mathrm{Cov}(x_1, y_1) & \mathrm{Cov}(x_1, y_2) & \cdots & \mathrm{Cov}(x_1, y_n) \\ \mathrm{Cov}(x_2, y_1) & \mathrm{Cov}(x_2, y_2) & \cdots & \mathrm{Cov}(x_2, y_n) \\ \vdots & \vdots & \vdots & \vdots \\ \mathrm{Cov}(x_m, y_1) & \cdots & \cdots & \mathrm{Cov}(x_m, y_n) \end{bmatrix} \end{aligned}$$

Matrix $\mathrm{Cov}(\mathbf{x}, \mathbf{y})$ has size $m \times n$.

6.5.2 Assumptions about the system noise

The following assumptions about the random processes **w** and **v** are considered reasonable for the gyro-board problem. In general, the system noise w is less known than the measurement noise **v**.

- The system and measurement noises are Gaussian.
  Each scalar noise component w_k^i, and respectively v_k^i, is Gaussian, that any linear combination of the components is also Gaussian.

$$\begin{cases} \mathbf{w}_k \sim \mathcal{N}(\mu_{\mathbf{w}_k}, Q_k), \\ \mathbf{v}_k \sim \mathcal{N}(\mu_{\mathbf{v}_k}, R_k), \end{cases} \tag{6.27}$$

  The probability density functions for the multi-variate normal distributions has the following form:

$$f(\mathbf{w}_k) = \frac{1}{(2\pi)^{(Ns+1)/2} \det(Q_k)^{1/2}} \exp\left(-\frac{1}{2} (\mathbf{w_k} - \mu_{\mathbf{w}_k})^T Q_k^{-1} (\mathbf{w_k} - \mu_{\mathbf{w}_k)}) \right) \tag{6.28}$$

$$f(\mathbf{v}_k) = \frac{1}{(2\pi)^{Ns/2} \det(R_k)^{1/2}} \exp\left(-\frac{1}{2} (\mathbf{v_k} - \mu_{\mathbf{v}_k})^T R_k^{-1} (\mathbf{v_k} - \mu_{\mathbf{v}_k}) \right) \tag{6.29}$$

  Vectors $\mu_{\mathbf{w}_k}$ and $\mu_{\mathbf{v}_k}$ are the means and matrices Q_k and R_k are the covariance matrices of random vectors $\mathbf{w}_k$ and respectively $\mathbf{v}_k$, as defined in the previous section.

  Noise **v** has been shown in the modeling chapter 5 on page 39 to be approximately Gaussian. It is not possible to directly examine the noise **w** as it is hidden in the bias component of the measurement and hidden beneath the measurement noise **v**, which is approximately two-decimal orders bigger in size.

- The processes **w** and **v** are stationary.
  The vector-mean and covariance matrix are constant over time for both processes. Moreover, the mean is 0.

$$\mathrm{E}\langle \mathbf{w}_{k_1} \rangle = \mathrm{E}\langle \mathbf{w}_{k_2} \rangle = \mathbf{0}, \ \ \forall k_1, k_2 \tag{6.30}$$
$$\mathrm{E}\langle \mathbf{v}_{k_1} \rangle = \mathrm{E}\langle \mathbf{v}_{k_2} \rangle = \mathbf{0}, \ \ \forall k_1, k_2 \tag{6.31}$$
$$\mathrm{Cov}(\mathbf{w}_{k_1}) = Q_{k_1} = \mathrm{Cov}(\mathbf{w}_{k_2}) = Q_{k_2} = Q, \ \ \forall k_1, k_2 \tag{6.32}$$
$$\mathrm{Cov}(\mathbf{v}_{k_1}) = R_{k_1} = \mathrm{Cov}(\mathbf{v}_{k_2}) = R_{k_2} = R, \ \ \forall k_1, k_2 \tag{6.33}$$

  The assumptions about the mean are supported for the **v** noise, considering a calibration session to estimate the deterministic offset.
  The covariance assumptions need to be further investigated, and chapter 11 partially approaches the problem of identifying the behavior and exact values of matrices Q and R.

- The random processes **w** and **v** are independent.
  This implies that there are no correlations between the scalar variables $w^i_{k_1}$ and $v^j_{k_2}$, $\forall$ k_1, k_2, i, j. The cross-covariance matrix $\mathrm{Cov}(\mathbf{w}, \mathbf{v})$ is 0. This assumption is not directly addressed in this report, but could be investigated with noise identification techniques at the same time with the identification of the Q and R matrices.

- The component random processes of **w** and **v** are white.
  The noise sequences are not correlated in time.

$$\mathrm{E}\langle w_{k_1} w_{k_2} \rangle = Q\delta(k_2 - k_1), \ \ \forall k_1, k_2$$
$$\mathrm{E}\langle v_{k_1} v_{k_2} \rangle = R\delta(k_2 - k_1), \ \ \forall k_1, k_2$$

  $\delta(k)$ is the discrete impulse function. In section 5, noise **v** has been shown to be approximately white.

6.5.3 Structure of Q and R

In this section the structure of covariance matrices Q and R are addressed further. If the component variables of the random vector are independent, the matrices Q and R are diagonal.

$$Q = \mathrm{Cov}[\mathbf{w}_k] = \begin{bmatrix} \sigma^2_{w_\omega} & 0 & \cdots & 0 \\ 0 & \sigma^2_{w_1} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & 0 & \sigma^2_{w_{N_s+1}} \end{bmatrix}$$
$$R = \mathrm{Cov}[\mathbf{v}_k] = \begin{bmatrix} \sigma^2_{v_1} & \cdots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \cdots & \sigma^2_{v_{N_s}} \end{bmatrix}$$

If the matrices Q and R are full, it implies that there are correlations between the noises registered at the same time step k on the states and respectively on the measurements. The correlation can be caused for example if a common disturbance or noise source affects each of sensors similarly. Because the w_ω is used to model the uncertainty in the rotational dynamics

of the system, it is defined independent of the noises w_i.

$$
\begin{aligned}
Q = \mathrm{Cov}[\mathbf{w}_k] = \\
&= \begin{bmatrix} \sigma^2_{w_\omega} & 0 & 0 & \cdots & 0 \\ 0 & \sigma^2_{w_1} & \mathrm{Cov}(w_1, w_2) & \cdots & \mathrm{Cov}(w_1, w_{N_s}) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \mathrm{Cov}(w_{N_s}, w_1) & \mathrm{Cov}(w_{N_s}, w_2) & \cdots & \sigma^2_{w_{N_s}} \end{bmatrix} = \\
&= \begin{bmatrix} q_\omega & 0 & 0 & \cdots & 0 \\ 0 & q_{11} & q_{12} & \cdots & q_{1N_s} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & q_{N_s1} & q_{N_s2} & \cdots & q_{N_sN_S} \end{bmatrix} = \begin{bmatrix} q_\omega & \mathbf{0}^T \\ \mathbf{0} & Q_w \end{bmatrix}
\end{aligned} \tag{6.34}
$$

$$
\begin{aligned}
R = \mathrm{Cov}[\mathbf{v}_k] = \\
&= \begin{bmatrix} \sigma^2_{v_1} & \mathrm{Cov}(v_1, v_2) & \cdots & \mathrm{Cov}(v_1, v_{N_s}) \\ \vdots & \vdots & \vdots & \vdots \\ \mathrm{Cov}(v_{N_s}, v_1) & \mathrm{Cov}(v_{N_s}, v_2) & \vdots & \sigma^2_{v_{N_s}} \end{bmatrix} = \\
&= \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1N_s} \\ \vdots & \vdots & \vdots & \vdots \\ r_{N_s1} & r_{N_s2} & \cdots & r_{N_sN_s} \end{bmatrix}
\end{aligned} \tag{6.35}
$$

A particular case of time constant noise correlations is described below and will be used in some of the estimation examples. The variance of the $\mathbf{w}$ noises are considered equal, $\sigma^2_{w_i} = \sigma^2_w$, and also the variances of the $\mathbf{v}$ noises, $\sigma^2_{v_i} = \sigma^2_v$. For the system noise $\mathbf{w}$, the cross-correlation factor is taken equal among the different sensors, $\mathrm{Cov}(w^i_k, w^j_k) = \rho\sigma^2_w$, for any $i \neq j$. The noises $\mathbf{v}$ are uncorrelated. The matrices Q and R have the following form:

$$
Q_w = \mathrm{Cov}[\mathbf{w}_k] = \sigma^2_w \begin{bmatrix} 1 & \rho & \cdots & \rho \\ \vdots & \vdots & \vdots & \vdots \\ \rho & \rho & \cdots & 1 \end{bmatrix} \tag{6.36}
$$

$$
R = \mathrm{Cov}[\mathbf{v}_k] = \sigma^2_v \begin{bmatrix} 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix} \tag{6.37}
$$

Since a covariance matrix is always be positive definite, ρ is limited in the negative range.

$$
-\frac{1}{N_s - 1} \leq \rho \leq 1 \tag{6.38}
$$

As example, three sensors cannot be each in pair of twos be negatively correlated more than 0.5, or four sensors more than 0.3.

In conclusion, an important note is that in the estimation simulations presented next it is assumed that the covariance matrices Q and R are constant. At this point, it is not know if correlations between the gyro-board sensors are in fact exist. The accepted assumption is the absence of additional investigations is to consider that individually packaged sensor units are uncorrelated. In chapter 11 the problem of identifying correlations using an EM algorithm.

6.6 Generating multi-variate correlated white noise

MATLAB® and other software can be used to generate pseudo-random sequences of uncorrelated data with different distributions. In particular, it is possible to generate a sequence with

a given number of elements that has characteristics consistent with that of a sequence extracted from a normal standard distribution, a Gaussian distribution with parameters $\mu = 0$ and $\sigma = 1$. By dividing this sequence into parts, it is also possible to generate a number of independent random vectors, equivalent with the simulation of a multi-variable discrete white noise with normal distribution.

In order to simulate the stochastic part in the state space model, a method is needed to generate delta-correlated random sequences. The starting point is the n order random vector with independent components, each with a standard normal probability distribution.

$$\begin{aligned} \mathbf{X} &= \begin{bmatrix} x_1 & x_1 & \cdots & x_n \end{bmatrix}^T \\ \mathrm{E}\langle \mathbf{X} \rangle &= 0_{n\times 1} \\ \mathrm{Var}(\mathbf{X}) &= I_{n\times n} \end{aligned}$$

It is of interest to find a linear transformation A such that given c_{ij} delta-correlations are created:

$$\mathbf{Y} = A\mathbf{X} \tag{6.39}$$

$$\mathrm{E}\langle \mathbf{Y} \rangle = 0_{n\times 1}$$

$$\mathrm{Var}(\mathbf{Y}) = \begin{bmatrix} 1 & c_{12} & \cdots & c_{1n} \\ \cdots & \cdots & \cdots & \cdots \\ c_{n1} & c_{n2} & \cdots & 1 \end{bmatrix} \tag{6.40}$$

Finally, a transformation is needed to scale the standard deviation of each variable component to a given value σ_i. To find the linear transformation A, the vector variance equation can be detailed:

$$\begin{aligned} \mathrm{Var}(\mathbf{Y}) &= \mathrm{E}\langle YY^T \rangle \\ &= \mathrm{E}\langle (A\mathbf{X})(A\mathbf{X})^T \rangle \\ &= \mathrm{E}\langle A\mathbf{X}\mathbf{X}^T A^T \rangle \\ &= A\mathrm{E}\langle \mathbf{X}\mathbf{X}^T \rangle A^T \\ &= A\mathrm{Var}(\mathbf{X})A^T \\ &= AA^T \end{aligned}$$

It is also noted that because the random variables were considered normal standard, the variance matrix $\mathrm{Var}(\mathbf{Y}) = C$ is also the correlation matrix. Thus if the correlation matrix C is given, the linear transformation A can be found as:

$$C = AA^T \tag{6.41}$$

C is a correlation matrix, therefore it is symmetric since the element entries of the matrix are the cross-covariances of scalar random variables, and in particular also with mean zero mean and $\mathrm{E}\langle y_i y_j \rangle = \mathrm{E}\langle y_j y_i \rangle$, and positive definite, since all $\sigma_i > 0$. A can be found the Cholesky decomposition of matrix C.

Next, to scale the standard deviations, the needed transformation is:

$$\mathbf{V} = \begin{bmatrix} \sigma_1 & \cdots & 0 \\ \vdots & \cdots & \vdots \\ 0 & \sigma_i & 0 \\ \vdots & \cdots & \vdots \\ 0 & 0 & \sigma_n \end{bmatrix} \mathbf{Y} \tag{6.42}$$

The code steps needed in the simulation to generate a N long sequence of delta correlated noises w, and respectively v, are listed below.

```
Ns = 4;                              %# Number of random variables
dataI = randn(N,Ns);                 %# Random multi-variate independent sequences
                                     %# w/ standard normal distribution properties.
                                     %# Each row represents a realization,
                                     %# and each column is a variable.
A = chol(C)';                        %# The correlation matrix Cor is given
dataC = (A*(dataI'))';               %# Linear combination of each realization
dataS = (diag(sigma)*(dataC'))'      %# Scaling to the given standard deviations
```

Alternatively, in MATLAB® the function `mvnrnd` from the Statistics Toolbox can be used directly:

```
dataS = mvnrnd(zeros(N,Ns), diag(sigma)*C*diag(sigma));
```

Moreover, random correlation matrices can be generated by creating a symmetric matrix and normalizing it in the following way:

```
R=rand(Ns)-0.5;                 %# Create a random square matrix
R=R'*R;                         %# Create a symmetric covariance matrix
r=sqrt(diag(R));                %# Square root of the diagonal elements
C=diag(1./r)*R*diag(1./r);      %# Normalize the covariance matrix
```

Chapter 7

Estimation: Averaging

The estimation by averaging is unbiased and formulas for the variance of the estimate are presented. The noises are considered identical between sensors to simplify the calculations and obtain interpretable formulas. The performance of the averaging technique is shown to be dependent on the noise correlations. Overall negative correlations will give a better estimate than in the case of independent sensors, while overall positive correlations will give a worse estimate.

A practical way of estimating the true angular rate from N_s noisy measurements is to simply average the measurements. Consider a set of N_s sensors providing new measurements at each time step k, with a sample period of T_s. According to the developed multi-MEMS model:

$$\begin{cases} z_1(k) & = \omega(k) + b_1(k) + v_1(k) \\ z_2(k) & = \omega(k) + b_2(k) + v_2(k) \\ \dots & \\ z_{Ns}(k) & = \omega(k) + b_{Ns}(k) + v_{Ns}(k) \end{cases} \tag{7.1}$$

7.1 Variance of the individual measurement

By using the formulas for the variance and covariance of sum of random variables the expected values and variances of the measurements and estimates can be written as follows:

$$\mathrm{Var}\left(\sum_{i=1}^{n} x_i\right) = \sum_{i=1}^{n} \mathrm{Var}(x_i) + \sum_{i=1}^{n} \sum_{i=1;j\neq i}^{n} \mathrm{Cov}(x_i, x_j) \tag{7.2}$$

$$\mathrm{Cov}\left(\sum_{i=1}^{n} x_i, \sum_{j=1}^{m} y_j\right) = \sum_{i=1}^{n} \sum_{j=1}^{m} \mathrm{Cov}(x_i, y_j) \tag{7.3}$$

Regarding the random walk bias process b_i, the expected value is theoretically zero, $E[b_i(t)] = 0$, however the variance increases with time, and the probability that the measured value is close to zero becomes also smaller with time. As stated in section 6.5 on page 55, the generator noise w_i has a constant variance in time: $\mathrm{Var}\left(w_i(k)\right) = \sigma_{wi}^2$. The variance of the bias, a random walk process, has been already calculated in (4.6) as $kT_s^2\sigma_{w_i}^2$.

Since w_i and v_i are uncorrelated and consequently the process b_i and noise v_i are uncorrelated, the expected values and variances of the measurements can be expressed as:

$$E[z_i(k)] = \omega(k) + (E[b_i(k)] = 0) + (E[v_1(k)] = 0) = \omega(k) \tag{7.4}$$

$$\mathrm{Var}\,(z_i(k)) = 0 + \mathrm{Var}\,(b_i(k)) + \mathrm{Var}\,(v_i(k)) = kT_s^2\sigma_w^2 + \sigma_v^2 \tag{7.5}$$

7.2 Variance of the averaged estimate

The averaged estimate $\bar{\omega}$ defined below is accurate (unbiased) and more precise (lower in variance) than any of the individual measurements.

$$\begin{aligned}\bar{\omega}(t) &= \frac{z_1(k) + \cdots + z_{Ns}(t)}{N_s} = \\ &= \omega(t) + \frac{b_1(t) + \cdots + b_{Ns}(t)}{N_s} + \frac{v_1(t) + \cdots + v_{Ns}(t)}{N_s}\end{aligned} \tag{7.6}$$

To show this, the simplifying assumption that the sensor noises are identical between sensors will be made and maintained throughout the rest of the chapter. The v_i white noise components, and respectively the bias generating noises w_i, have the identical variance.

$$\begin{cases}\mathrm{Var}(v_i(k)) = \sigma_{vi}^2 = \sigma_v^2 \\ \mathrm{Var}(w_i(k)) = \sigma_{wi}^2 = \sigma_w^2\end{cases} \tag{7.7}$$

As it where the case for the measurements, the averaged estimate is unbiased, since b_i and v_i have each the expected value equal to 0.

$$\begin{aligned}E[\bar{\omega}(k)] &= \omega(k) + \frac{1}{N_s^2}E[b_1(k) + \cdots + b_{Ns}(k)] + \frac{1}{N_s}E[v_1(k) + \cdots + v_{Ns}(k)] = \\ &= \omega(k) + \frac{1}{N_s}(E[b_1(k)] + \cdots + E[b_{Ns}(k)]) + \frac{1}{N_s}(E[v_1(k)] + \cdots + E[v_{Ns}(k)]) = \\ &= \omega(k)\end{aligned} \tag{7.8}$$

Using again the fact that there are no correlations between the noises $\mathbf{w}$ and $\mathbf{v}$, the variance of the averaged estimate can be shown to have the following form:

$$\begin{aligned}\mathrm{Var}(\bar{\omega}(k)) &= 0 + \mathrm{Var}\left(\frac{1}{N_s}[b_1(k) + \cdots + b_{Ns}(k)]\right) + \mathrm{Var}\left(\frac{1}{N_s}[v_1(k) + \cdots + v_{Ns}(k)]\right) = \\ &= 0 + \frac{1}{N_s^2}\mathrm{Var}(b_1(k) + \cdots + b_{Ns}(k)) + \frac{1}{N_s^2}\mathrm{Var}(v_1(k) + \cdots + v_{Ns}(k)) \\ \mathrm{Var}(\bar{\omega}(k)) &= \frac{T_s^2}{N_s}k\sigma_w^2 + \frac{1}{N_s^2}\sum_{i=1}^{N_s}\sum_{j=1;i\neq j}^{N_s}\mathrm{Cov}[b_i(k), b_j(k)] + \\ &+ \frac{1}{N_s}\sigma_v^2 + \frac{1}{N_s^2}\sum_{i=1}^{N_s}\sum_{j=1;i\neq j}^{N_s}\mathrm{Cov}[v_i(k), v_j(k)]\end{aligned} \tag{7.9}$$

The sums in the formula represent the contributions of the instantaneous covariances or correlations of the noises $\mathbf{w}$ and $\mathbf{v}$ on the variance of the averaged estimate.

7.3 Uncorrelated sensors

If the noises $\mathbf{v}$ and respectively $\mathbf{w}$ are not delta-correlated, expression (7.9) is simplified since the terms $\mathrm{Cov}[v_i(k), v_j(k)]$ and $\mathrm{Cov}[w_i(k), w_j(k)]$ will disappear.

$$\mathrm{Var}(\bar{\omega}(k)) = \frac{1}{N_s}T_s^2k\sigma_w^2 + \frac{1}{N_s}\sigma_v^2 \tag{7.10}$$

The expression that we arrive at is a well known result in relation to averaging uncorrelated sensor signals. The rate averaged estimator $\bar{\omega}(k)$ has a standard deviation $\sqrt{N_s}$ times more precise than that of any of the individual measurements.

$$\mathrm{Var}(\bar{\omega}(k)) = \frac{1}{N_s}\mathrm{Var}(z_i(k))$$

It is noted that the variance of the averaged estimator, same as that of a single measurement, increases with time, since it depends on the step k.

7.4 Correlated sensors

Under the working assumptions of the multi-MEMS noises presented section 6.5, including that the correlations are constant in time and that the noise levels are identical between sensors, (7.9) can be further simplified.

$$\begin{cases} \mathrm{Cov}[w_i(k), w_j(k)] & = \mathrm{Cor}[w_i(k), w_j(k)]\sigma_w^2 = c_{i,j}^w(k)\sigma_w^2 = c_{i,j}^w\sigma_w^2 \neq 0 \\ \mathrm{Cov}[v_i(k), v_j(k)] & = \mathrm{Cor}[v_i(k), v_j(k)]\sigma_v^2 = c_{i,j}^v(k)\sigma_v^2 = c_{i,j}^v\sigma_v^2 \neq 0 \end{cases} \tag{7.11}$$

Since the bias terms $b_i(k)$ can be written as:

$$b_i(k) = \sum_{p=1}^{k} w_i(p)T_s \tag{7.12}$$

and using property (7.2), the variance of the averaged estimate becomes after some work:

$$\mathrm{Var}(\bar{\omega}(k)) = \frac{1}{N_s^2}T_s^2 k \left(\sum_{i,j}^{N_s,N_s} c_{i,j}^w \right) \sigma_w^2 + \frac{1}{N_s^2} \left(\sum_{i,j}^{N_s,N_s} c_{i,j}^v \right) \sigma_v^2 \tag{7.13}$$

The terms $(\sum c_{i,j})$ are the sums of correlation factors of the sensors, and in the case when noises $\mathbf{w}$, and respectively $\mathbf{v}$, are delta uncorrelated, they have each value N_s. For example, (7.10) can be recovered as a particular case of (7.13). It also noticed that the variance of the rate estimate is growing linearly with the time index k.

The variance components coming from both the noises $\mathbf{w}$ and $\mathbf{v}$ can be reduced compared to the uncorrelated case whenever $(\sum c_{i,j}) < N_s$. And the theoretical minimum value for $(\sum c_{i,j})$ is 0. When sufficiently many negative correlations are present between the sensors it can make the parenthesis terms to become zero, and obtain zero variance for the rate estimate.

In conclusion, the averaging estimate is dependent on the correlations present in the noises. It improves with overall negative correlations and it deprecates in the presence of overall positive correlations. It will be seen in the next chapters however, that a Kalman estimate, which weights differently the measurements and uses also past information, can have a better performance. Its variance will also go to 0 under good correlation and will improve "faster" with the correlations than the averaged estimator.

Another observation is that the contribution of the $\mathbf{v}$ noise in the angle drift over a longer period of time N is smaller than that of the bias generating noise $\mathbf{w}$, even considering a fast sampling rate or a small sampling period T_s and looking at the term T_s^2. In a long run, more angle drift improvement is gained from the $\mathbf{w}$ correlations than from the $\mathbf{v}$ correlations, because the $\mathbf{w}$ correlations improve the bias estimate which accumulates over time and this is the main contribution in the angle drift.

7.5 Variance of the angle drift

Formulas for the angle travel between time $k = 1$ and $k = N$ are calculated below by integrating the rate signal with a simple hold method. The variance in the angle is:

$$\mathrm{Var}(\alpha(N)) = \mathrm{Var}\left(\sum_{k=1}^{N} \omega_{est}(k) T_s\right) \tag{7.14}$$

The variance of the angle computed directly from a measurement is:

$$\begin{aligned}\mathrm{Var}(\alpha_i(N)) &= \mathrm{Var}\left(\sum_{k=1}^{N} z_i(k) T_s\right) = \\ &= \left(\frac{N(N+1)}{2} + 2\sum_{k=1}^{N-1} k(N-k)\right) T_s^4 \sigma_w^2 + N T_s^2 \sigma_v^2\end{aligned} \tag{7.15}$$

While the variance of the angle computed from the rate averaged estimation is:

$$\begin{aligned}\mathrm{Var}(\bar{\alpha}(N)) &= \mathrm{Var}\left(\sum_{k=1}^{N} \bar{\omega}(k) T_s\right) = \\ &= \left(\frac{N(N+1)}{2} + 2\sum_{k=1}^{N-1} k(N-k)\right) \frac{1}{N_s^2} T_s^4 \left(\sum_{i,j}^{N_s,N_s} c_{i,j}^{w}\right) \sigma_w^2 + N \frac{1}{N_s^2} T_s^2 \left(\sum_{i,j}^{N_s,N_s} c_{i,j}^{v}\right) \sigma_v^2\end{aligned} \tag{7.16}$$

The formulas has been also checked by simulation and match consistently.

It is noted that in the case of uncorrelated sensors, the improvement factor of the averaged angle estimation over that of from a single measurement is $\sqrt{N_s}$, expressed in standard deviation.

It is also noted that the angle drift dependency of the time-index k is nonlinear, and more complicated than in the case of the rate drift.

Chapter Conclusion

In this chapter exact analytical expressions of the rate and angle variances has been derived. Using these expressions it is now now possible to compare the averaging method rigorously with other estimation techniques.

Chapter 8

Estimation: Recursive Kalman

The Kalman filter, described in 4.3, is a discrete estimation algorithm that operates on a linear system with Gaussian noise to calculate the optimum state estimate from measurements. The multi-MEMS state space model fits into the Kalman framework, and the algorithm can be implemented. The multi-MEMS model, however, does not meet important requirements (observability or detectability) to assure the exponential stability of the Kalman filter and equivalently the convergence of the estimate to the true state. From simulations, it can be seen that the variance of the Kalman estimate P_k grows unbounded over time. Still, the Kalman filter remains the optimum estimator for this model and it is shown to have special properties.

8.1 Implementation

By using the particular expressions of the state-space matrices (6.4) in the recursive Kalman equations (4.30)-(4.34), the following particular filter is obtained, where some of terms that are reduced or simplified.

Predict step:

$$\begin{cases} \hat{\mathbf{x}}_k^- = \hat{\mathbf{x}}_{k-1}^+ & (8.1) \\ \hat{\mathbf{z}}_k = H\hat{\mathbf{x}}_k^- & (8.2) \\ P_k^- = P_{k-1}^+ + Q & (8.3) \end{cases}$$

Update step:

$$\begin{cases} K_k = P_k^- H^T (H P_k^- H^T + R)^{-1} & (8.4) \\ \hat{\mathbf{x}}_k^+ = \hat{\mathbf{x}}_k^- + K_k(\mathbf{z}_k - \hat{\mathbf{z}}_k) & (8.5) \\ P_k^+ = (I - K_k H) P_k^- (I - K_k H)^T + K_k R K_k^T & (8.6) \end{cases}$$

The filter estimates simultaneously the angular rate and the biases of the sensors. Further more, the state estimation equation can be compacted using only the *a posteriori* term.

$$\hat{\mathbf{x}}_k^+ = (I - K_k H)\hat{\mathbf{x}}_{k-1}^+ + K_k \mathbf{z}_k \quad (8.7)$$

8.2 Simulation example

The filter equations from the previous section have been implemented as digital computations, and the numerical results of the gain matrices K_k and the covariance P_k are here analyzed in the following.

Consider an example with three sensors. The system state vector will have size four. The simulation parameters are otherwise chosen to reflect the 100Hz signal analyzed from the gyro-board: the sigma values of the noises w and v are taken to match the identified ARW and RRW coefficients ARW_c and RRW_c. Correlations $\mathbf{w}$ are chosen randomly, and measurement noises $\mathbf{v}$ are independent. The measurement time is one hour, equivalent to 360,000 points. The uncertainty noise was taken $\sigma_\omega = 10^3$. The covariance matrix Q_w is:

$$Q_w = \begin{bmatrix} 0.0029 & 0 & 0 \\ 0 & 0.0029 & 0 \\ 0 & 0 & 0.0029 \end{bmatrix} \begin{bmatrix} 1.0000 & 0.2305 & -0.6411 \\ 0.2305 & 1.0000 & -0.8927 \\ -0.6411 & -0.8927 & 1.0000 \end{bmatrix} \begin{bmatrix} 0.0029 & 0 & 0 \\ 0 & 0.0029 & 0 \\ 0 & 0 & 0.0029 \end{bmatrix} \tag{8.8}$$

8.3 Divergence of the covariance

One of the important properties of the Kalman filter is that the performance of the estimate can be evaluated in the design phase, by analyzing the evolution of the covariance matrix P_k [10], as this is independent of the actual measurement data. The diagonal elements of the covariance matrix are the variances corresponding to each element in the state vector.

When the state-space representation of the system is observable, or fulfilling slightly weaker conditions including detectability, the covariance sequence converges over time, and the performance of the filter is guaranteed by the fact that the variance of the estimated will be bounded in the long time horizon. However, the multi-MEMS model is neither observable nor detectable.

It is shown here with an example, that the covariance matrix P_k diverges and it is unbounded. The figure below plots the square root of the diagonal elements of the covariance, as functions of simulation time.

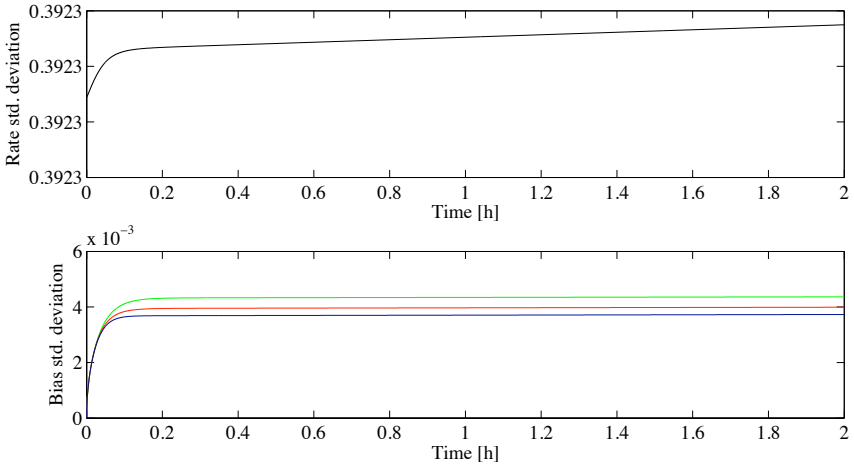

Figure 8.1: Divergence of the covariance elements, $\sqrt{p_{ii}}$

In this case the divergence of the covariance P is very slow, but nevertheless, in practice, the recursive Kalman filter cannot be reliably used without further precautions since the covariance calculation will eventually overflow or would be corrupted by small numerical imprecision on another hardware.

8.4 Convergence of the gain

In this section the gain is investigated and it can be seen that the matrix sequence K_k is converging, despite the divergence of the covariance sequence P_k. The convergence is rigorously proved in [4] and it is due term cancellations in the expression (4.33). In the figure below, each matrix element is plotted as a function of the iteration step k.

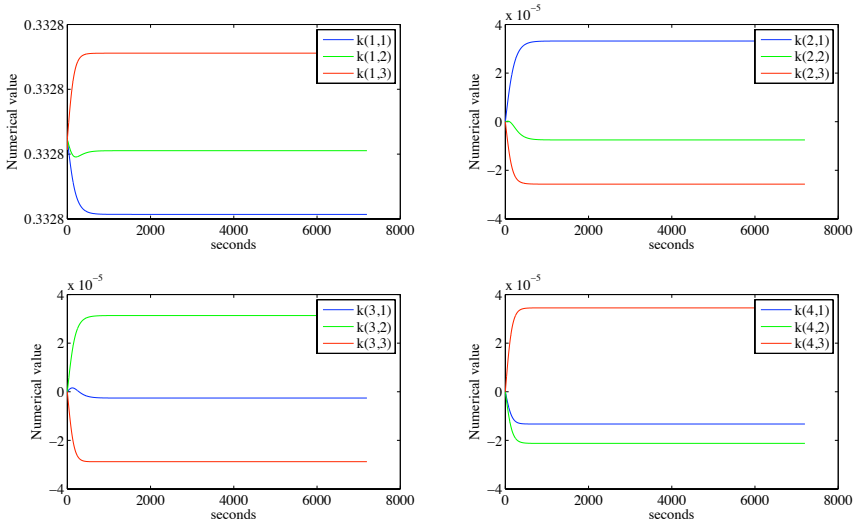

Figure 8.2: Convergence of the gain elements k_{ij}

Bayard and Ploen [4] shows that K converges to a bounded value asymptotically, in an infinite number of steps. The converged recursive gain can be said to be reached numerically when in the difference matrix of two consecutive K_k the maximum element in absolute value becomes close to the smallest floating point number of the system. The values of the converged recursive matrix K in this example are as follows:

$$K = \begin{bmatrix} 0.332815351545374 & 0.332820262807916 & 0.332827785499759 \\ 3.31879037392269e-005 & -7.49804087920127e-006 & -2.56898612103661e-005 \\ -2.58677834142207e-006 & 3.13564393958052e-005 & -2.87696601089407e-005 \\ -1.32559067993142e-005 & -2.12469682356657e-005 & 3.45028735409536e-005 \end{bmatrix} \tag{8.9}$$

To investigate some properties of the K elements, the equation (8.7) will be expanded to the form:

$$\begin{bmatrix} \hat{\omega}_k \\ \hat{b}_{1,k} \\ \hat{b}_{2,k} \\ \hat{b}_{3,k} \end{bmatrix} = \begin{pmatrix} 1-k_{11}-k_{12}-k_{13} & -k_{11} & -k_{12} & -k_{13} \\ -k_{21}-k_{22}-k_{23} & 1-k_{21} & -k_{22} & -k_{23} \\ -k_{31}-k_{32}-k_{33} & -k_{31} & 1-k_{32} & -k_{33} \\ -k_{41}-k_{42}-k_{43} & -k_{41} & -k_{42} & 1-k_{43} \end{pmatrix} \begin{bmatrix} \hat{\omega}_{k-1} & \hat{b}_{1,k-1} & \hat{b}_{2,k-1} & \hat{b}_{3,k-1} \end{bmatrix} + \\ + \begin{pmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \\ k_{41} & k_{42} & k_{43} \end{pmatrix} \begin{bmatrix} z_{1,k} & z_{2,k} & z_{3,k} \end{bmatrix} \tag{8.10}$$

Because ω_k is not dependent of ω_{k-1}, relation expressed by design parameter $q_\omega = \sigma_\omega >> 1$, the term $(1 - k_{11} - k_{12} - k_{13})$ should be very close to zero. Also, since the bias states are also independent of ω, the terms $(-k_{i1} - k_{i2} - k_{i3})$ will be zero. These relations check on any of the matrix gains in sequence K_k, not only on the stabilized form and show that the estimate practically does not use any dynamic information about the angular rate, and that q_ω was chosen sufficiently large.

If let to converge for a sufficiently long number of steps, the approximated gain is stable enough to be used in a filtering scheme, in the same way the constant Kalman gain can be used instead of the recursive iterations for a detectable system. The figure below shows the

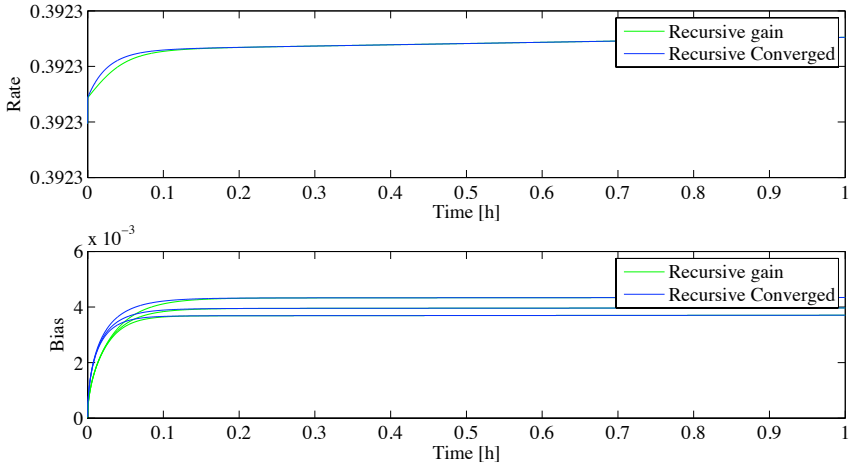

Figure 8.3: Diagonal elements of the covariance matrix, $\sqrt{p_{ii}}$

difference in the filter performance between the recursive gain filter and a constant gain filter running with the converged approximation.

Furthermore, it is expected that the Kalman gain matrix will converge for every situation discussed in the model variations section, 6.4 on page 54.

Chapter 9

Estimation: Bayard constant gain

Bayard method gives the exact close-form solution to the continuous Riccati differential equation of the multi-MEMS gyroscope system. It then shows that although the covariance matrix P of the state estimate diverges over time, the Kalman gain will converge to a steady state matrix. The method includes a discrete-time filtering scheme that uses a similarity transformation of the state-space system and a constant gain matrix.

Bayard and Ploen [4] and patent [5] are recent works and the original starting point for our thesis project. They analyze the multi-MEMS gyroscope problem using a continuous state space model, equivalent to the discrete model described in chapter 6, employ the Kalman filter as the natural optimization framework for this linear model, and rigorously study the Riccati equation that governs the evolution of the state covariance. This chapter describes the main points in the Bayard method, presents the filter implementation described in Bayard and Ploen [4], and a numerical example.

9.1 Concept and Framework

The multi-MEMS problem is first analyzed in a continuous-time framework in [4] and [5] to obtain analytical results. These results are then translated for the discrete-time problem and implementation. The continuous state-space model is equivalent with the discrete representation (6.4) used in this report.

$$\begin{cases} \dot{\mathbf{x}}(t) &= F\mathbf{x}(t) + G\mathbf{w}(t) \\ \mathbf{z}(t) &= H\mathbf{x}(t) + \mathbf{v}(t) \end{cases} \tag{9.1}$$

F is the dynamic matrix, H is the measurement matrix and G is the noise input matrix.

$$F = \mathbf{0}_{(N_s+1)\times(N_s+1)}, H = \begin{bmatrix} \mathbf{1}_{N_s} & \mathbf{I}_{N_s} \end{bmatrix}, G = \mathbf{I}_{N_s+1} \tag{9.2}$$

The Kalman-Bucy filter used for the estimation, and this filter is a direct equivalent of the discrete Kalman filter, presented in section 4.3, but for continuous time.

The dynamic equation of the filter is:

$$\dot{\hat{\mathbf{x}}}(t) = F\hat{\mathbf{x}}(t) + K(t)[z(t) - H\hat{\mathbf{x}(t)}] \Leftrightarrow \tag{9.3}$$

$$\Leftrightarrow \dot{\hat{\mathbf{x}}}(t) = [F + K(t)H]\hat{\mathbf{x}(t)} + K(t)z(t) \tag{9.4}$$

where the gain matrix has the following form:

$$K(t) = P(t)H^T R^{-1} \tag{9.5}$$

The optimum gain matrix $K(t)$ is a function of the state covariance $P(t)$ that, similar to the discrete Kalman filter, propagates through a dynamic equation.

$$\dot{P}(t) = FP(t) + P(t)F^T - P(t)HR^{-1}H^T P(t) + Q \tag{9.6}$$

This is the continuous Riccati differential equation. While this Riccati equations cannot be solved to obtain a closed form expression for P(t) in the general case, a main contribution of the Bayard method is that it finds the explicit solution of the particular multi-MEMS continuous Riccati differential equation:

$$\dot{P}(t) = -P(t)HR^{-1}H^T P(t) + Q \tag{9.7}$$

The solution P(t) is shown to be composed of 4 terms: a group of exponential elements that decay with time, two constant matrix elements that are algebraic expression of the model matrix parameters e.g. Q, R, H, and a term that is growing linearly with the time.

$$P(t) = C_1 t + C_0 + C_{01} + \mathcal{O}\left(te^{-c\cdot t}\right) \tag{9.8}$$

These are discussed in detail in [4], and are here addressed only in summary. We note here that the simulation results of the discrete recursive Kalman filter, figure 8.1, show both a time-linear and a decaying exponential trend in accordance with the analytical result of Bayard: the covariance grows unbounded in time, with a limiting linear behavior.

Subsequently, the optimum gain (9.5) can now be expressed in an explicit form by replacing the term $P(t)$ with the detailed expression of its solution, and [4] shows that by post multiplying $P(t)$ with $H^T R^{-1}$ the linear term and one of the constant parts disappear. Thus, the gain matrix $K(t)$ depends only on one constant term of the $P(t)$ solution and on the exponentially decaying terms.

$$K(t) = C_0 H^T R^{-1} + \mathcal{O}\left(te^{-c\cdot t}\right) \tag{9.9}$$

Because of the exponential terms $\mathcal{O}(\cdot)$ are decaying, $K(t)$ converges asymptotically to the value $C_0 H^T R^{-1}$. C_0 is a notation for a collective term composed by algebraic manipulations on the system matrices and will described in the implementation section below. The Bayard method continues by analyzing and designing a filter around the asymptotic gain K_∞, the constant component. An equivalent discrete gain $\bar{K}_\infty$ is also presented.

We note here that without the analytical solution for $P(t)$, it might not be possible to simply and precisely find K_∞. Numerical algorithms rely on having the covariance P(t) converge to a steady state to find the constant gain K, and in the multi-MEMS problem P(t) does not converge. For example MATLAB® function such as `kalman()` will return error and cannot be used to find the asymptotic K_∞ gain.

9.2 Properties of the filter

There are two important properties of the asymptotic filter.

The dynamic equation of the continuous filter, e.g. by particularizing (9.4) to the multi-MEMS problem, is:

$$\dot{\hat{\mathbf{x}}}(t) = K_\infty H\hat{\mathbf{x}}(t) + K(t)\mathbf{z}(t) \tag{9.10}$$

In [4], matrix $K_\infty H$ is shown to have one pole converging to $-\infty$ with the increase of the rate uncertainty noise element, q_ω. This is the typical case, to have q_ω as large as possible so

that the estimate is not using any dynamic information about the angular rate. Equivalently, for the discrete filter this pole is converging to 0 and it is easy to implement in a numerical algorithm. This effect and the existence of a 0 pole has also been seen in the recursive Kalman implementation, in section 8.4.

Secondly, the Bayard analysis shows that all the poles of the filter matrix $K_\infty H$ are stable, with the exception of a natural integrator that arises from the structure of the matrices. An integrator pole is 0 in continuous time and equivalent 1 for the discrete filter. This information is important, because it puts the system at the margin of stability.

It is already known from the study of the state covariance P that the error of the filter grows unbounded. Equivalently, by looking at the the error equation of the multi-MEMS discrete filter system with $\bar{K}_\infty$, as derived in (4.41),

$$\epsilon_{k+1} = (I - \bar{K}_\infty H)\epsilon_k + K_k H \mathbf{w}_k \tag{9.11}$$

it can again be seen that error state ϵ_{k+1} will not not converge to 0, but will accumulate with time due to the integrator state.

Furthermore, the discrete equation of the continuous filter as derived in (4.43),

$$\hat{\mathbf{x}}_k^+ = (I - \bar{K}_\infty H)\hat{\mathbf{x}}_{k-1}^+ + \bar{K}_\infty \mathbf{z}_k \tag{9.12}$$

can become incorrect as a result of the numerical roundoff errors, if for the computed matrix $(I - \bar{K}_\infty H)$ the integrator eigen value 1 is corrupted.

This problem is also solved in the Bayard and Ploen [4] implementation by using a system similarity transformation that organizes the $\bar{K}_\infty$ matrix in two diagonal blocks: the first block has size 1, and isolates the zero poles of the gain matrix. Numerical problems are avoid by explicitly overwriting this element to 0, so that any residual errors are corrected. The matrix $(I - \bar{K}_\infty H)$ will now have a pure integrator pole 1.

In conclusion, having an explicit analytical solution for the Riccati differential equation of the state covariance P(t) allows for a rigorous study of the Kalman filter behavior. The next section presents the step-by-step the Bayard method, a Kalman filter with a similarity transformation and using the asymptotic gain matrix.

9.3 Implementation

The filter equations are prepared for implementation by a similarity transformation that generally acts on a state space system as in (6.16). This particular transformation has the purpose of isolating the pure integrator term, and acts on the estimate state in the following way:

$$\xi(kT_s) = S^{-1}\hat{\mathbf{x}}(kTs) \tag{9.13}$$

The transformed filter system is given as:

$$\begin{cases} \hat{\xi}\left((k+1)T_s\right) & = \begin{bmatrix} e^{\left(-\bar{D}^{1/2}\right)T_s} & 0 \\ 0 & 1 \end{bmatrix} \hat{\xi}(kT_s) + S^{-1}\bar{K}_\infty \mathbf{z}(kT_s) \\ \hat{\omega}(kT_s) & = e_1^T S \hat{\xi}(kT_s) \end{cases} \tag{9.14}$$

where T_s is the sample period, and $e_1^T = \begin{bmatrix} 1 & 0 & \dots & 0 \end{bmatrix}$ a base vector that extract the first element of the state vector, the angular rate.

The only step left to finish the implementation is to define the matrices $\bar{D}$, S and $\bar{K}_\infty$. This matrices arise from the solution of the Riccati Differential equation. We present next the algebraic manipulations needed to construct these matrices, as described in [4] and avoiding a typographic error in [5], without addressing the derivation of these relations.

The computation procedure for the $\bar{D}$, S and $\bar{K}_\infty$ matrices are given as follows:

1. The symmetric matrix L is constructed:

$$L = Q^{1/2} H^T R^{-1} H Q^{1/2} \tag{9.15}$$

2. The eigen value decomposition of symmetric matrix L yields matrices A and D:

$$L = A D A^T \tag{9.16}$$

where A is the orthogonal matrix and D is the diagonal matrix, and the order is such that the zero eigen value is last.

3. The $\bar{A}$ and $\bar{D}$ matrices are obtained below:

$$D = \begin{bmatrix} \bar{D} & 0 \\ 0 & 0 \end{bmatrix} \tag{9.17}$$

$$A = \begin{bmatrix} \bar{A} & | & \nu \end{bmatrix} \tag{9.18}$$

4. The auxiliary matrix S has the form:

$$S = Q^{1/2} A \tag{9.19}$$

where notation $^{1/2}$ represents square root of matrix, i.e. $Q = Q^{1/2} Q^{1/2}$. Because matrix Q is positive definite matrix, it has exactly one positive square root.

5. The Kalman gain $\bar{K}_\infty$ and the transformed gain $S^{-1}\bar{K}_\infty$ used in the transformed system (9.14):

$$\begin{aligned} \bar{K}_\infty &= S \begin{bmatrix} \bar{D}^{-1} \left(I - e^{-\bar{D}^{1/2} T_s} \right) & 0 \\ 0 & 0 \end{bmatrix} S^T H^T R^{-1} = \\ &= Q^{1/2} \bar{A} \bar{D}^{-1} \left(I - e^{-\bar{D}^{1/2} T_s} \right) \bar{A}^T Q^{1/2} H^T R^{-1} \end{aligned} \tag{9.20}$$

$$S^{-1} K = \begin{bmatrix} \bar{D}^{-1} \left(I - e^{-\bar{D}^{1/2} T_s} \right) & 0 \\ 0 & 0 \end{bmatrix} S^T C^T R^{-1} \tag{9.21}$$

6. It is important that the last row of the matrix $S^{-1}K$, corresponding to the pure integrator state, is set exactly to zero.

A simple MATLAB script that implements these algebraic computations is shown here. We note that this particular implementation is not especially designed to be numerically stable, for example for larger values of the q_ω parameter, the eigen value decomposition of matrix L becomes ill-conditioned.

```
%# Computes the constant Kalman gain using Bayard method
function [ S, Sm1K, Dbar12  ] = bayard_gain(Q, R, H, Ts)

Q12=sqrtm(Q);
L=(Q12*H'/R)*H*Q12;

n=size(R,1);

%# Sort eigenvalues in descending order
[A,D]=eig(L);
[D order] = sort(diag(D),'descend');
D=diag(D); A = A(:,order);

Dbar = D(1:n,1:n);

%# Not used further
%Abar = A(:,1:Ns);
```

```
%v = A(:,Ns+1);

S = Q12*A;
Dbar12=sqrtm(Dbar);

Sm1K=blkdiag(Dbar\(eye(n)-expm(-(Dbar12*Ts))),0)*S'*H'/R;
Sm1K(n+1,:)=zeros(1,n);
```

9.4 Numerical example

The same case of 3 sensors described in section 8.2 is taken as example next, to show the numerical structure of the discussed matrices.

$$S^{-1}\bar{K}_\infty = \begin{bmatrix} -0.000333333333330811 & -0.000333333333330576 & -0.000333333333329761 \\ 0.0048526992069739 & 0.00708486821303693 & -0.0119375674200434 \\ -0.0109827455892444 & 0.00969398610657207 & 0.00128875948268326 \\ 000 & & \end{bmatrix} \tag{9.22}$$

$$\bar{D}^{1/2} = \begin{bmatrix} 2547.13354054293 & 0 & 0 \\ 0 & 0.00620224995350911 & 0 \\ 0 & 0 & 0.00370273256444844 \end{bmatrix} \tag{9.23}$$

S is a full matrix, omitted for brevity. To directly compare the gain of the Bayard method with the converged recursive gain result (8.9), matrix $SS^{-1}\bar{K}_\infty$ is shown below, although this computation is not normally performed in the Bayard transformed filter.

$$SS^{-1}\bar{K}_\infty = \bar{K}_\infty = \begin{bmatrix} 0.333327551591338 & 0.333332462853876 & 0.33333998554575 \\ 3.31879037433893e-005 & -7.49804087759995e-006 & -2.56898612135909e-005 \\ -2.58677833992e-006 & 3.13564394003242e-005 & -2.87696601135067e-005 \\ -1.3255906802376e-005 & -2.12469682399718e-005 & 3.45028735460221e-005 \end{bmatrix} \tag{9.24}$$

The matrices (8.9) and (9.24) are nearly identical, the most visible differences are on the first line elements.

The snippet below shows how the two matrices compare with respect to the relations discussed in section 8.4: the terms $(1 - k_{11} - k_{12} - k_{13})$ and $(-k_{i1} - k_{i2} - k_{i3})$ should be very close to 0. The recursive gain seems slightly worse in this case.

```
sum(Kb(1,:)) = 0.999999999990964
sum(K(1,:))  = 0.998463399853048


sum(Kb(2,:)) = 1.65219848910557e-012      sum(Kb(3,:)) = 9.46897515095597e-013
 sum(K(2,:)) = 1.64965959254947e-012       sum(K(3,:)) = 9.45442420912431e-013

 sum(Kb(4,:)) = -1.49632568027759e-012
 sum(K(4,:))  = -1.49402627720513e-012
```

The integrator pole, equivalent to a pole 0 in the matrix KH, is also less corrupted in the Bayard gain but the difference is extremely small.

```
      eig(Kb*H)                    eig(K*H)
 0.999999999991332             0.998463399853415
 6.20205761802106e-005         6.20205761697209e-005
 3.70266401419350e-005         3.70266401392396e-005
-1.26357778835430e-021 <--0--> 2.84738385085346e-021
```

We plot next also the performance given by the state covariance obtained by iterating the Riccati differential equation with the two constant gains. The plot is of the square root

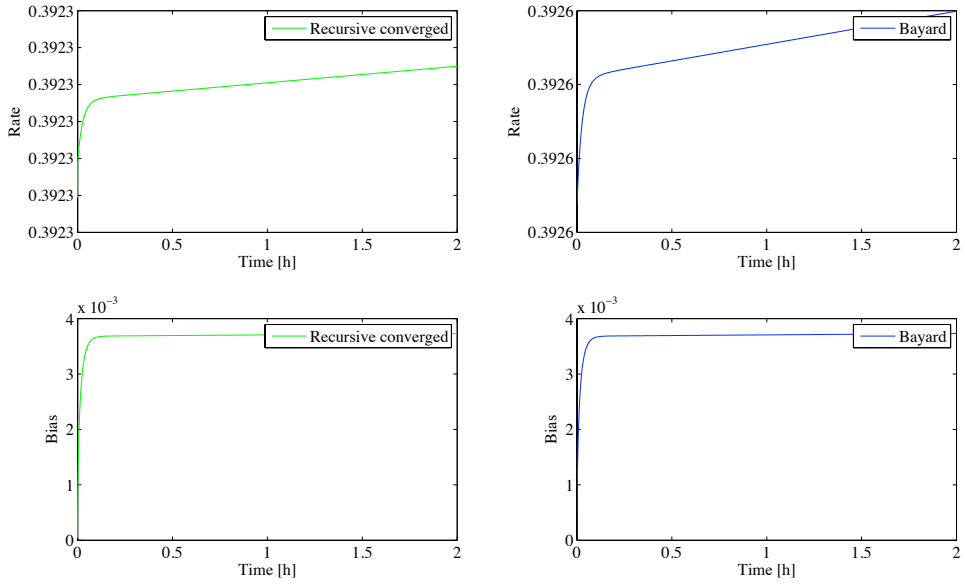

Figure 9.1: Performance of the two Kalman gains

diagonal elements of the covariance, representing the standard deviation of the rate and bias estimates.

It can be seen that differences are so small that they could be caused purely by numerical imprecisions of the digital calculations. We can conclude the performances of the Bayard gain and that of the converged recursive gain are identical for all practical purposes.

Chapter 10

Estimation: Comparative results

In this section the results of the estimation algorithms are presented along with evaluation of the results. Sensor data is generated according to the state space model and the three estimation methods, signal averaging, Kalman filter with the approximated gain and Kalman filter with the Bayard gain. All algorithms are run as online algorithms where perfect parameter identification is assumed.

The following simulations use an eight gyroscope configuration with parameters matching the gyro-board. The standard deviation for the noises w and v are inferred from the identified noise coefficients ARW_c and RRW_c and the sampling frequency is set to 100Hz. In the following, the gyroscopes are considered absolutely identical in noise levels and the uncertainty noise is set to $\sigma_\omega = 10^3$. The measurement noises $\mathbf{v}$ independent and thus the R covariance matrix is diagonal.

If present, correlations in the $\mathbf{v}$ noise will also contribute to the overall performance improvement of the final estimate, however they they are not considered for two reasons. The main idea, investigated in the test is that estimation of the biases can be improved by using $\mathbf{w}$-noise correlation information. The $\mathbf{v}$ correlations bring no information on the bias states. Secondly, if known and present, the signal averaging technique can also easily take into account the v-correlations and become optimized by using weights in a manner similar to the procedure described in section 1.3 on page 5. As such, the "fair" comparison would be between a Kalman technique and a weighted average scheme, and new formulas would need to be derived in chapter 7 on page 61. By keeping noises v independent, the comparison of Kalman and averaging techniques remains relevant.

Within this general set-up of the example, different $\mathbf{w}$-correlations are considered. The project CD contains a MATLAB® package called `multi-gyro-sim` that allow the results and graphics presented next to be recreated. The package can also be used to test other test scenarios than the ones considered here.

10.1 Uncorrelated w-noises

When the $\mathbf{w}$-noises are uncorrelated, it is expected that averaging is the optimum estimator as no other information can be used to improve the bias estimate. This is confirmed by simulations, and shown below with an example. The file 1 in the `multi-gyro-sim` MATLAB® file corresponds to the uncorrelated sensors case.

Eight gyroscope signals are generated according to the state-space model: a true angular rate changing over time measured through sensors that contain a slowly changing bias component and white noise. The figures 10.1 and 10.2 show a realization of random signals with the given parameters, and the corresponding angle drift for each sensor taken independently. The

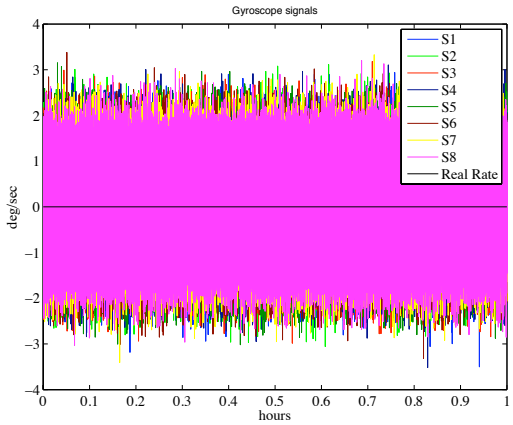

Figure 10.1: Simulated rate signals

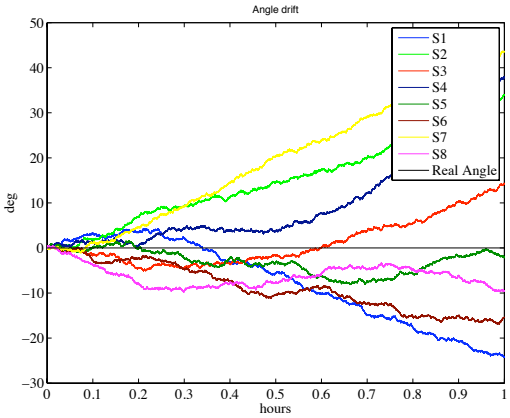

Figure 10.2: Angle drift for each sensor

rate and angle error of the estimation can be seen in 10.3, and they are identical between the three techniques: signal averaging, Kalman filter with the approximated constant gain, and with the Bayard gain. It is noted here that the actual value of the angle drift depends from

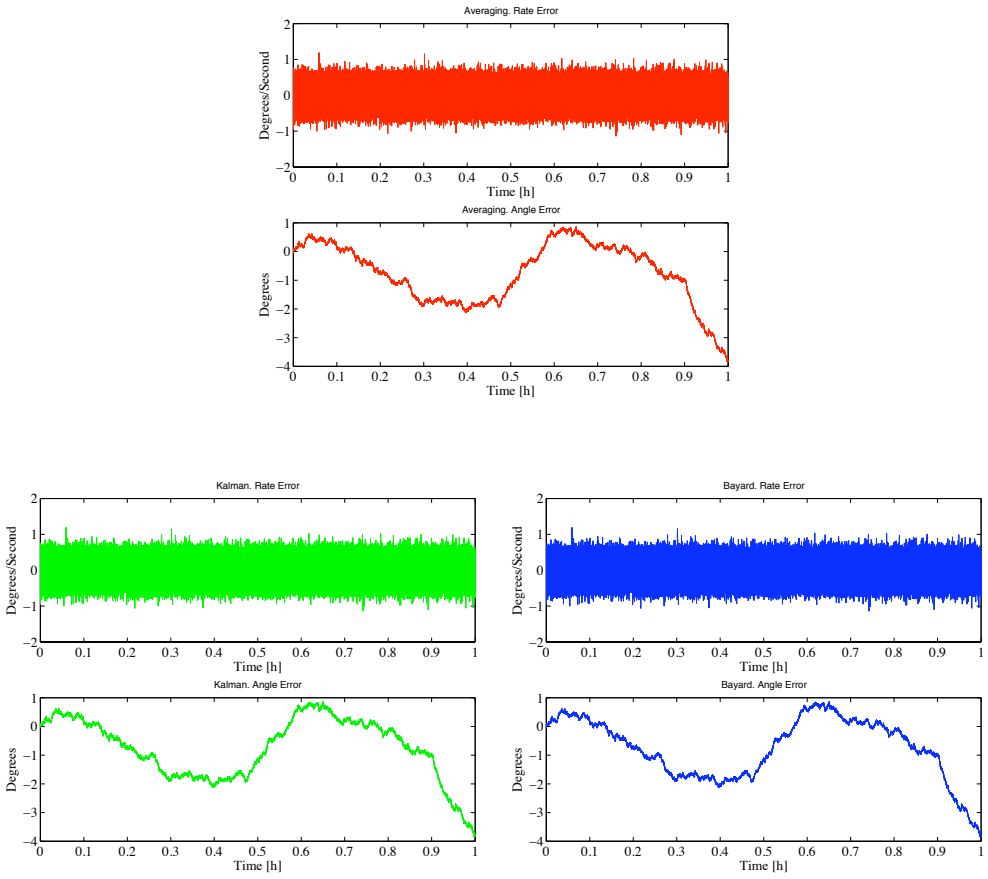

Figure 10.3: Rate and angle errors of Averaging, Kalman with approximated gain and with Bayard gain estimations

run-to-run in the simulations, and figure 10.3 is only one possible realization. This is due to the random nature of the noises. The deterministic measure of performance is the standard deviation, or equivalently the average absolute drift of the estimated signal in a period of time, for example in an hour. Figure 10.4 plots 100 simulations and it is possible to see the overall statistical performance of the estimation for uncorrelated sensors. Further more, it is

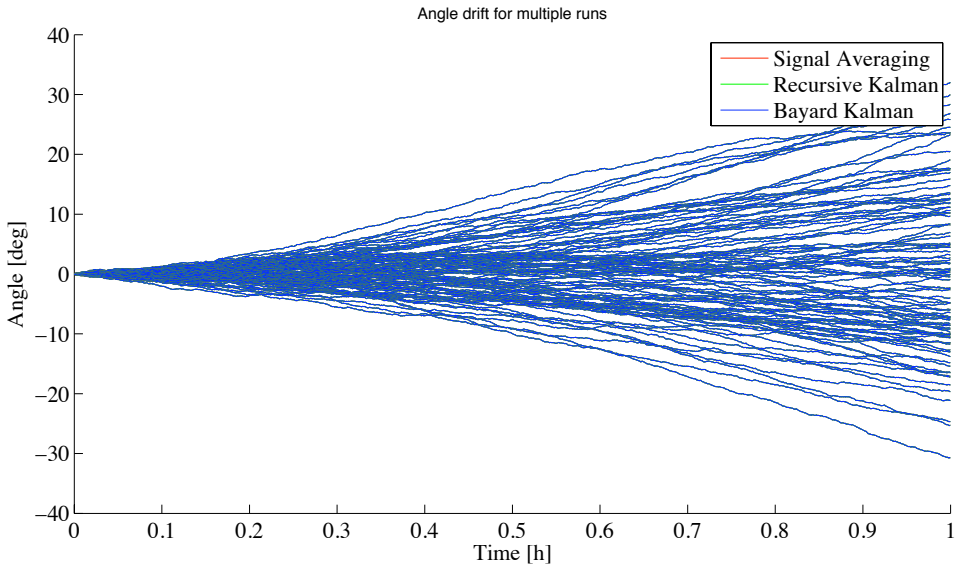

Figure 10.4: Multiple runs

not only the angle estimation variance that is increasing over time. The variance of the rate estimate is also increasing, although this is not noticeable in the one hour time horizon. The rate estimate is increasing due to the random walk noise component present in each of the rate signals. This effect is very small and is not noticeable in the graphics above because the RRW coefficient is small.

For signal averaging, both the variance of the rate estimate and that of the angle estimate as functions of time have been calculated in chapter 7. For the uncorrelated sensors case, the rate variance can be computed with specific formula (7.10), but also with the more general (7.13). The variance of the angle build by integrating the rate estimation can be calculated using (7.5). Figure 10.5 shows the performance of the rate and figure 10.6 that of the angle estimate. It can be seen that if the gyroscopes of the simulation are uncorrelated, the angle drift over an hour will be in average 13°.

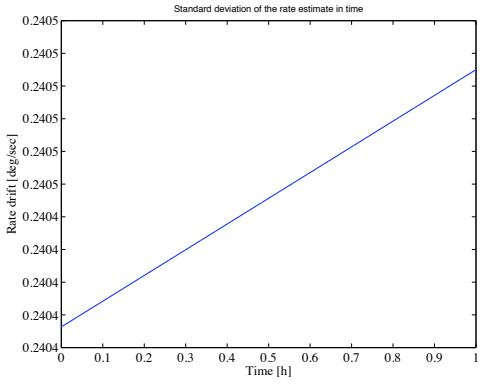

Figure 10.5: Rate drift

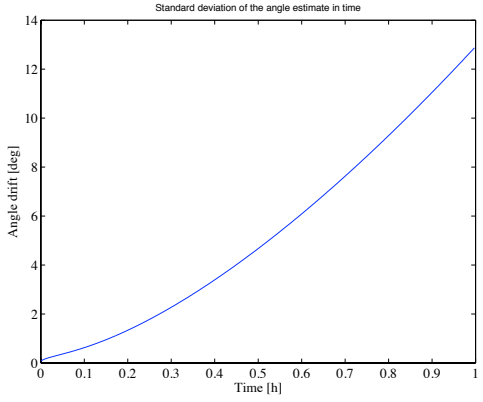

Figure 10.6: Angle drift

In conclusion, both the rate and angle estimation error increase over time. The multi-sensor

problem with biases will always have drifting estimates. In the following cases, it will be seen that it is sometimes possible reduce the size of the drift below that of the averaging technique.

10.2 Identical correlation

When the w-noises are identically correlated with each other, the correlation matrix Q has the form described in 6.36 on page 58. The signal averaging is again the best possible estimate in this situation, and a simplified argument for this is that when the correlations coefficients are identical the best way to combine signals is by identical weights.

10.2.1 Negative correlations

When ρ is negative, the overall estimate is improved as the negative correlations make it possible to partially distinguish noise components since they are picked up with different sign by the gyroscopes from the actual rate which affects the sensors with the same sign.

Consider next a simulation with $\rho = -0.1$ which corresponds to the saved parameter file "2". For clarification, in this case, matrix Q has the following form:

$$Q = \mathrm{diag}_8\big(0.0029\big) \begin{bmatrix} 1 & -0.1 & -0.1 & -0.1 & -0.1 & -0.1 & -0.1 & -0.1 \\ -0.1 & 1 & -0.1 & -0.1 & -0.1 & -0.1 & -0.1 & -0.1 \\ -0.1 & -0.1 & 1 & -0.1 & -0.1 & -0.1 & -0.1 & -0.1 \\ -0.1 & -0.1 & -0.1 & 1 & -0.1 & -0.1 & -0.1 & -0.1 \\ -0.1 & -0.1 & -0.1 & -0.1 & 1 & -0.1 & -0.1 & -0.1 \\ -0.1 & -0.1 & -0.1 & -0.1 & -0.1 & 1 & -0.1 & -0.1 \\ -0.1 & -0.1 & -0.1 & -0.1 & -0.1 & -0.1 & 1 & -0.1 \\ -0.1 & -0.1 & -0.1 & -0.1 & -0.1 & -0.1 & -0.1 & 1 \end{bmatrix} \mathrm{diag}_8\big(0.0029\big) \qquad (10.1)$$

The error plot 10.7 from a single run simulation supports the fact that all three techniques give the same results.

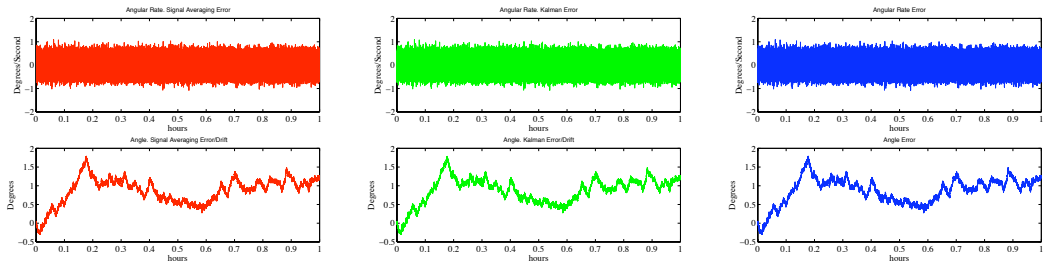

Figure 10.7: Rate and angle errors of Averaging, Kalman with approximated gain and with Bayard gain estimations

The multiple simulations plot 10.8 gives a statistical overview about the drift of the estimated angle. It can be seen that the angle drift is reduced.

Equations (7.13) and (7.5) from chapter 7 are used again to plot the average drift of the rate and of the angle estimates.

The average angle drift is little over 7° for a time horizon of one hour, less than that of uncorrelated sensors.

Taking an extreme case, with the negative correlation factor close to its minimum limiting value of $-\frac{1}{N_s-1}$, $\rho = -0.1428571$, the average angle drift is improved to below 1.5° in one hour.

Furthermore, because the angle drift has been calculated to closed form expression in (7.5), it is fast and easy to examine the drift also for longer time horizons. The angle drift in 10

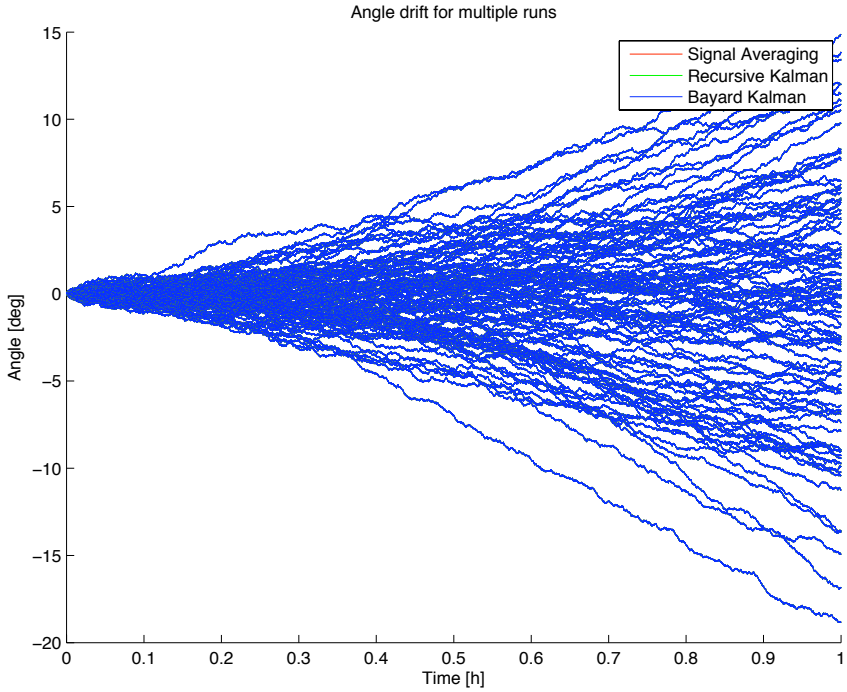

Figure 10.8: Multiple runs, ρ=-0.1

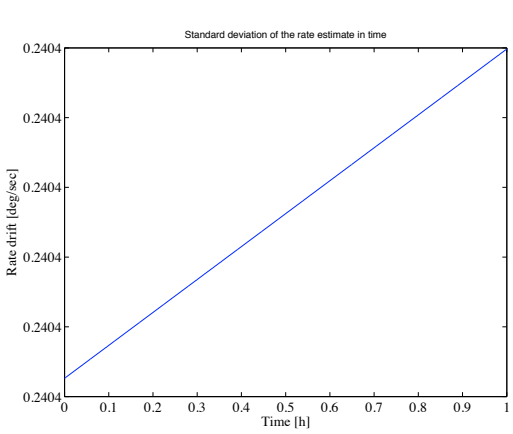

Figure 10.9: Rate drift

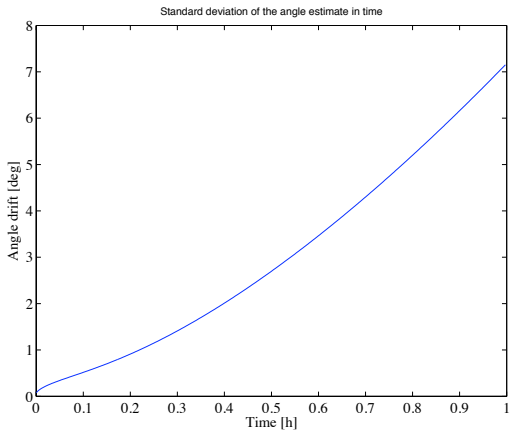

Figure 10.10: Angle drift

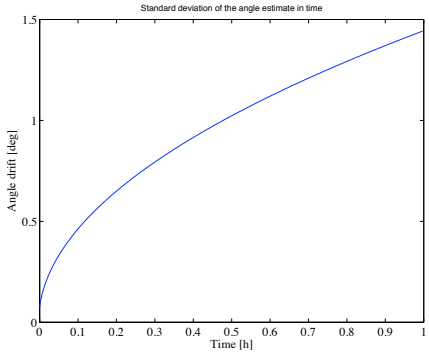

Figure 10.11: Drift of the angle estimation

hours for $\rho = -0.1428571$ is 4.57°. It is hard to imagine in practice however an array of multiple perfectly identical sensors, with a precise and equal extreme correlation factor of -0.1428571.

10.2.2 Positive correlations

When ρ is positive, the overall estimate deprecates as positive correlations decrease the possibility of identifying noises. When a noise is affecting positive correlated sensors, all of these react with the same signal tendency, similar to the action of a real rate.

Consider next an example with correlation factor $\rho = 0.3$ equivalent to the saved parameter set "2bis".

The error plot 10.12 from a single run simulation supports the fact that all three techniques give the same results. The average angle drift plot shows a value of approximately 22.5° in an one hour time horizon, more than in the case of uncorrelated sensors.

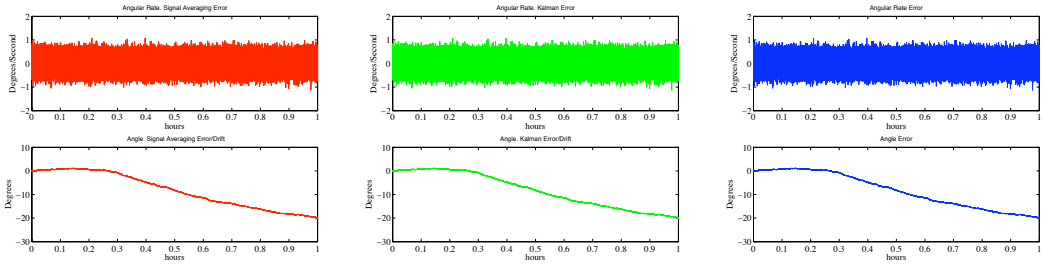

Figure 10.12: Rate and angle errors of Averaging, Kalman with approximated gain and with Bayard gain estimations

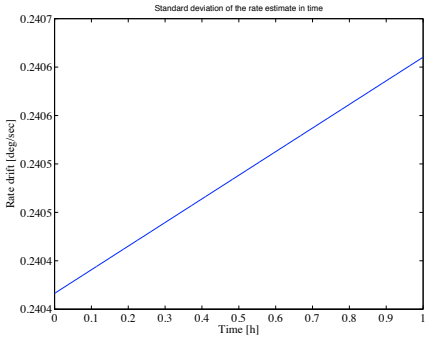

Figure 10.13: Rate drift

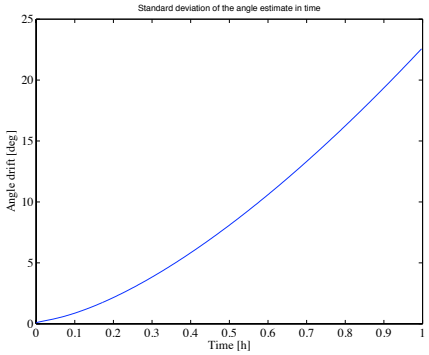

Figure 10.14: Angle drift

10.3 Partial results

For both uncorrelated sensors and equal correlations with factor ρ, averaging is the optimum estimation technique.

The averaged rate signal is similar in content with that of a single sensor, it is composed of a bias component that is drifting and a white noise. These noises are however reduced by the averaging operation. The bias component drive noise is very small, such that the bias drifts slowly. For example the drift is not visible in the one hour time plots 10.7 or 10.3 because the white noise is the dominant component by two orders of magnitude. The rate drift can only be seen in figures such as 10.5.

The table below shows the variance of the rate signal after one hour for the different cases discussed above. The average rate drift for a single sensor is 0.4627°, and was computed using (7.5).

It can be seen that the negative **w** correlations do not bring direct significant improvement in the rate estimate signal, because the dominant component is the **v** noise and the bias is too small comparative to appear.

Rate noise after 1hour	Single sensor	8 sensors
Uncorrelated	$0.4627[^\circ/s]$	$0.2405\ [^\circ/s]$
$\rho = -0.1$	-	$0.2405\ [^\circ/s]$
$\rho = -0.1428571$	-	$0.2404\ [^\circ/s]$
$\rho = 0.3$	-	$0.2407\ [^\circ/s]$

Table 10.1: Noise level of the estimated rate, standard deviation after one hour

The angle estimate is computed by integrating the rate estimate. As such, it contains two noise components: the accumulated bias $\sum_1^N \mathbf{b}_k$, an integrated random walk, and the accumulated white noise $\sum_1^N \mathbf{v}_k$, a random walk. In the angle signal, because of the double integration action, the accumulated bias becomes the predominant contributor in the error. Thus the relatively small improvements in the rate bias estimate obtained from the negative correlation of noise **w** become increasingly significant at the angle signal level. And opposite, positive correlation will increase the errors compared to the uncorrelated/independent case.

The table below summarizes the average angle drift for one hour obtained in the different simulation cases presented above. The average angle drift for a single sensor is 36.4°, and was computed using (7.15).

Angle drift, 1h	Single sensor	8 sensors
Uncorrelated	36.4°	12.87°
$\rho = -0.1$	-	7.15°
$\rho = -0.1428571$	-	1.44°
$\rho = 0.3$	-	22.56°

Table 10.2: Summary of average angle drift, [°], for a one hour simulation

10.4 Random correlations

The advantage of using the Kalman filter becomes apparent in the case of **w**-noise delta correlations with an unequal pattern. Different patterns of correlations could be investigated, and in Bayard and Ploen [4] the subject is addressed to greater lengths. In the following we only address a case of arbitrarily chosen correlations (parameter sets 3 and 5 in the saved parameter folder). Arbitrary correlations have been created starting with a random matrix of uniform distribution, as described at the end of section 6.6 on page 58.

The Kalman filter is able to use different negative correlations to improve the rate estimation, and this translates into improvement of the angle estimate calculated by integrating the rate. While a sensor can be added to an existing array because it has a negative correlation with a particular unit, it could also introduce positive correlations with another units. This deteriorates the performance of the averaging, which relies on the overall sum of the correlation as it can be seen in (7.13) and (7.16). The Kalman filter however creates a weighting scheme when it combines the information from the new measurement with the previous state bias estimation, making it possible to fully use the benefits of negative correlations while limiting the effect of positive correlations. This is illustrated in the following through simulation.

The two arbitrarily matrices are show below for clarity, as the exact results will vary depending on the Q matrix.

$$Q_3 = \mathrm{diag}_8\left(0.0029\right) \begin{bmatrix} 1.0000 & 0.1632 & 0.4347 & 0.3839 & -0.4230 & 0.0966 & -0.2550 & -0.2585 \\ 0.1632 & 1.0000 & 0.3964 & 0.1554 & -0.0921 & 0.2230 & -0.6627 & -0.3817 \\ 0.4347 & 0.3964 & 1.0000 & -0.3017 & -0.6241 & -0.2330 & -0.2450 & -0.6012 \\ 0.3839 & 0.1554 & -0.3017 & 1.0000 & 0.0564 & 0.3664 & -0.3457 & -0.0694 \\ -0.4230 & -0.0921 & -0.6241 & 0.0564 & 1.0000 & 0.4264 & -0.3485 & 0.4517 \\ 0.0966 & 0.2230 & -0.2330 & 0.3664 & 0.4264 & 1.0000 & -0.3990 & -0.2575 \\ -0.2550 & -0.6627 & -0.2450 & -0.3457 & -0.3485 & -0.3990 & 1.0000 & 0.3837 \\ -0.2585 & -0.3817 & -0.6012 & -0.0694 & 0.4517 & -0.2575 & 0.3837 & 1.0000 \end{bmatrix} \mathrm{diag}_8\left(0.0029\right) \tag{10.2}$$

$$Q_5 = \mathrm{diag}_8\left(0.0029\right) \begin{bmatrix} 1.0000 & -0.5325 & -0.0919 & 0.4838 & 0.0233 & -0.2234 & -0.1018 & -0.4034 \\ -0.5325 & 1.0000 & 0.6489 & -0.7680 & 0.0383 & 0.4067 & 0.2381 & -0.4280 \\ -0.0919 & 0.6489 & 1.0000 & -0.2542 & 0.4135 & 0.1250 & 0.5287 & -0.3604 \\ 0.4838 & -0.7680 & -0.2542 & 1.0000 & 0.5041 & -0.5885 & 0.2546 & 0.2022 \\ 0.0233 & 0.0383 & 0.4135 & 0.5041 & 1.0000 & -0.2143 & 0.6344 & 0.0025 \\ -0.2234 & 0.4067 & 0.1250 & -0.5885 & -0.2143 & 1.0000 & 0.2515 & -0.3377 \\ -0.1018 & 0.2381 & 0.5287 & 0.2546 & 0.6344 & 0.2515 & 1.0000 & -0.2807 \\ -0.4034 & -0.4280 & -0.3604 & 0.2022 & 0.0025 & -0.3377 & -0.2807 & 1.0000 \end{bmatrix} \mathrm{diag}_8\left(0.0029\right) \tag{10.3}$$

The following multiple runs plots, showing an estimation time horizon of one hour, are illustrative. It can be seen that the averaging angle estimates, the red lines, are more spread while the Kalman estimates, both the approximated gain and Bayard gain, keep a more constrained and focused drift.

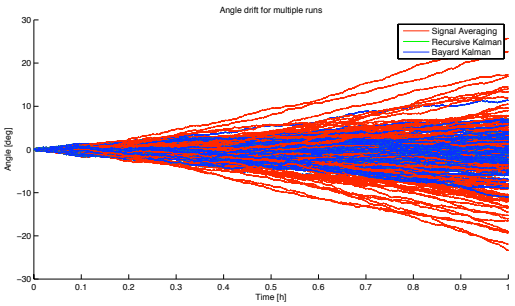

Figure 10.15: Multiple runs, Q_3, 1h

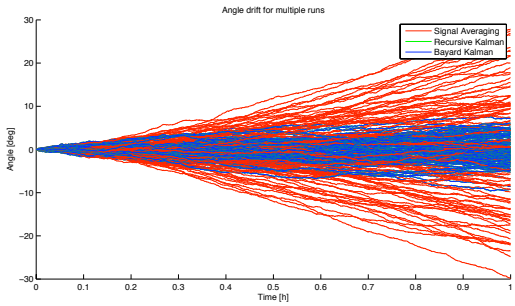

Figure 10.16: Multiple runs, Q_5, 1h

A two hours multiple-runs plot shows even better how the angle estimation drift is reduced compared to averaging.

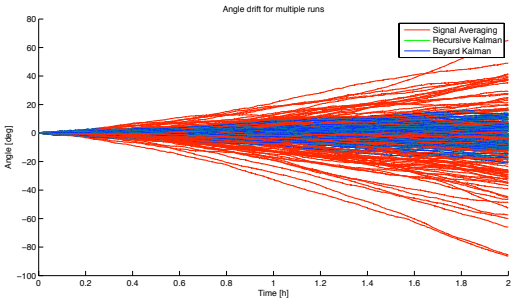

Figure 10.17: Multiple runs, Q_3, two hour

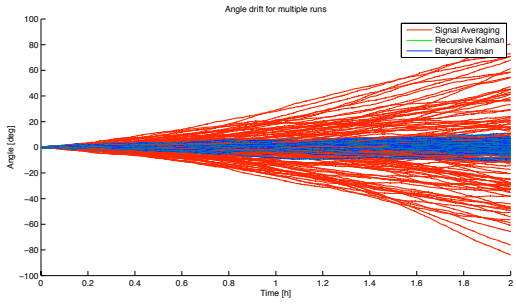

Figure 10.18: Multiple runs, Q_5, two hour

A table with the average and Kalman angle drifts is presented below. The values are computed as the standard deviation of the 100 runs, and also explicitly in the case of averaging. For the Kalman estimation however, a formula has not been developed. The aspects of deriving a formula for the Kalman performance is further addressed in chapter 13 on page 95. A formula would be particularly useful for an accurate evaluation of the average drift over longer periods of time, without having to run multiple computation heavy simulations.

In conclusion, negative correlations in the **w** noise are better used in a Kalman filter scheme. The improvement is visible in the angle estimation, while the rate estimate only benefits of the averaging effect of using multiple sensors.

It is mentioned here that negative **v** correlations can further improve the angle estimate, and also the rate estimate. This result is more obvious, and related to the simple example

	Averaging (sim)	Averaging (eq)	Kalman (sim)	Kalman (eq)
Q_3, 1 hour	9.66	9.24	4.23	?
Q_3, 2 hour	29.17	25.91	8.03	?
Q_5, 1 hour	13.88	13.14	3.23	?
Q_5, 1 hour	37.32	36.98	4.38	?

Table 10.3: Summary of average angle drift, [°] for random correlations

in 1.3, and not discussed at length. Secondary improvement can also be obtained if small variations in the noise parameters of the unit sensors in the configuration, different σ_w and σ_v coefficients values for each sensor, can be reliably identified. Both **v** correlations and the noise variations can be easily and directly introduced the Kalman filter, via matrices R and Q.

10.5 Approximated gain vs. Bayard gain

In all numerical examples considered in this project, the performance of the approximate converged recursive gain was found to be virtually identical with that of the Bayard gain. An example for 3 sensors was already presented in 9.4 on page 73. A second performance plot for the eight sensor case with the arbitrary correlation matrix Q_3 with is shown in the figure below.

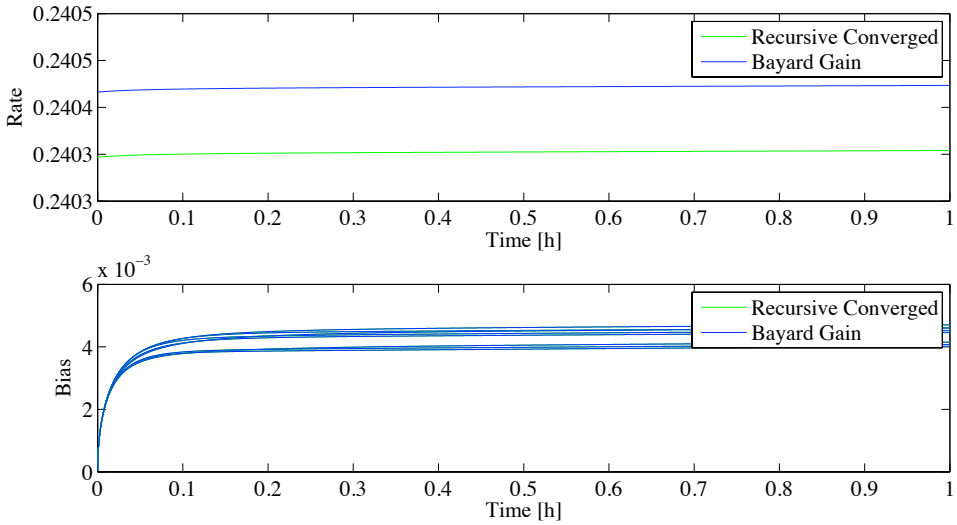

Figure 10.19: Performance of the two Kalman gains

Chapter Conclusion

The estimation methods, both simple averaging and different versions of Kalman filters, have shown that, when present, negative w-noise correlation can be used to improve the performance of the multi-MEMS array. The **w**-noise correlations can be used to better estimate each sensor bias value, and by doing so also improving the angular rate estimate.

Using **w**-noise correlations, it is mainly the angle output of the gyroscope array that has a significant gain in performance, not the rate signal.

While we focused exclusively on the **w**-correlations, other possibilities to improve the performance of the gyro-board have not been addressed at length. The **v**-noise correlations can also be used and the will also improve the rate signal of the gyroscope array and they can be directly included in the Kalman optimization scheme through matrix R, and additional information about the dynamic of the property to be measured can be included in the state-space model, making it more particular. For example, dynamic smoothing similar to low-pass filtering can be added by imposing a smaller dynamic range for the state when this is known.

Chapter 11

Identification of Noise Correlations

In the previous chapter methods of combining measurements from multiple homogeneous MEMS gyroscopes have been presented. It has been shown that performance improvement possible through Kalman filtering, provided that the gyroscope noise processes are favorably correlated. The existence of such favorable correlations is thus the the foundation for using Kalman filtering in relation to drift reduction as presented in the previous chapters. In this chapter the EM algorithm, as presented in section 4.5 on page 34 is applied to data captured from the gyroscope in an effort to investigate the existence of such correlations.

The chapter consists of two main sections, the first being a presentation of the framework under which the identification algorithm is implemented. The main underlying assumption is, that by specifying the operational conditions under which the data is acquired, the system model can be reduced dramatically. The second section address the feasibility of the developed identification method, by applying it to simulated data, which contain known noise correlations.

11.1 Modified System model

In this section the modified system model and data acquisition process is addressed in detail. The main objective of this section is to present the framework under which the noise correlations are identified.

In section 6.1 on page 49 the model used in the process of estimating the angular rate ω of the system has been presented. The main objective of this model is to relate the angular rate ω to the gyroscope bias b. In this section a modified model, that exclude the angular rate ω and only consider the sensor bias is presented. The reasoning behind this modification is the circumstances under which the data used for the correlation identification is acquired.

Consider the gyro board project containing N_s homogeneous gyroscopes lying motionless on a stable base. The gyroscopes are subjected only to the rotational input caused by the rotation of the earth. To this extent the system can be assumed to be motionless. Considering that the system is subjected to no input, the inclusion of the angular rate ω in the model is not relevant. Further it can be assumed that the main components in measurement will be the noise components $\mathbf{w}$ and $\mathbf{v}$. The model can be modified in accordance with these assumption to only include the sensor biases, process and measurement noise $\mathbf{w}$ and $\mathbf{v}$ respectively. The

modified system model is

$$\begin{cases} \mathbf{x}(k+1) & = \Phi\mathbf{x}(k) + \mathbf{w}(k) \\ \mathbf{z}(k) & = C\mathbf{x}(k) + \mathbf{v}(k) \end{cases} \tag{11.1}$$

where

$$\mathbf{x} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{N_s} \end{bmatrix} \qquad \mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_{N_s} \end{bmatrix} \qquad \mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{N_s} \end{bmatrix} \qquad \mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_{N_s} \end{bmatrix} \tag{11.2}$$

$$\Phi = \mathbf{I}_{N_s} \qquad G = \mathbf{I}_{N_s} \tag{11.3}$$

This model and the assumptions under which it has been devised is the basis for this chapter. The final objective this chapter is the identification of process and measurement noise covariances Q and R respectively

$$Q = \underbrace{\begin{bmatrix} \sigma_{w,1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_{w,N_s} \end{bmatrix}}_{\text{diag}(\sigma_{\mathbf{w}})} \underbrace{\begin{bmatrix} 1 & q_{1,2} & \cdots & q_{1,N_s} \\ q_{2,1} & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & q_{N_s-1,N_s} \\ q_{N_s,1} & \cdots & q_{N_s,N_s-1} & 1 \end{bmatrix}}_{Q_c} \underbrace{\begin{bmatrix} \sigma_{w,1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_{w,N_s} \end{bmatrix}}_{\text{diag}(\sigma_{\mathbf{w}})}$$

$$R = \underbrace{\begin{bmatrix} \sigma_{v,1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma_{v,N_s} \end{bmatrix}}_{\text{diag}(\sigma_{\mathbf{v}})} \underbrace{\begin{bmatrix} 1 & r_{1,2} & \cdots & r_{1,N_s} \\ r_{2,1} & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & r_{N_s-1,N_s} \\ r_{N_s,1} & \cdots & r_{N_s,N_s-1} & 1 \end{bmatrix}}_{R_c} \underbrace{\begin{bmatrix} \sigma_{v,1} & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \sigma^2_{v,N_s} \end{bmatrix}}_{\text{diag}(\sigma_{\mathbf{v}})}$$

It can be noticed that the the diagonal elements in Q and R are the variance of the ARW and RRW noise processes for the individual gyroscopes respectively. The value of these has already been identified in chapter 5 on page 39 using the AV analysis. The main focus of this chapter will thus be the identification of the off-diagonal elements of Q and R. In the following section the EM algorithm is applied to simulated data in order to investigate the feasibility of applying it to actual data captured from the project gyroscopes.

11.2 Test By Simulation

In this section the EM algorithm is applied to simulated data. The basis of the simulation is the developed gyroscope model `multigyrosim.m`, which model the scholastic characteristics of the project gyroscopes [3]. The typical values for the ARW and RRW noise, as identified in chapter 5 on page 39 are used to generate the simulated data.

$$\text{ARW}_{C,TYP} \triangleq 0.068^\circ/\sqrt{s} \qquad \text{and} \qquad \text{RRW}_{C,TYP} \triangleq 0.00029^\circ/\sqrt{s^3} \tag{11.4}$$

The correlation matrices Q_c and R_c, which are used to generate the simulated data a generated using the function `randcorr.m`, as illustrated below.

```
1 function [matrix] = randcorr(Ns)
2   S = rand(Ns);
3   S = S'*S; s = sqrt(diag(S));
4   matrix = diag(1./s)*S*diag(1./s);
5 end
```

In the following the randomly generated correlations matrices Q_c and R_c will has been used.

$$Q_c = \begin{bmatrix} 1.0000 & 0.3340 & -0.2752 \\ 0.3340 & 1.0000 & -0.8569 \\ -0.2752 & -0.8569 & 1.0000 \end{bmatrix} \qquad R_c = \begin{bmatrix} 1.0000 & -0.5745 & -0.9722 \\ -0.5745 & 1.0000 & 0.4557 \\ -0.9722 & 0.4557 & 1.0000 \end{bmatrix} \tag{11.5}$$

By applying the known σ_v and σ_w values the following covariance matrices are obtained.

$$Q = \begin{bmatrix} 8.552 \cdot 10^{-8} & 2.977 \cdot 10^{-8} & -2.46 \cdot 10^{-8} \\ 2.977 \cdot 10^{-8} & 8.535 \cdot 10^{-8} & -7.325 \cdot 10^{-8} \\ -2.46 \cdot 10^{-8} & -7.325 \cdot 10^{-8} & 8.594 \cdot 10^{-8} \end{bmatrix} \tag{11.6}$$

$$R = \begin{bmatrix} 0.482522 & -0.277007 & -0.467663 \\ -0.277007 & 0.483537 & 0.218383 \\ -0.467663 & 0.218383 & 0.479254 \end{bmatrix} \tag{11.7}$$

The objective of the following is thus to estimate these two covariance matrices based solely on the data and the known standard deviations σ_v and σ_w. The data generated simulates and array of three gyroscopes where the process and measurement noise are correlated as described by (11.5). The used sampling frequency for the generated data is 1Hz and the duration is one hour i.e. 3,600 samples. The process of applying the EM algorithm to the simulated data is implemented in by the m-file `simcorrident.m` which can be viewed on the project CD. The file also provides all the figures and results used in this chapter. In figure 11.1 the first two minutes of the generated data is illustrated.

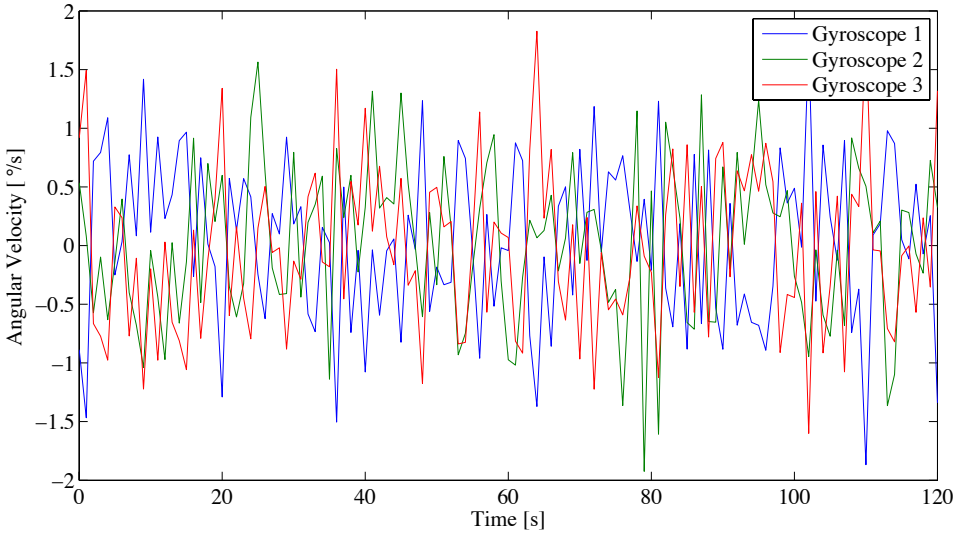

Figure 11.1: Simulated gyroscope data

The next step is to apply the developed MATLAB® implementation of the EM algorithm, `corrident.m`, to the generated data. The function requires the simulated data, the number of individual initializations and the number of iterations. There is no guarantee that the EM algorithm will converge to a global optimum so the algorithm must be run several times with different initialization each time. Evaluating the Log-likelihood for each run is assumed to be a good indication to whether a global optimum has been reached. The final estimate of the process and measurement noise covariance matrices is generated by taking the mean of the individual estimates from each run. Provided that the iterations converge to a similar level of performance the estimate mean value is considered be a good estimate of the actual noise covariance matrices.

One approach of initializing Q and R are to use *a priori* knowledge about the process and measurement noise. Often this information will stem from sensor data sheets or similar

sources and should be taken with a level of uncertainty. In this example however fairly accurate *a priori* knowledge about the noises are present as the σ_v and σ_w values are known in advance. These values will be used in the scaling of the initialization. The effects of using this *a priori* knowledge is illustrated in figures 11.2 and 11.3. Here the development of the Log-likelihood for 10 different initializations of the EM algorithm with and without scaling of the initial guess for Q and R are illustrated.

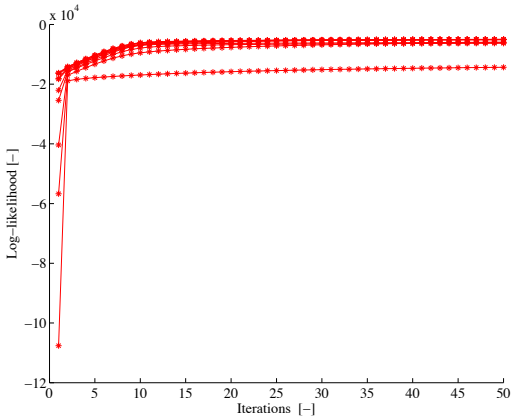

Figure 11.2: Non-scaled initializations

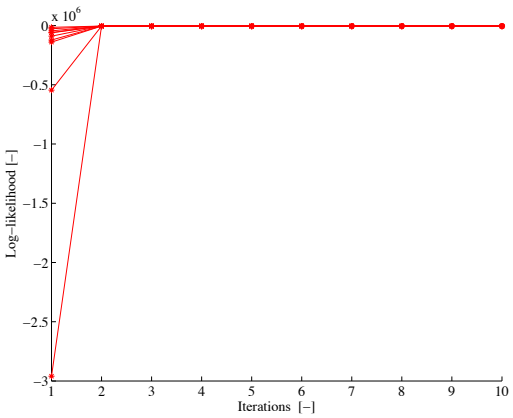

Figure 11.3: Scaled initializations

From figures 11.2 and 11.3 it is clear that the estimate converges much more rapidly when the scaled initialization is used. This approach is considered valid as the sigma values could be acquired from the data using the AV analysis. To this degree the method is considered generic and applicable to data sets which are captured from gyroscopes which can be modeled using the project model.

For the processing below of the simulated data the algorithm is ran 25 times with different initialization each time. The number of iterations used a set to 10 which has been found to be sufficient for the estimate to convergence when scaling of the initialization is used. In figure 11.4 the Log-likelihood as is develops for each iteration is illustrated.

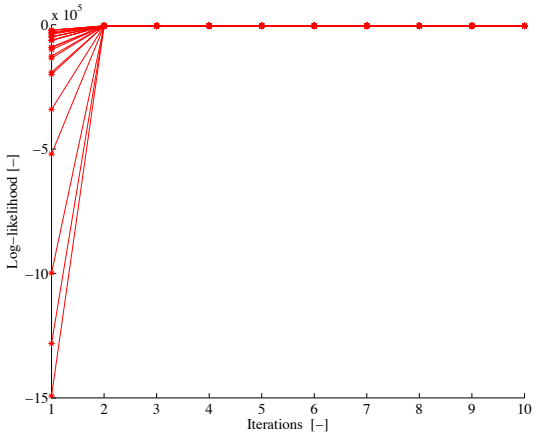

Figure 11.4: Plot of the Log-likelihood development with 10 different initializations

From figure 11.4 it can be seen that the Log-likelihood is improved from one iteration to another and that the performance seams to converge to the same level. This indicates that a global optimum has been reached and further iteration will not add to the quality of the estimate.

11.2.1 Results

By applying the `corrident.m` to the generated data a sequence of estimates for Q and R, related to the different initializations, are created. In the following the collective results from the test are presented.

The mean value of the covariance matrices Q and R, after running the EM algorithm 25 times are as follows.

$$\bar{Q} = \begin{bmatrix} 8.531 \cdot 10^{-8} & 6.473 \cdot 10^{-8} & 6.252 \cdot 10^{-8} \\ 6.473 \cdot 10^{-8} & 8.654 \cdot 10^{-8} & 6.63 \cdot 10^{-8} \\ 6.252 \cdot 10^{-8} & 6.63 \cdot 10^{-8} & 8.504 \cdot 10^{-8} \end{bmatrix} \tag{11.8}$$

$$\bar{R} = \begin{bmatrix} 0.482234 & -0.276864 & -0.467406 \\ -0.276864 & 0.484247 & 0.218070 \\ -0.467406 & 0.218070 & 0.479071 \end{bmatrix} \tag{11.9}$$

The standard deviation for the covariance matrices Q and R, after running the EM algorithm 25 times are as follows:

$$\sigma_Q^2 = \begin{bmatrix} 4.572 \cdot 10^{-9} & 1.792 \cdot 10^{-8} & 1.607 \cdot 10^{-8} \\ 1.792 \cdot 10^{-8} & 8.167 \cdot 10^{-9} & 1.597 \cdot 10^{-8} \\ 1.607 \cdot 10^{-8} & 1.597 \cdot 10^{-8} & 4.078 \cdot 10^{-9} \end{bmatrix} \tag{11.10}$$

$$\sigma_R^2 = \begin{bmatrix} 1.664 \cdot 10^{-5} & 6.189 \cdot 10^{-5} & 1.694 \cdot 10^{-5} \\ 6.189 \cdot 10^{-5} & 0.0002369 & 7.373 \cdot 10^{-5} \\ 1.694 \cdot 10^{-5} & 7.373 \cdot 10^{-5} & 2.578 \cdot 10^{-5} \end{bmatrix} \tag{11.11}$$

It can be seen from the above, that standard deviation for the covariance matrices Q and R estimates are quite small. To evaluate how well the algorithm estimates the actual Q and R the estimation error has been plotted in figure 11.5 and 11.6. The figure is generated using the function `plotestcorr.m` which can be found on the project CD. The figures illustrate the error in estimation the individual elements of Q and R for different initializations. As Q and R are symmetric only the diagonal and the upper triangular elements are plotted. The lower triangular elements are identical to the upper and are not plotted. The blue dots indicate the estimation error for the individual initializations and the red line indicate the estimation error for the final estimate, that is the mean value of the collective element estimates. From

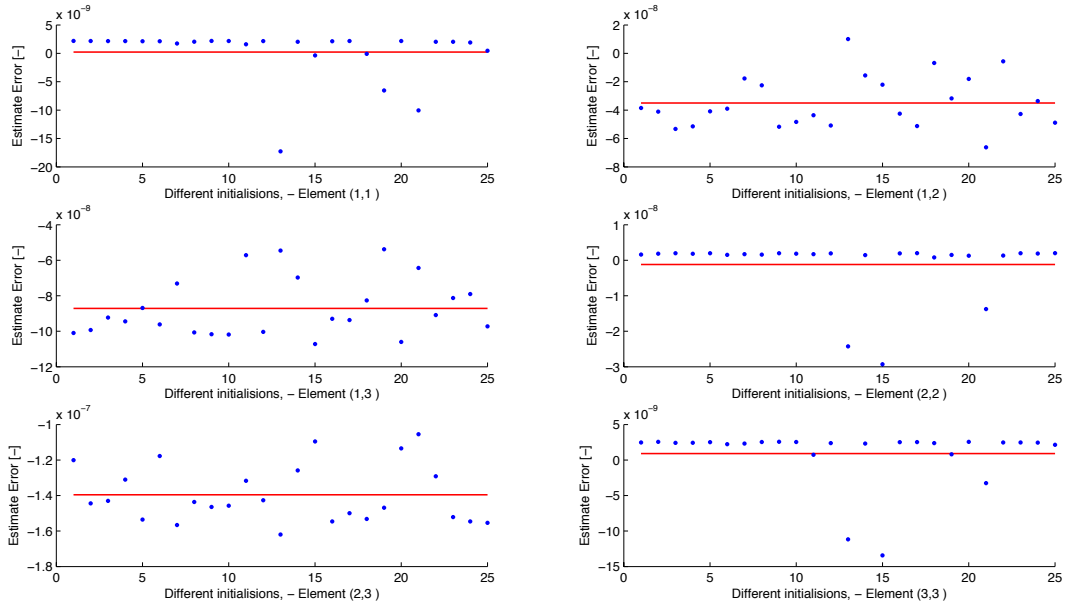

Figure 11.5: Estimate error Q elements

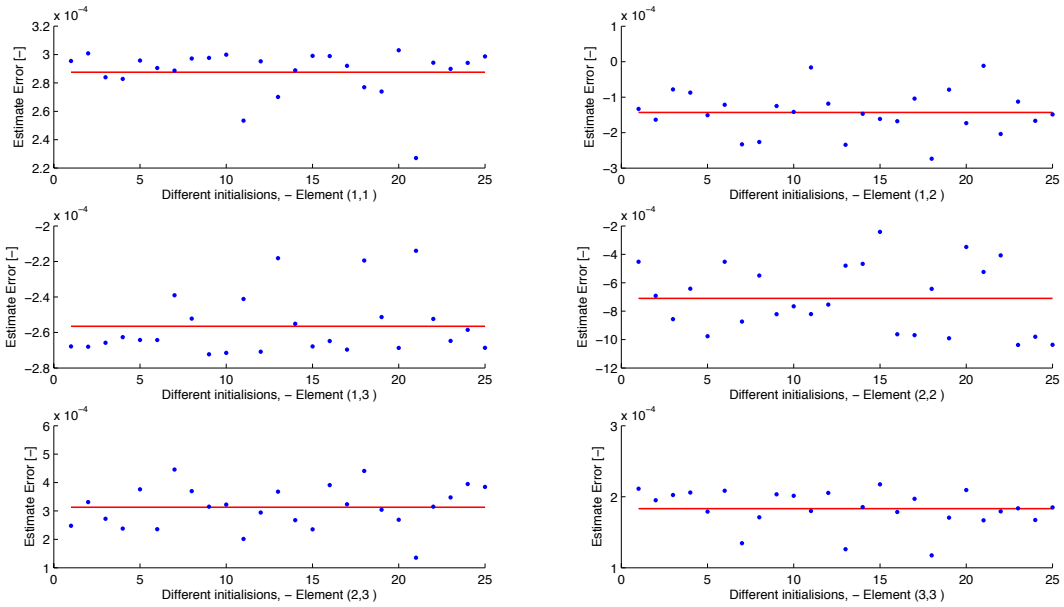

Figure 11.6: Estimate error R elements

figure 11.5 - 11.6 is can be seen that the values for R tend to cluster with a constant offset relative to the correct correlation. The values for Q_c are more spread. The percentage error in estimating the covariances Q and R has been calculated using the function `errorcalc.m`. The resulting percentage error is:

$$\text{Error}_Q = \begin{bmatrix} 0.26\% & 117.43\% & 354.14\% \\ 117.43\% & 1.39\% & 190.51\% \\ 354.14\% & 190.51\% & 1.05\% \end{bmatrix} \text{Error}_R = \begin{bmatrix} 0.06\% & 0.05\% & 0.05\% \\ 0.05\% & 0.15\% & 0.14\% \\ 0.05\% & 0.14\% & 0.04\% \end{bmatrix} \tag{11.12}$$

From the above it can be seen that the EM algorithm estimates the R quite well. On the other hand it can also be seen that$Q$ is not estimated well with the same precision. This is unfortunate as it is know that the larges performance improvement is possible if the correlations related to the process noise $w(t)$ can be identified. To investigate whether the values presented above are representative of the algorithm performance, three additional tests are conducted. In order to condense the results of these tests only the percentage error for these tests will be addressed. Notice that different sets of data are generated for each tests, but using the same parameters as above.

$$1) \quad \text{Error}_Q = \begin{bmatrix} 23.85\% & 170.29\% & 258.58\% \\ 170.29\% & 39.31\% & 171.96\% \\ 258.58\% & 171.96\% & 30.78\% \end{bmatrix} \quad \text{Error}_R = \begin{bmatrix} 0.05\% & 0.11\% & 0.03\% \\ 0.11\% & 0.03\% & 0.12\% \\ 0.03\% & 0.12\% & 0.02\% \end{bmatrix}$$

$$2) \quad \text{Error}_Q = \begin{bmatrix} 0.14\% & 123.86\% & 371.02\% \\ 123.86\% & 2.08\% & 189.71\% \\ 371.02\% & 189.71\% & 0.24\% \end{bmatrix} \quad \text{Error}_R = \begin{bmatrix} 0.05\% & 0.14\% & 0.04\% \\ 0.14\% & 0.08\% & 0.21\% \\ 0.04\% & 0.21\% & 0.01\% \end{bmatrix}$$

$$3) \quad \text{Error}_Q = \begin{bmatrix} 29.61\% & 29.94\% & 527.34\% \\ 29.94\% & 47.34\% & 141.00\% \\ 527.34\% & 141.00\% & 47.74\% \end{bmatrix} \quad \text{Error}_R = \begin{bmatrix} 0.01\% & 0.02\% & 0.01\% \\ 0.02\% & 0.38\% & 0.17\% \\ 0.01\% & 0.17\% & 0.01\% \end{bmatrix}$$

From the above it can be seen that the percentage error in estimating the two noise covariances remain at the same level. This indicate that the achieved level of precision is fairly constant for the specific parameters used in the test.

These test presented in this section can only be considered as a guiding measurement of the algorithms performance. Several parameters are assumed to be significant for the algorithm

performance in relation to correctly identifying the process and measurement noise covariance. From test is is know that the algorithm is able to identify the process noise covariance matrix if the standard deviation σ_v is increased. This situation is equivalent to the the RRW coefficient is being significantly higher than the value it is known to have. This is however a clear indication of that the algorithm is in fact able to identify the process noise covariance, provided that the contribution in the data caused by the process noise is significantly higher than the measurement noise.

This property has been investigated by setting the the measurement noise $v(t)$ to be minuscule. Three tests, holding all other parameters identical to the previous simulation, where conducted. The simulation file `simcorridentlownoise` used for this test can be found on the project CD. The ARW coefficient, which correspond to the measurement noise $v(t)$ was set as follows:

$$\text{ARW}_c = 0.00001^\circ/\sqrt{s} \tag{11.13}$$

Setting the measurement noise to this small value will result in that the main component in the observation is the process noise $w(t)$. This is illustrated in figure 11.7 where the first 120 seconds of data from the of the first of the three simulations are illustrated.

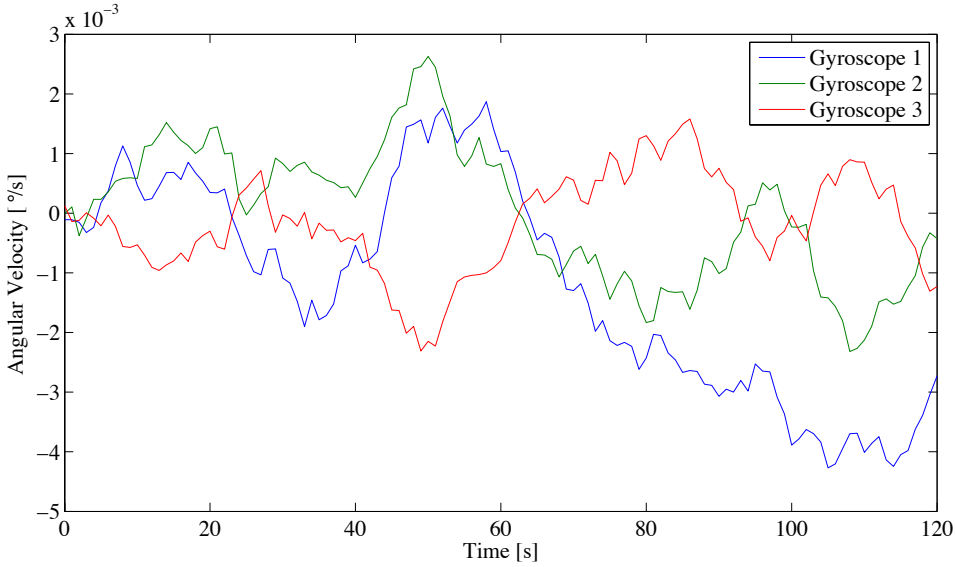

Figure 11.7: Simulated gyroscope data - Low measurement noise $v(t)$

From figure 11.7 is is clear that the main component in the simulated data is no longer the measurement noise $v(t)$ (which is modeled as white noise), but the process noise $w(t)$ which is modeled as a random walk process. The random drift of the signal is fairly obvious from the figure. The percentage error for the three test where as follows

$$1)\quad \text{Error}_Q = \begin{bmatrix} 27.62\% & 73.75\% & 15.45\% \\ 73.75\% & 38.72\% & 55.04\% \\ 15.45\% & 55.04\% & 35.48\% \end{bmatrix} \quad \text{Error}_R = \begin{bmatrix} 49.49\% & 93.19\% & 80.32\% \\ 93.19\% & 31.06\% & 84.24\% \\ 80.32\% & 84.24\% & 40.40\% \end{bmatrix}$$

$$2)\quad \text{Error}_Q = \begin{bmatrix} 32.36\% & 71.47\% & 0.97\% \\ 71.47\% & 35.97\% & 52.53\% \\ 0.97\% & 52.53\% & 37.78\% \end{bmatrix} \quad \text{Error}_R = \begin{bmatrix} 47.24\% & 95.79\% & 72.21\% \\ 95.79\% & 45.90\% & 75.20\% \\ 72.21\% & 75.20\% & 40.93\% \end{bmatrix}$$

$$3)\quad \text{Error}_Q = \begin{bmatrix} 34.75\% & 65.13\% & 4.24\% \\ 65.13\% & 37.56\% & 51.17\% \\ 4.24\% & 51.17\% & 34.66\% \end{bmatrix} \quad \text{Error}_R = \begin{bmatrix} 33.34\% & 59.15\% & 77.86\% \\ 59.15\% & 19.63\% & 82.72\% \\ 77.86\% & 82.72\% & 39.52\% \end{bmatrix}$$

From the above results it can be seen that the algorithm is now able to estimate the process covariance matrix Q with a much higher accuracy. It should be noted that the values that

are estimated are still in the same range as illustrated in (11.6) and are thus quite small. It can also be noted that the error percentage in estimating the measurement noise covariance matrix R has increased significantly. This is assumed to be because the measurement noise $v(t)$ now only make up a small portion of the observations.

The above results indicate the that performance of the algorithm is highly dependent on the ratio between the measurement and process noise. This is unfortunate, at is it know that the measurement noise will always be the predominant component in the observation. This means that the algorithm will not be able to accurately identify the process noise covariance Q. It should be noted that the above results have been presented relative to specific correlations. To this extent the results may not be representative for other levels of noise correlation.

Chapter Conclusion

In this chapter the a modified system model, describing N_s motionless gyroscopes has been presented. Based on the modified system model and randomly generated correlation matrices three correlated gyroscopes have been simulated. The EM algorithm has been applied to the simulated data in order to investigate to what degree the algorithm is able to identify the noise correlations. It was found that the algorithm was able to estimate the measurement noise covariance matrix R quite will, with a low error percentage. The estimation of the process noise covariance matrix Q was not estimated with sufficient accuracy.

The EM algorithm was found be provide significantly better estimates if the measurement noise was lowered, equivalent to a situation where the main component in the observation is the RRW process. This result indicate that the implementation of the algorithm is correct, but that the method will not be applicable in relation to identifying the noise correlations of the gyro-board. Since the results of the simulation have been less than impressive the method has not been applied to actual data from the gyro-board and the existence of noise correlations have thus not been substantiated.

Chapter 12

Project Conclusion

In this chapter the main results of the are summarized and the project is evaluated in relation to the objectives formulated in chapter 2 on page 9. The objective of the chapter is to underline the accomplished work and relevant conclusions.

Modeling

The main starting point for the project has been the analysis of the gyro-board that features eight medium grade MEMS gyroscopes. Based on Allan Variance analysis and classical signal analysis methods, a stochastic model which includes the predominant stochastic errors in the gyro-board output signal has been developed. The model, which considers ARW and RRW noise processes has been implemented in MATLAB® and validated through a comparative analysis with the Root Allan Variance plot for the actual gyro-board gyroscopes. The developed model was found to sufficiently approximate the stochastic characteristics of the gyro-board output signal.

It is noted that the stochastic model is not a complete characterization of the errors in the gyroscope output, but focuses on characterizing the random bias component. Error effects such as scaling, nonlinearities, temperature, magnetic or other environmental variations are not considered directly as they are either partially compensated for by the gyroscope already such as correction tables for temperatures, or considered to have a small contribution to the working assumptions of the project. The stochastic model is complementary to these further modeling and optimizations.

Including modeling of an actual gyroscopes in the project has allowed the group to retain first hand information regarding the performance and limitations of MEMS gyroscopes. The use of the gyro-board has forced the group to include sensor specific considerations, that would not have come into play if the project had been implemented based on a purely theoretical basis.

Sensor Fusion

In this project the group has investigated methods of improving the performance of MEMS gyroscopes through Kalman filtering under the assumption of correlated noise processes. Through simulations it has been shown that by including information regarding noise correlations the effects of angle integration drift can be reduced.

One of the key elements in the project has been to provide a transparent evaluation of the performance improvement made possible due to favorable noise correlations. This has been achieved by introducing a simple averaging strategy as a suitable benchmark. By benchmarking the performance improvement to averaging, the group believes that a more nuanced and easier understood comparison of the performance improvement has been made possible. Compared with averaging, Kalman techniques have shown significant improvements in performance in the angle drift is possible when favorable correlations exist between the gyroscopes noise processes.

The Kalman filter algorithm has uncharacteristically been applied to a non-observable and non-detectable linear state space model. The Kalman filter will not be stable or convergent in its estimation since the uncertainty of the Kalman estimate grows over time, however it is the optimum estimate and no better solution can be found under the assumptions of the model. Also importantly, although the uncertainty of the estimate grows over time, the Kalman filter can be used with a constant gain matrix, making it very convenient for implementation.

Another results of the project relates to the analysis of an approximate constant gain computed by repeating the recursive iterations of the Kalman filter. This approximate gain has been used in simulations without any apparent loss of performance. To this extent, results point to the fact that the method for calculating the constant Kalman gain as presented by [4] might not be necessary in all cases. This is an important result as it opens for the possibility of implementing a filtering strategy based on noise correlations without using the method patented by Bayard and Ploen.

Correlation Identification

The sensor fusion strategies addressed in this project are based on the existence of favorable corrections between noise processes of the gyroscopes. One of the main elements in the project has been to investigate methods of identifying such correlations. The method investigated in this project is based on EM and has been found to correctly identify time constant noise correlation related to the measurement noise under simulation. The method was however not able to identify the noise correlations related to the process noise with sufficient accuracy. A significantly higher level of accuracy is attainable if the ratio between the system and measurement noise levels is increased compared to the gyro-board model, and to this extent the current implementation of the algorithm is considered to function correctly, but not it is not appliable the system of interest. Since the identification method did not perform well during simulated tests, it was not applied to data acquired from the gyro-board.

Chapter 13

Feasibility and Future Work

This chapter address the feasibility of the sensor fusion strategy presented in the project from a product development perspective. A number of open points and recommendations are given as ideas that could be taken into account if the methods presented in the project are subjected to further analysis and investigation.

There are two items of immediate interest, that should be added to the tool and method box of multi-MEMS project: a computation formula for the angle drift computed from the Kalman filter angular rate estimate, and a method that can be successfully employed to estimate w-noise constant correlations for the multi-gyro state-space model. These two problems are formulated in the following sections.

13.1 Kalman Angle Drift

While the increase in performance of estimating the angle has been tested and confirmed through simulation in chapter 10, a precise numerical expression for evaluating the average angle drift for different Kalman filter runs has not been developed yet. The average angle drift will depend on the pattern and nature of the correlations, and a formula similar to the the expression (7.16) obtained for the signal averaging method is needed. This is an important missing piece, because while simulations can be used to evaluate the angle drift for a Kalman scheme, they are laborious and require many runs to average over.

A starting point in developing this expression could be the error uncertainty of the Kalman rate estimate, $\epsilon(k)$, (4.41). This is given by the covariance matrix P that can be calculated using the Riccati recursion.

$$\mathrm{Var}\big(\hat{\epsilon}(N)\big) = P_N^+(1,1) \tag{13.1}$$

The uncertainty of the angle estimate can be expressed in the following way:

$$\begin{aligned}\mathrm{Var}\big(\hat{\alpha_\epsilon}(N)\big) &= \mathrm{Var}\left(\sum_{k=1}^{N}\hat{\epsilon}(k)\right) = \\ &= \sum_{k=1}^{N}\mathrm{Var}\big(\hat{\epsilon}(k)\big) + \sum_{k_1=1}^{N}\sum_{k_2=1}^{N}|_{k_1\neq k_2}\mathrm{Cov}\big(\hat{\epsilon}(k_1),\hat{\epsilon}(k_2)\big) \\ &= \sum_{k=1}^{N}P_k^{+}(1,1) + \sum_{k_1=1}^{N}\sum_{k_2=1}^{N}|_{k_1\neq k_2}\mathrm{Cov}\big(\hat{\epsilon}(k_1),\hat{\epsilon}(k_2)\big) \qquad (13.2)\end{aligned}$$

The first term containing the covariance term P^+ can be evaluated, however the correlation between the rate estimates at different times is more difficult to evaluate in a form that is suitable for fast numerical implementation. The covariance $\mathrm{Cov}\big(\hat{\epsilon}(k_1),\hat{\epsilon}(k_2)\big)$ will not be zero because the error in our Kalman estimation grows with time.

Another approach could be to investigate the angle drift starting with the analytical solution of the Riccati equation as presented by Bayard and Ploen [4].

13.2 Identification of Noise correlations

The existence and identification of noise correlation is an important open point that is missing for a decisive evaluation of the feasibility of the gyro-board project. Once a reliable method of identifying correlation is resolved and confirmed by simulations, the investigation of the actual existence and properties correlations on the gyro-board can begin.

While practical correlations might have time-varying characteristic or other parameters variations - they might depend on the intensity of external disturbances such as forces, vibrations, temperature, etc - it is first of interest to set-up a method of identifying time constant correlations in the w noise. In the thesis, a recursive form of maximum likelihood, Expectation Maximization has been investigated with moderate results. The set-up of the problem remains however relevant. Given a set of stationary measurements from N_s sensors, ω known to be 0, a modified system model is created by eliminating the ω state. The purpose of the identification is to recover Q and R matrices from a set of simulated data obtained with modified system model given arbitrary w and v correlations and with the appropriate noise levels σ_w and σ_v.

A recommendation in this direction is for a method called Autocovariance Least Squares (ALS) [6] developed in 2003-2005 at University of Wisconsin-Madison. The website jbr-www.che.wisc.edu contains a GNU Octave/MATLAB® implementation that is easy to start up. The method gives necessary and sufficient conditions on the model for uniqueness of the estimated covariance, it guarantees unbiased and convergent estimates with an increasing number of points in the data set, and also guarantees positive semidefinite of the estimate. The problem is framed as an convex optimization with constraints.

A preliminary implementation of the method has provided encouraging results, with convergent diagonal estimates for the Q and R matrices when the initial state is a good guess, however the off-diagonal covariance elements have failed to converge. As such, both the implementation and the prerequisite condition for applying ALS need to further addressed before a final conclusion can be drawn.

It is mentioned here that the general assumption about individually packaged sensors units, such as th ADIS16265, is that they are independent. The original [5] proposal points towards a on-chip manufacturing process that can yield consistent and favorable correlation. As such, a general level of reserve is maintained about feasibility of product before correlations can be better investigated. In conclusion, identification of correlations is vital for evaluating the multi-gyroscope board.

Bibliography

[1] Pieter Abbeel and Jared Wood. Kalman filtering, ekf, unscented kf, smoother, em. Technical report, University of California, Berkeley, Department of Electrical Engineering and Computer Sciences.

[2] C. Acar and A.M. Shkel. *MEMS vibratory gyroscopes: structural approaches to improve robustness*. MEMS reference shelf. Springer, 2009. ISBN 9780387095356.

[3] ADIS16265. *Datasheet - Programmable Digital Gyroscope Sensor*, December 2009.

[4] David S. Bayard and Scott R. Ploen. Foundations of virtual gyroscope synthesis. Technical report, California Institute of Technology: Jet Propulsion Laboratory, January 2002.

[5] David S. Bayard and Scott R. Ploen. High accuracy inertial sensors from inexpensive components. Technical report, United States Patent and Trademark Office, April 2005.

[6] James B. Rawlings Brian J. Odelson, Murali R. Rajamani. A new autocovariance least-squares method for estimating noise covariances. *Elsevier Automatica*, 82(42):303–308, 2005.

[7] Datasheet. *Hardware User Manual CM-BF537E V3.x*, February 2010.

[8] John A. Geen and Analog Devices Inc. Coupling for multiple masses in a micromachined device. Technical report, United States Patent and Trademark Office, Junel 1995.

[9] Mohinder S. Grewal and Angus P. Andrews. *Kalman Filtering: Theory and Practice Using MATLAB*. Wiley InterScienceE - John Wiley and Sons, Inc., Hoboken, New Jersey, second edition edition, 2001.

[10] Mohinder S. Grewal and Angus P. Andrews. *Kalman Filtering: Theory and Practice Using MATLAB*. Wiley InterScienceE - John Wiley and Sons, Inc., Hoboken, New Jersey, third edition edition, 2008.

[11] Mohinder S. Grewal, Lawrence R. Weill, and Angus P. Andrews. *Global Positioning Systems, Inertial Navigation, and Integration*. 2007.

[12] IEEE-647. *IEEE Standard Specification Format Guide and Test Procedure for Single-Axis Laser Gyros*, September 1995.

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

[14] Andrew H. Jazwinski. *Stochastic Processes and Filtering Theory*. Academic Press, INC., 1970.

[15] Rudolph Emil Kalman. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME-Journal of Basic Engineering*, 82(Series D):35–45, 1960.

[16] Martin Møller Sø rensen. Learning parameterized maneuvers from multiple demonstrations. Master's thesis, Aalborg University Section for Automation and Control Department of Electronic Systems, Aalborg, Denmark, June 2009.

[17] Jean Lynch-Stieglitz and Kim Cobb. *Lecture Notes - Autocorrelation*, May 2010.

[18] Mao-Lin NI. Existence condition on solutions to the algebraic riccati equation. *Acta Automatica Sinica*, 34(1):85 – 87, 2008. ISSN 1874-1029. doi: DOI:10.3724/SP.J.1004.2008.00085. URL `http://www.sciencedirect.com/science/article/B8H0V-4RKVHND-4/2/6a31c43d377ae0e704bd7fd905852932`.

[19] George T. Schmidt. Ins/gps technology trends. 2010.

[20] Raymond Serway, Robert Beichner, and John Jewett. *Physics for Scientists and Engineers*. Brooks Cole, 6th edition edition, July 2003.

[21] Padhraic Smyth. *Note Set 4: The EM Algorithm for Gaussian Mixtures*. University of California, Irvine, Center for Machine Learning and Intelligent Systems, 2011.

Appendix A

Data Preprocessing

In this appendix the used method of preprocessing data captured from the gyro-board is presented. The method was originally developed for evaluating data sampled using a Windows XP implementation of the gyro-board sampling software which was found to induced a series of errors in the captured data. The sampling software was later moved to a Linux implementation that was found be be significantly more stable and minimize errors in the data. The majority of the sample sessions used in this project are to large to evaluate by physical inspection and therefor the preprocessing has been maintained to give a measure of data quality.

In the following the chain of reasoning behind the development of the preprocessing method is presented. This include evaluation of observed errors and considerations for the compensation of these errors.

Data Acquisition

The basis of the development of the preprocessing method is the identification of specific errors in data sampled using the Windows (XP) implementation of the gyro-board sampling software. In this appendix a specific sample session where data was sampled from the gyro-board as 100Hz for 48 hours. The used setting for the gyro-board where the default settings. The sampled data has been found to contain two main types of error.

Spikes/Outliers

The sampled data where found to contain spikes, that are clearly are not valid measurements of angular rate. An example of these spikes are illustrated in figure A.1, which illustrate raw data from the sample session. As illustrated in figure A.1 unprocessed data contain spikes in the measurement of the angular rate. It is noted that the spikes do not occur in all the gyroscopes at the same time indicating that the problem is sensor specific. It is also noted that the spikes only make up a small part of the collective data set.

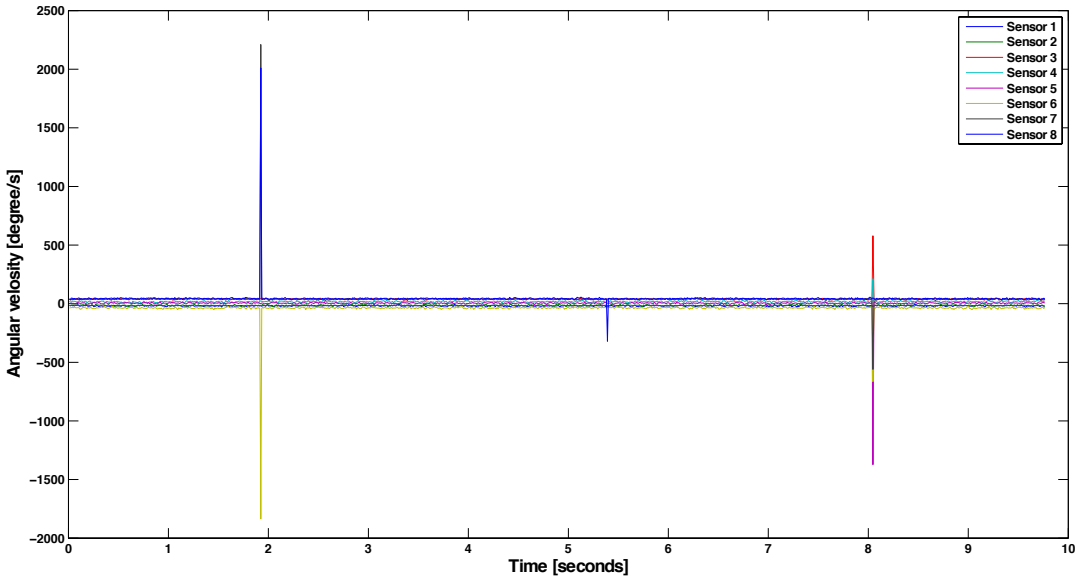

Figure A.1: Raw data from sample session 1

Non-equidistant samples

The length of the sample sessions where defined to be 48 hours. Based on this duration and the sample frequency the number of samples to be taken where calculated. After the number of samples where taken it became apparent that the actual length of the session surpassed the expected 48.

The nature of the problem is further investigated by plotting the time duration between samples (sample period) in the session. In figure A.2 typical sample periods related to the sample sessions are illustrated. Figure A.2 is based on session 1, but shows the typical pattern in the session sample period. It is clear that the duration between samples are varying as the graph would simply be a constant value of $\frac{1}{S_f} \approx 0.01$ if the samples where indeed equidistant. The reason for the delays that are visible in figure A.2 may be processes running behind the

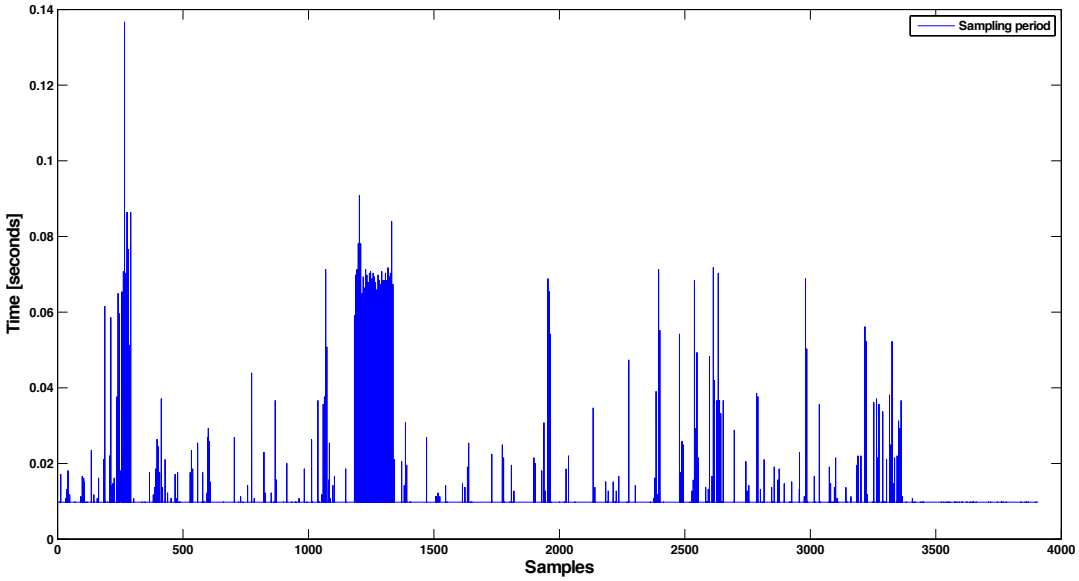

Figure A.2: Illustration of Non-equidistant samples

gyro-board sampling software. As some of the delays can be identified to be of the same length and further more somewhat periodical, the possibility of the delays being caused by processes running behind the sampling software is further substantiated. This problem was anticipated and sought compensated for by changing the priority of the sampling software to the highest level. In spite of this the control PC operating system still failed to provide equidistant samples.

In this section a series of errors related to the sampled data has been identified and presented. In the following section some of this errors are compensated for by implementing means of preprocessing the data prior to usage.

Error compensation

In this section the methods of compensating for the errors are described.

Removal of spikes

The approach chosen in relation to compensating for the spikes is a crude one. The approach is based on the fact that range of the noise in individual gyroscopes are very similar. This allow the use of a single comparison filter that removes samples that are outside the expected range. If a sample from one of the gyroscopes are found to be outside the range the collective samples in the particular instance of time is removed. This approach will remove valid samples but will also enable that samples from the individual gyroscopes can be related to each other.

Implementation

The removal of the data spikes is implemented using the MATLAB® function `spike.m`, which is presented hereunder. The scalar value of `minrange` and `maxrange` are determined empirically from looking at the the data.

```
function [data] = spike(data,minrange,maxrange)
% function [data] = spike(data,minrange,maxrange)
% INPUTS:       data:       N × 1 data array [.]
%               minrange:   Minimum value allowed   [.]
%               maxrange:   Maximum value allowed   [.]
% OUTPUTS:      data:       Filtered data           [.]

for x = 2:size(data,2)-1
    [I] = find( data(:,x) > maxrange(:,x-1) );
    if (~isempty(I))
        data(I,:)=[];
    end
end

for x = 2:size(data,2)-1
    [K] = find( data(:,x) < minrange(:,x-1) );
    if (~isempty(K))
        data(K,:)=[];
    end
end
end
```

Removal of Non-equidistant samples

A recurring sample delay problem was often observed. The delay can be detected by looking at the Blackfin timestamp field, and the first sample after the delay was corrupted. The followings script detects long delays between samples and processes the data to drop each first sample after such a delay.

Implementation

The removal of the non-equidistant samples is implemented using the MATLAB® function `gyrotimestamp.m`, which is presented hereunder.

```
function [data] = gyrotimestamp(data,sfc,sfb)
% INPUTS:       data:       N × 1 data array                    [−]
%               sfc:        Computer clock frequency            [Hz]
%               sfb:        Gyro board sample frequency         [Hz]
% OUTPUTS:      data:       Processed data                      [−]

bc=0;
for jj=1:1:length(data)−2
    if ( data(jj+1,1) − data(jj,1) ≥ 10000*sfc/sfb )
        bc=bc+1;
        bad(bc)=jj+1;
    end
end
if ( bc>0 )
    data(bad',:)=[];
end
```

Appendix Conclusion

Using the methods described in this appendix the it is possible to not only remove the faulty data points, but also evaluate the quality of the sampled data. Based on the amount of sampled that are removed from the original data set a measure of the quality of the sampled data is given. This information is used in evaluation of the data acquired using the new Linux implementation of the sampling software.

Using the Linux implementation provides data which do not contain the errors described above. The preprocessing method is however implemented as a means of evaluating the quality of the data as apposed to investigating the data manually.

In order to substantiate the claim that the data sampled using the Linux implementation of the sampling software we now present a small evaluation of a test session. In the following a 10 minutes sample from one of the gyro-board gyroscopes are considered. The number of samples are $N = 1228800$. The data is illustrated in figure A.3. By just looking at the data it is clear outlines are no longer a problem. In figure A.4 the delays between samples are illustrated. From figure A.4 is can be seen that the samples are much more equidistant than in the previous implementation on XP. Figure A.2 shows the delays in a sample session using the XP implementation. It is clear the the samples are now almost equidistant.

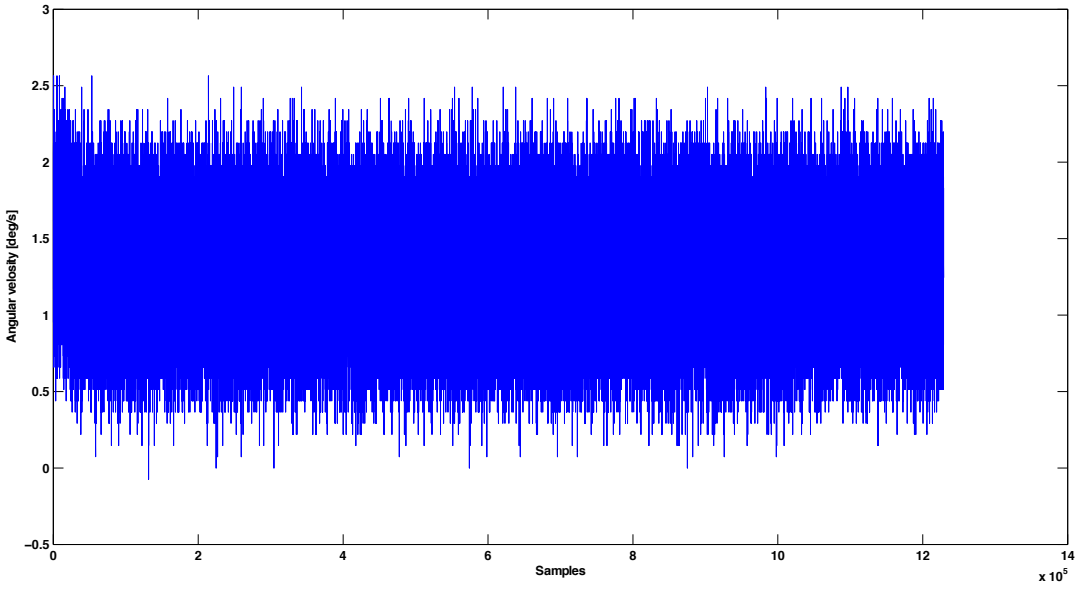

Figure A.3: Illustration of the sampled data (Linux implementation)

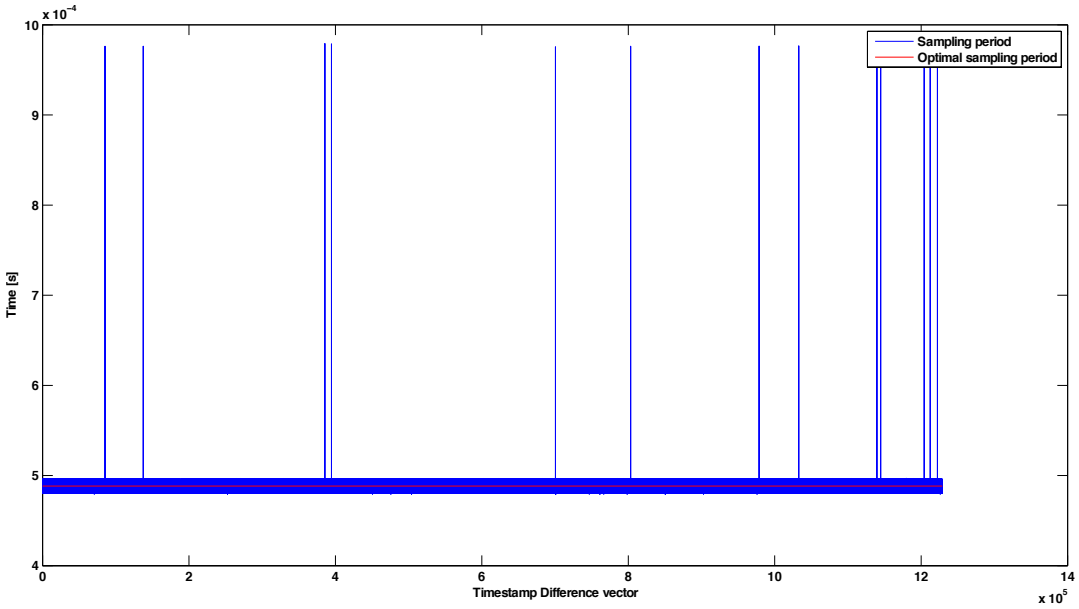

Figure A.4: Illustration of the delays in the sample (Linux implementation)

This page intentionally left blank

Appendix B

Replication of ADIS16265 Root Allan Variance Plot

In this appendix Root Allan variance plot as presented in the ADIS16265 [3] data sheet is recreated. The appendix describes the process of acquiring the necessary data, including the relevant considerations and the used test setup.

This test is different from the others test performed in the project as it uses an environment chamber, that allow the temperature of the gyro-board to be controlled. The opportunity of using such a chamber presented itself while the group visited CDL in the beginning of April 2011. CDL have several high grade test equipment at their disposal and the group where so fortunate to be able to perform tests of the gyro-board during our visit. In the following the tests concerning the the replication of the data sheet Root Allan variance plot is addressed.

Considerations

The ADIS16265 [3] data sheet provides the Root Allan variance plot related to the gyroscope ADIS16265 [3]. This plot is illustrated in figure B.1. In effect it is desired to replicate figure

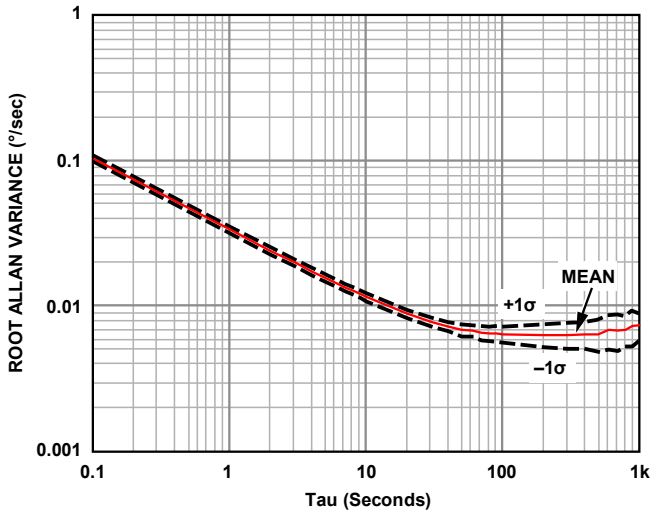

Figure B.1: Typical Performance Characteristics [3] - Root Allan variance plot

B.1 to substantiate that the implemented method of identifying noise processes in the gyro-board output signal is valid. From figure B.1 it can be seen that the maximum cluster size used to generate the plot has been 1000 seconds. Based on this piece of information we are able to backwards calculate the required session length to replicate the figure.

The larges cluster size considered in figure B.1 is 1000 seconds. It is known that if a 25% accuracy in the estimate is desired at least 9 clusters must be considered. This means that a minimum of 9000 seconds or 2.5 hours is needed to recreate the Root Allan variance plot. We however consider 2.5 hours to be the very minimum required session time as it will only provide a 25% accuracy in the estimate. In order to further improve the estimate we consider a much longer session with a duration of 20 hours. In this appendix we use a under-sampling factor of 20. This results in the following sampling frequency.

$$S_f = \frac{2048 \text{ Hz}}{20} = 102.4 \text{ Hz} \tag{B.1}$$

Using this sampling frequency and sampling and a sample duration of 20 hours results in a collective sample with length N = 102.4·20 hours = 7372800 samples. Using this information the accuracy of the estimate at clusters with n = 1000 seconds can be calculated as below.

$$\sigma_{AV} = \frac{1}{\sqrt{2\left(\frac{N}{n} - 1\right)}} \tag{B.2}$$

$$= \frac{1}{\sqrt{2\left(\frac{7372800}{1000} - 1\right)}} = 0.00823\% \tag{B.3}$$

This is sufficiently close to the actual value of the variance at n = 1000, and well within the $\pm 1\sigma$, indicated by figure B.1. In the following the process of acquiring the 20 hours of data is addressed.

Test-setup and procedure

As mention previously the the data used in this appendix was acquired by sampling the gyro-board while in a temperature controlled environment chamber. In the following the tests setup is described.

The test setup, as illustrated in figure B.2, consists of an isolated wooden box (environment chamber illustrated in figure B.2) fitted with a Peltier heating/cooling element. The Peltier element is controlled by a control PC, that uses temperature information measured inside the box to maintain a specific temperature in the box. The temperature control system and the Peltier element from will not be addressed in further.

The data from the gyro-board is sampled at 102.4Hz. The complete step by step description of the test is given hereunder.

1. Turn on Sampling PC
2. Turn on Control PC
3. Connect data and power cables to gyro-board
4. Connect data and power cables to Sampling PC
5. Connect temperature sensor and Peltier element control cable to the Control PC
6. Turn on Peltier element power supply

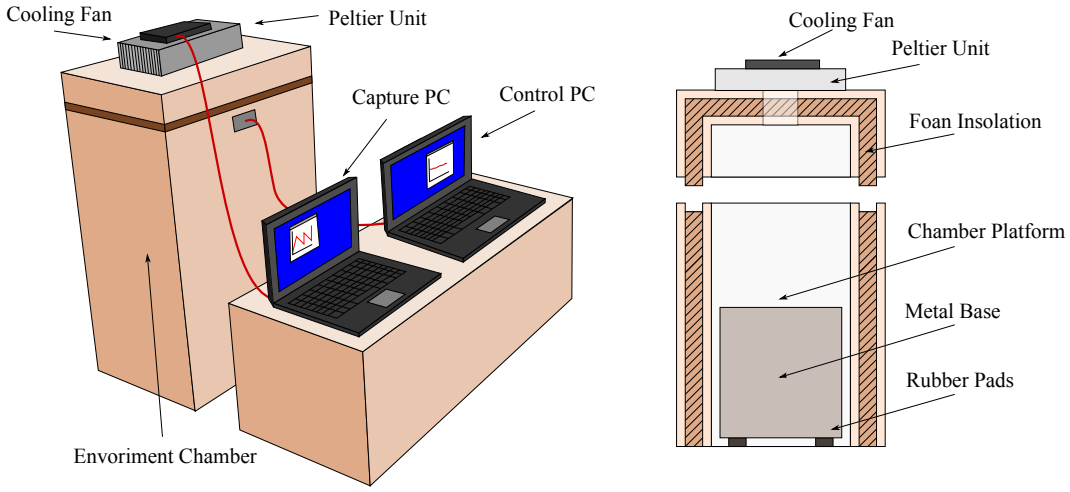

Figure B.2: Illustration of test setup - CDL 06.04.2011

7. Securely position the gyro-board on the dampened platform in the in environment chamber using adhesive pads
8. Test that the temperature sensor is working by monitoring the temperature as the sensor is held in a closed hand
9. Start the gyro-board sampling software on the Sampling PC
10. Verify that the gyro-board is providing samples by taking a few samples
11. Place temperature sensor in close proximity to gyro-board using adhesive tape
12. Close chamber access door
13. Initialize the temperature control using the Control PC software to maintain a constant temperature of 20°
14. Initiate the measurement by requesting 24 hours = 1000,00 samples from the gyro-board sampling software prompt
15. Leave the room and return after 24 hours
16. Verify the that the requested amount of samples has been sampled by looking at gyro-board sampling software prompt
17. Stop the temperature control using the Control PC software
18. End of test session

We acknowledge that this description of the used test setup is somewhat crude. We however do not consider this to be of significant importance as the test can only be recreated perfectly using the setup at CDL. We therefor reserve our right to state that similar results should be obtainable using a similar setup and not go into details about the used equipment. Provided that information about the specific equipment is required we advise that CDL is contacted directly for support. The main focus point in the test of the gyro-board is the unit is isolated from external vibrations and that the temperature is maintained at a constant 25° during the test session.

Session Data

In this section we present the raw results from the sampling session and the results of the preprocessing. Note that the preprocessing implements both evaluation of the sampled data and unit conversion from bits to $^\circ/s$.

In figure B.3 the first 1 second of the sampled data is illustrated. The raw data is processed

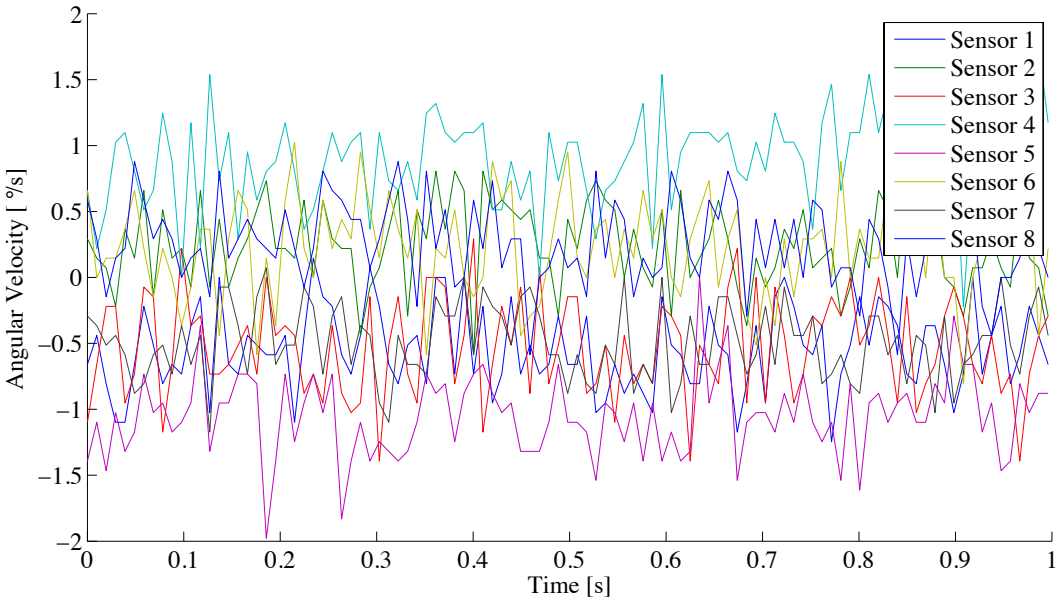

Figure B.3: Sample data from gyro-board - CDL Test

using the script file `datafilt.m`. By applying the script to the acquired data a measure of the data quality can be obtained. The result of the processing was the no samples where removed indicating that no faulty samples where present in the data, and that no data packets where lost.

In figure B.4 the temperature log during the sample session is illustrated.

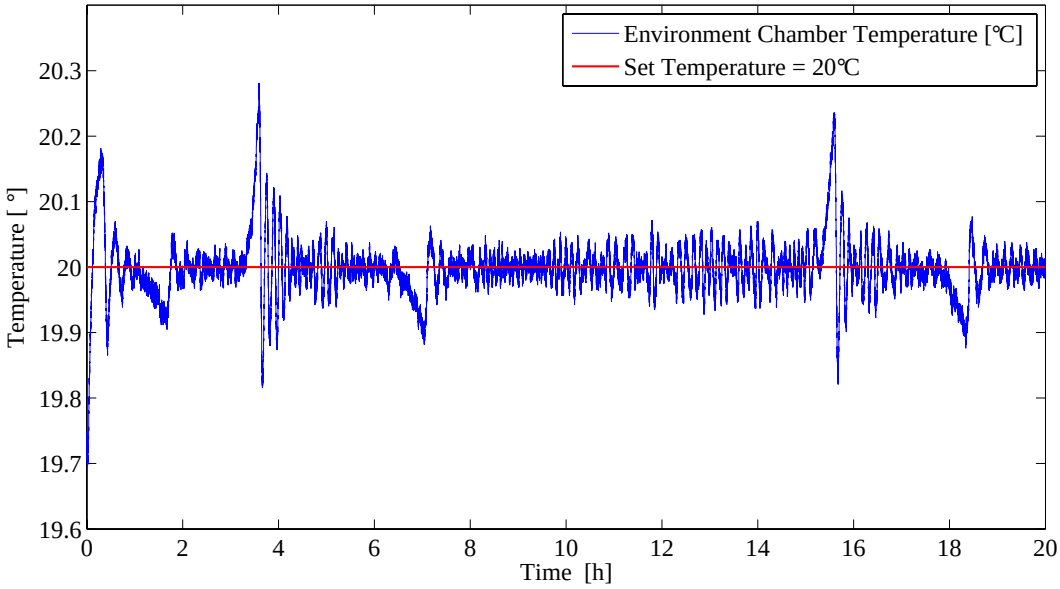

Figure B.4: Environment chamber Temperature Log - CDL Test

From figure B.4 it can be seen that the temperature is maintained close to the desired set temperature throughout the session. In the beginning of the session there is a short period where the temperature has not risen to the desired set temperature. This is however considered to be of minuscule importance as the initial temperature of the temperature chamber was approximately 19° which is relatively close to the set temperature.

Data processing

In this section we process the acquired data with the Allan variance MATLAB® script file `allan.m`, as described in appendix D. The file calculates the Root Allan variance and is used in connection with the plotting tool `plotallan.m` to identify the values of the Bias stability and ARW noise coefficients. In the following we present the Root Allan variance for the eight gyroscopes that make up the gyro-board, as calculated from the acquired data.

In figure B.5 the Root Allan variance plot of the eight gyroscopes are illustrated.

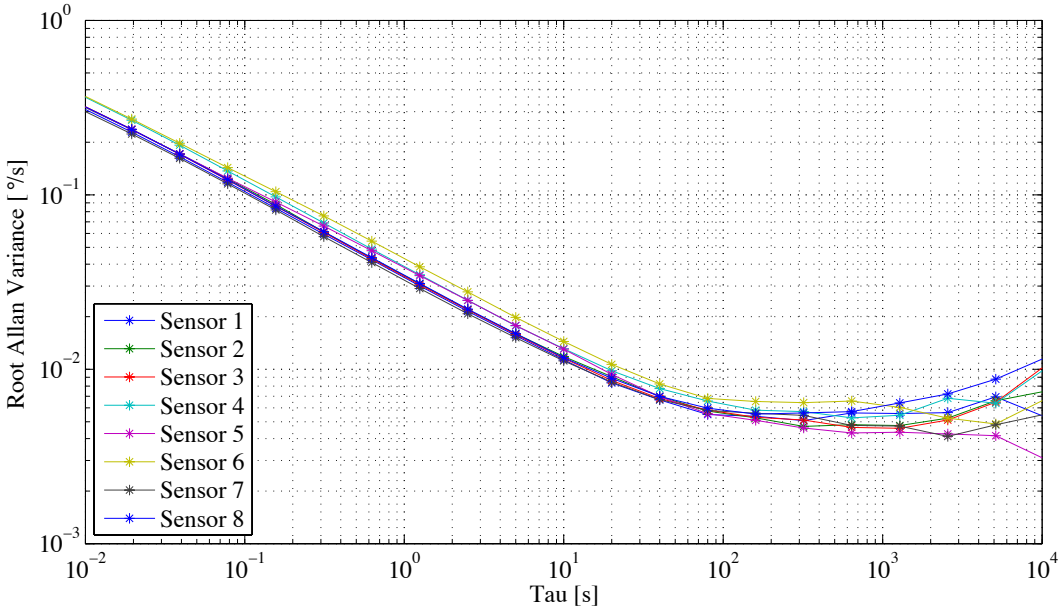

Figure B.5: Root Allan variance plot

Figure B.5 illustrates the collective Root Allan variance plots. Based on the figure it is obvious that the individual gyroscopes, as expected, has similar noise characteristics. Using the MATLAB® script file `cdl_figures.m` the Bias stability and ARW noise coefficient for the individual gyroscopes have been found. For clarity we have omitted the individual plots and only present the values as they have been estimated. The collective values are given in table B.1. The plots used in the process of identifying the values of the noise components can be viewed on the project CD.

Gyro	Bias Stability $°/s$	ARW $°/\sqrt{s}$
1	0.005187	0.033261
2	0.0046887	0.033939
3	0.0045031	0.033939
4	0.0052928	0.037546
5	0.0042383	0.038312
6	0.0063479	0.042383
7	0.0045031	0.032596
8	0.005511	0.033939
Mean	0.005034	0.035739
Variance	$480.0500 \cdot 10^{-9}$	$11.3953 \cdot 10^{-6}$

Table B.1: Results from Allan Variance Analysis

The data sheet [3] provides the value of the Bias stability and the ARW coefficients. The

Bias stability is given in $[^\circ/s]$ and the ARW is given in $^\circ/\sqrt{h}$. In order to compare the ARW noise coefficient to the estimated value the parameter must be converted to the same units. The transformation is given in equation B.5.

$$\begin{aligned} \text{ARW} &= 2 && [^\circ/\sqrt{h}] && \text{(B.4)} \\ &= 2^\circ/\sqrt{60 \cdot 60} && [^\circ/\sqrt{s}] && \text{(B.5)} \\ &= 0.03333 && [^\circ/\sqrt{s}] && \text{(B.6)} \end{aligned}$$

From equation B.5 it can be seen that the estimated value for the ARW noise coefficient is quite close to that given in the ADIS16265 [3] data-sheet. The variance of the found values is impressively small and indicate that the sensor are ended very similar.

Appendix Conclusion

In this appendix we have sought to recreate the Root Allan variance from citetgyro data-sheet. This has been done by sampling the gyro-board at 102.4 Hz for for 20 hours while at a constant external temperature of 20°. Based on the acquired data the Root Allan variance has been calculated and plotted.

The Bias stability and ARW noise coefficients has been estimated based on the Root Allan variance plots and have been found to be reasonably similar to the values presented in the citetgyro data sheet. Based on this we consider it proven that the

Appendix C

Quantization Noise Analysis

During the initial analysis of the gyro-board, what is believed to be quantization noise, was observed in the board output signal. The appearance of the noise however exhibited some properties, which are not considered standard in relation to quantization noise. In this appendix the nature of the noise is investigated further to substantiate that noise is in fact caused by quantization.

Considerations

While working with the gyro-board it was noticed that the data showed plateaus at the sensor resolution values. This is normally a clear indication that quantization of the signal has occurred. The signal was however found to contain some level of abnormality as it contained fast varying element immediately followed by plateaus, where the signal varied significantly less. The combination of different characteristics have been found to be somewhat suspicious and is therefor investigated further.

The properties of the alleged quantization noise is investigated by changing the project gyroscope dynamic range settings. By sampling the gyro-board output signal and evaluating the resulting data it is possible to identify whether or not the noise in the signal is in fact caused by quantization. The specifications under which the data is sampled are addressed in the following.

Test-setup and Procedure

In this section the used test-setup and procedure is described. The main objective of this appendix is to investigate whether or not the described noise characteristic is caused by quantization noise. This is investigated by sampling the gyro-board using two different dynamic range settings.

The range of the project gyroscopes limits the obtainable level of accuracy in the output signal as the number of bits available to represent the measured rate do not change. If the dynamic range is set to the maximum $320°/s$ the sensor resolution becomes much more coarse. As the range of the sensor noise is unchanged the appearance of noise will now seam quantized as less bits are used to represent the signal. The question of dynamic range is obviously related to the properties of the process that is being measured. The problem can be compared to

the process of looking at an object. If the object is large the observer must stand with some distance to the object to fully observe the properties of the object. If the object is small it will be hard to observe from a distance and a much close look will be advisable.

For the test a sampling frequency of 2048Hz was used along. Note that only one sensor is considered in the following as all sensors signals exhibit similar characteristics. The following is a step-by-step description of the test procedure used for the two tests. The two settings for the gyro-board dynamic range are $80°/s$ and $320°/s$ respectively.

1. Initialize the gyroscope dynamic range setting
2. Turn on Sampling PC
3. Connect data and power cables to gyro-board
4. Connect data and power cables to Sampling PC
5. Securely position the gyro-board on the foam isolation pad
6. Start the sampling software on the Sampling PC
7. Initiate the measurement by requesting 2 minutes = 245760 samples from the gyro borad sampling software prompt
8. Wait until the requested number of samples have been captured.
9. End of test session

The test is performed using two different settings for the dynamic range. The used settings where $320°/s$ and $80°/s$.

Results

In this section the raw results from the sampling session are presented and the results of the preprocessing. Note that the preprocessing implements both evaluation of the sampled data and unit conversion from bits to $°/s$. This is an important step as the scalar conversion value changes according to the dynamic range was used during the test. In accordance with the gyroscope data sheet ADIS16265 [3], the conversion values 0.07326 and 0.01832 where used for the $320°/s$ and $80°/s$ dynamic range respectively.

The raw data is processed using the script file `datafilt.m` as described in appendix A on page 99. By applying the script to the acquired data a measure of the data quality can be obtained. The result of the processing was the no samples where removed indicating that no faulty samples where present in the data, and that no data packets where lost. The script provides the collective measurement for each of the individual sensors in the gyro-board, but in the following only gyroscopes 1 is considered as the characteristics of the gyroscope are more or less identical.

In figure C.1 data from sensor 1 during the two tests are illustrated. Only half a second of the data is considered as it illustrates the properties of interest must better that the collective data set. From figure C.1 it is easy to see that the noise is in fact caused by quantization. The plateaus that are visible in in the data sampled using the $320°/s$ dynamic range are not present in the data sampled using the $80°/s$ dynamic range. This indicates that the plateaus, as expected, are caused by quantization of the gyroscope noise. It can also be noted from the figure that the overall appearance of the two signals are quite similar, the only difference being that different resolution has been used for the to sample sessions.

Appendix Conclusion

In this appendix the properties of the the gyro-board output signal, in particular the dynamic range, has been investigated. The objective of the investigation has been to substantiate

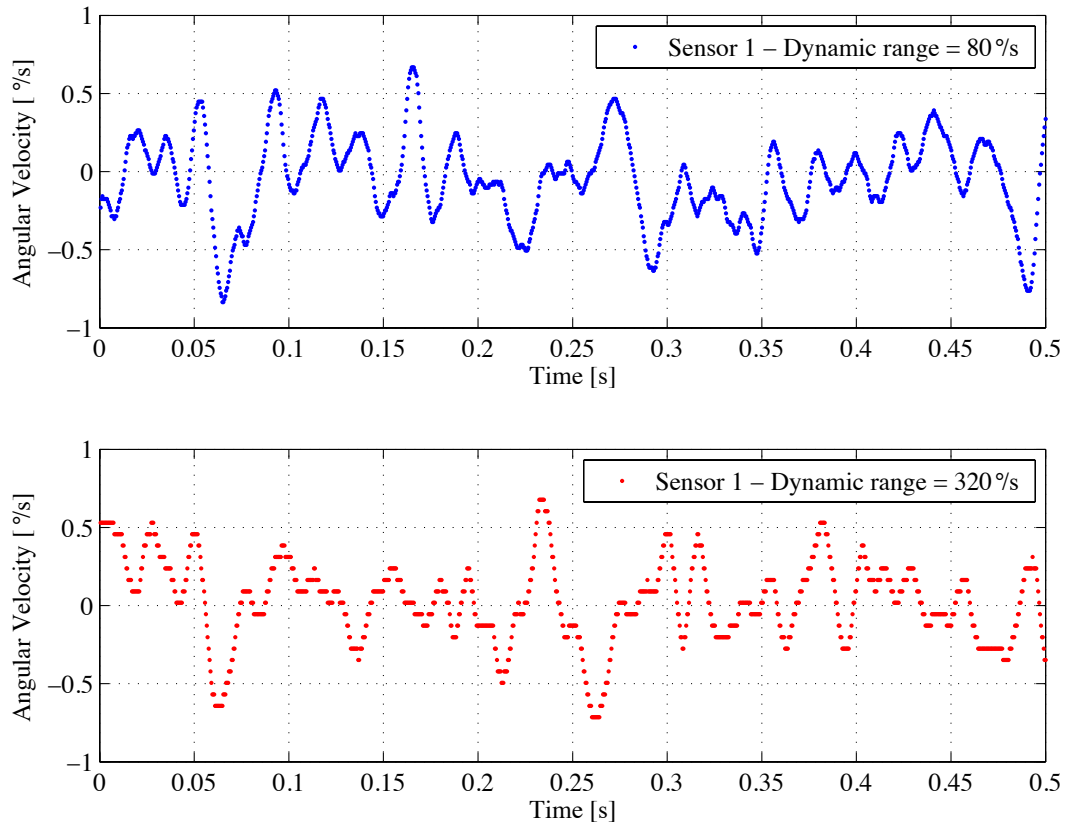

Figure C.1: Sample data from gyro-board - AAU Test

that the plateaus observed in the data using the $320°/s$ dynamic range is indeed caused by quantization of the signal and not some other phenomenon.

The properties of the the gyro-board output signal has been investigated by sampling the it using different sensor settings. The only difference between the two session where the dynamic range setting which was set to $80°/s$ and $320°/s$ respectively. The data was sampled at 2048Hz for 2 minutes, though only half a second was considered during the results evaluation.

By plotting the data from the two sample sessions it is possible to see that that plateaus in data captured using the $320°/s$ dynamic range is caused by quantization. The data captured using the $320°/s$ dynamic range setting show similar properties. From the figure, it is obvious that the resolution of the gyroscope is the critical reason for the plateaus in the data.

This page intentionally left blank

Appendix D

Allan Variance Method

In this appendix the AV method, as used in this project, is described in detail. The objective of the appendix is to address the points that where neglected in the methods chapter 4 on page 17. In addition the usefulness of the AV method is validated by applying it to simulated data which contain known noise processes.

Cluster analysis

For clarity the cluster operations which is the main element in the AV method is repeated in the following. Consider the output z from a sensor sampled in discrete time given as $t = k\tau_0$, $k = 1, 2, 3, \ldots N$. Thus the finite dataset from the sensor consists of N consecutive data points with a sample period of τ_0. To better illustrate the following cluster operations, a sequence of simulated data have been generated. The data contains 1000 samples and has been generated using the file MATLAB® `allanexample.m` which can be viewed on the project CD. The data, illustrated in figure D.1, resemble the output from a MEMS gyroscope sampled at 100Hz and consists of 1000 samples i.e. 5 seconds.

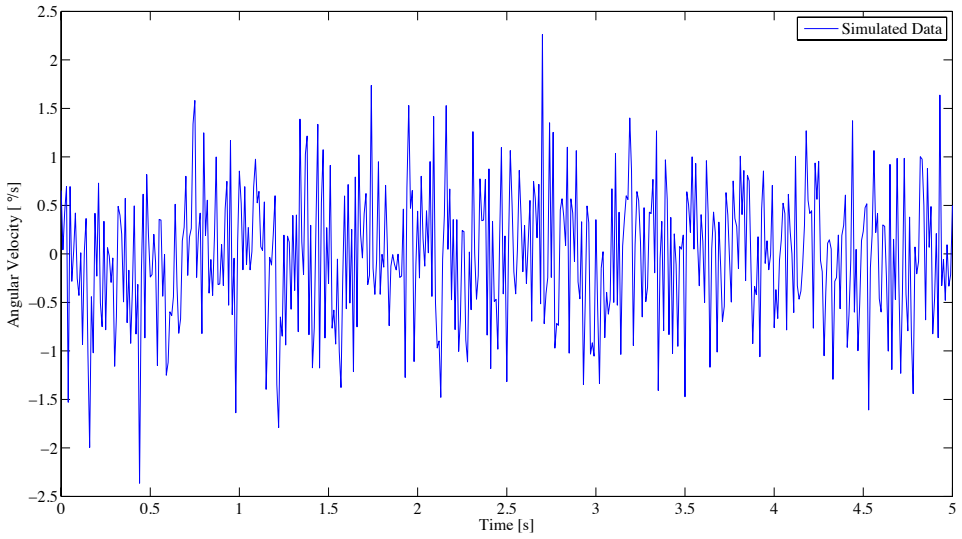

Figure D.1: Simulated MEMS gyroscopes output - AV method

Having acquired a finite data set, the next step is to divided the data into clusters with specific length τ_0, $2\tau_0$, ... $M\tau_0$, with $M < \frac{N}{2}$. M denotes the number of data points within the individual clusters and τ denotes the cluster correlation time equal to $M\tau_0$. The process of dividing the data into clusters with different correlation time τ is illustrated in figure D.2 for $\tau = 0.1$s, $\tau = 0.2$s and $\tau = 1$s respectively. This diversion along with the sampling period $\tau_0 = 100$Hz results in clusters that contain $M = 10$, $M = 20$ and $M = 100$ samples respectively.

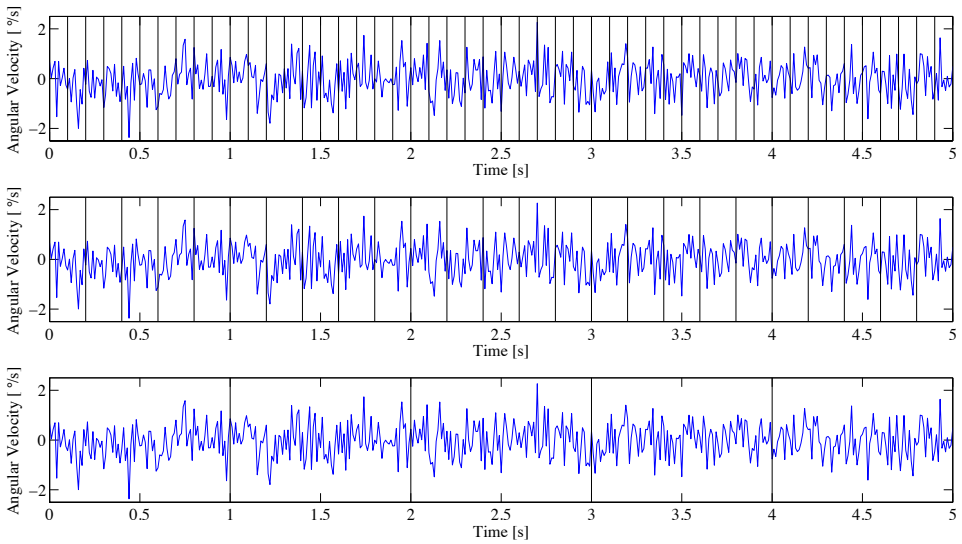

Figure D.2: Creation of clusters with different correlation time - AV method

Associated with each cluster k, containing M data points, is a cluster average $\bar{z}$ which is calculated as follows.

$$\bar{z}_k(\tau_M) = \frac{1}{M}\sum_{i=1}^{M} z_{(k+1)M+i} \tag{D.1}$$

$$\underbrace{z_1, z_2, \ldots, z_M}_{k=1}, \underbrace{z_{M+1}, \ldots z_{2M}}_{k=2}, \ldots, \underbrace{z_{N-M}, \ldots z_N}_{k=K} \tag{D.2}$$

where $k = 1, 2, \ldots, K$ and $K = \frac{N}{M}$. The process of dividing the data into clusters with different correlation time τ and performing the cluster averaging operation can be illustrated as in figure D.3 where $\tau = 0.1$s, $\tau = 0.2$s and $\tau = 1$s respectively. Notice how the variance of the cluster averages changes relative to the used correlation time τ.

For each set of clusters with length M, that can be formed from the data set, the successive cluster averages as defined above form a set of random variables. The object of interest, the Allan Variance σ_A^2, is the variance over all the clusters of length M. The AV $\sigma_A^2(\tau)$ evaluated for cluster with length M is defied as

$$\sigma_A^2(\tau) = \frac{1}{2}\mathrm{E}\left\langle [\bar{z}_{k+1}(M) - \bar{z}_k(M)]^2 \right\rangle \tag{D.3}$$

In this project the standard estimator (D.3), is used in which the mean is simply estimated using the sample mean. This approach was chosen as this estimator is known to provides sufficient results with acceptable processing time. Other AV estimators exist for example the overlapping estimator. The overlapping estimator, works by shifting the data and creating a much higher number of independent clusters, than possible with the standard method of computation. The overlapping estimator is calculated as follows

$$\sigma_A^2(\tau) = \frac{1}{2(K-1)}\sum_{k=1}^{K-1} [\bar{z}_{k+1}(M) - \bar{z}_k(M)]^2 \tag{D.4}$$

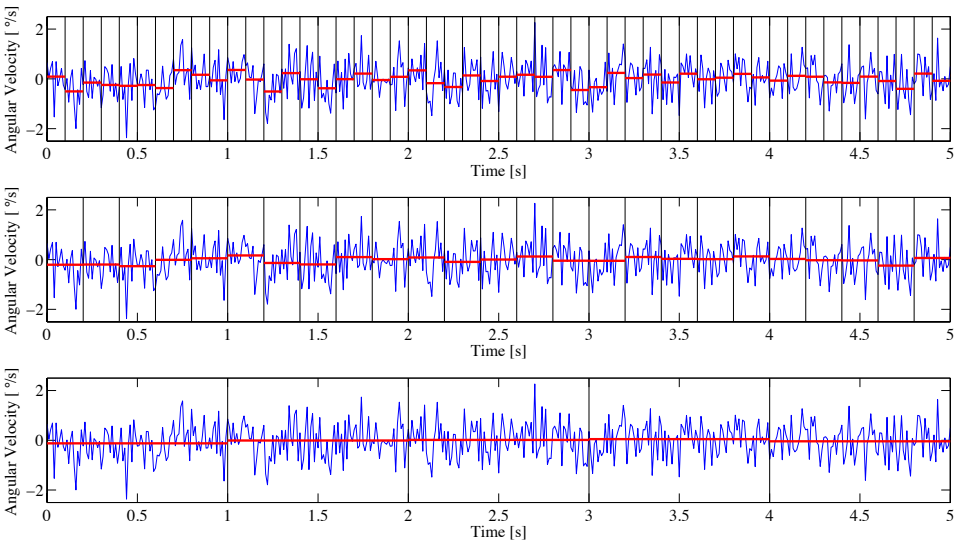

Figure D.3: Cluster averages for different correlation time - AV method

The overlapping AV estimator (D.4), is the method recommended by IEEE [13] and is therefor widely accepted method of characterizing MEMS gyroscopes. The estimator is considered superior to the estimate used in this project as it enables the use of overlapping clusters. During the preliminary work with the development of the MATLAB® implementation of the AV estimator both types of estimators where implemented and tested. The performance of the standard estimator was found to be similar to that of the overlapping estimator, the later being the most computational heavy. In order to reduce the computational load and thereby the time necessary for the calculation, the standard estimator was chosen as the main method of estimating the AV. It should be noted that higher level of accuracy in estimating the sensor process noise coefficients should be possible by using the overlapping estimator, though this was not the conclusion of the preliminary work. CDL who has extensive experience with characterizing both medium and high grade MEMS gyroscopes, use the standard estimate. The use of the standard estimate is thus considered a valid decision.

MATLAB® Implementation

In this section the algorithm implementing the AV method is described. The algorithm is implement in MATLAB® as illustrated here below. The MATLAB® function file `allan.m` containing the algorithm can be viewed on the project CD. `allan.m` is designed as a function that takes a vector of data and the sample frequency as input and supplies the τ vector and the Root Allan Variance vector $\sigma_A(\tau)$ as output.

```
1  function [sig,tau] = allan(data,Fs)
2  % function [sig,tau] = allan(data,Fs)
3  % INPUTS:       data:     N x 1 data array                        [°/s]
4  %               Fs:       Sampling Frequency F_s                  [Hz]
5  % OUTPUTS:      sig:      Standard Root Allan Variance σ_A        [°/s]
6  %               tau:      Cluster time τ                          [s]
7
8  tau0 = 1/Fs; N=length(data);            % Sampling period and Length of data set
9  max_clusternumber = floor( log((N-1)/3)/log(2) );
10 tau = zeros(1,max_clusternumber+1);
11 sig = zeros(1,max_clusternumber+1);
12 for j=0:max_clusternumber
13     M = 2^j;                            % Cluster-size being evaluated
14     tau(j+1)=M*tau0;                    % Cluster time τ
```

```
15      index = 0;                          % Index is the d vector index
16      d=zeros(1,floor(N/M));              % Cluster difference vector
17      for i=1:M:N-M+1
18          index = index + 1;
19          d(index)=sum(data(i:i+M-1))/M;  % The average for a cluster, size m
20      end
21      sig(j+1)=sqrt(0.5*mean((diff(d(1:length(d))).^2)));
22  end
23  end
```

The above implementation of the standard AV estimator is used throughout the project. Below is a implementation of the overlapping AV estimator which has not been used in the project, as it was found be be too computational heavy and not supply results significantly better than the standard estimator.

```
1   function [sig,tau] = overallan(data,Fs)
2   % function [sig,tau] = allan(data,Fs)
3   % INPUTS:       data:     N x 1 data array          [°/s]
4   %               Fs:       Sampling Frequency, Fs    [Hz]
5   % OUTPUTS:      sig:      Overlapped Root Allan Variance, σA   [°/s]
6   %               tau:      Cluster time, τ            [s]
7
8   tau0 = 1/Fs; N=length(data);            % Sampling period and Length of data set
9   max_clusternumber = floor( log((N-1)/3)/log(2) );
10  tau = zeros(1,max_clusternumber+1);
11  sig = zeros(1,max_clusternumber+1);
12  for j=0:max_clusternumber
13      M = 2^j;                            % Cluster-size being evaluated
14      tau(j+1)=M*tau0;                    % Cluster time τ
15      index = 0;                          % Index is the d vector index
16      d=zeros(1,floor(N/M));              % Cluster difference vector
17      for i=1:M:N-M+1
18          index = index + 1;
19          d(index)=sum(data(i:i+M-1))/M;  % The average for a cluster, size m
20      end
21      diff1=D(M+1:N+1-M);
22      diff2=D(1:N+1-2*M);
23      s=sum((diff1-diff2).^2);
24      sig(j+1)=sqrt(s/(N+1-2*M)/2);
25  end
26  end
```

Having presented the used method for estimating the AV and the used implementation the method of identifying noise processes using the AV is presented.

Root Allan Variance plot

Having presented the method of implementing the AV calculation the next step is to use the Root Allan Variance plot to identify the magnitude and nature of the noise processes contained in the captured data. This is accomplished by plotting the Root Allan Variance, $\sigma_A(\tau)$ as a function of correlation time τ in a loglog plot. The Root Allan Variance plot is thus defined as a plot whose x axis is $x = log(\tau)$ and whose y axis is $y = log(\sigma_A(\tau))$. An often used spacing of theτ values is $M = 2^j$ where j = 0,1,2,3... .

By examining the slope of the Root Allan Variance plot curve, it is possible to identify and determine the magnitude of the different noise processes present in the captured data. In figure D.4 a standardized Root Allan Variance plot, containing several different noise processes and their appurtenant curve slope is illustrated. The figure is a standardized representation of how processes are represented in the Root Allan Variance plot and should be interpreted as such. When using actual data sampled from a real sensor the slopes and transitions may not be as sharp as represented i the figure.

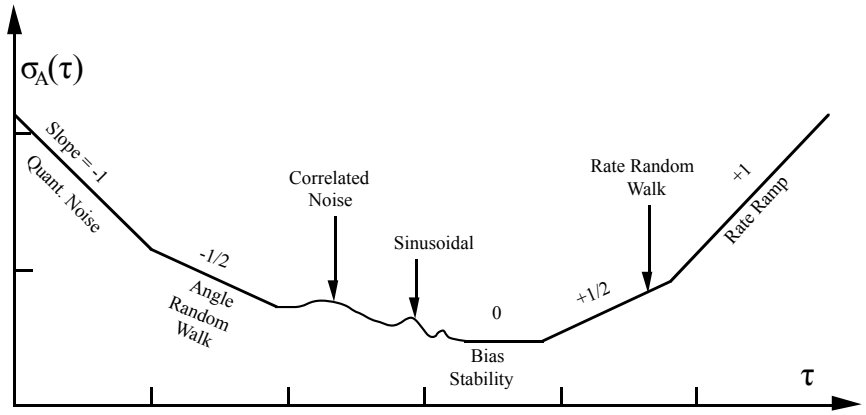

Figure D.4: Standardized Root Allan Variance plot after [13, p. 71] - AV method

The contributions from the different noise processes are highly dependent on the number of samples in the data set as well as the duration of the measurement. Considerations in relation to what type of noise sources that are to be examined are vital for the usefulness of the plot. Such consideration should not be taken lightly as sampling sessions can have duration of several days. The typical cluster time τ range for computation of the random processes are several minutes for ARW, several hours for bias stability and several days for RRW.

The connection between the random processes and the slope of the Root Allan Variance plot curve can be inferred from the AV expressed in the frequency domain. The AV as expressed in the frequency domain is given as follows [13].

$$\sigma_A^2(\tau) = \frac{1}{2}\mathrm{E}\left[\left[\bar{z}_{k+1}(M) - \bar{z}_k(M)\right]^2\right] = 4\int_0^\infty S_\Omega(f)\frac{\sin^4(\pi f\tau)}{(\pi f\tau)^2}\,df \tag{D.5}$$

Where τ is the correlation time and $S_\Omega(f)$ is the two-sided Power Spectral Density (PSD) of the gyroscope output. It should be remembered that the two-sided ratePSD is only applicable for stationary random processes. One interpretation of (D.5) is, that the AV is proportional to the total noise power of the gyroscope output when passed through a filter of the form

$$h(x) = \frac{\sin^4(x)}{(x)^2} \tag{D.6}$$

The filter is the result of the method used to cerate and operate on the clusters. The the filter bandpass depends on the correlation time τ and therefor different types of random processes can be examined by changing τ. In effect this variation of the filter bandpass is exactly what is through the AV method. In the following sections the physical origin of various noise processes are described along with their characteristics when plotted in the Root Allan Variance plot.

Angle Random Walk

The ARW is one of the main sources of error in angle estimation using gyroscopes. The output of a gyroscope is typically a signal proportional to the measured rotational velocity about the gyroscopes sensitive axis. As the gyroscope is at rest the assumption is that the output will be zero but due to white noise in the output the will output will not be exactly zero but vary between being above and below zero. The effects of this white noise in terms of limited accuracy, may be acceptable if angular velocity is indeed the state of interest. However most often the angle and not the angular velocity is the state of interest and this renders the

effects of the white noise more noticeable. The process of integrating the gyroscopes output accumulates the white noise and results in drift in the measured angle. In this way ARW is directly applicable to angle calculations as it describes the average deviation which occur when the gyroscope output is integrated. The error will increase indifferently over time and provides a fundamental limitation to any angle measurement that relies solely on integration of the measured output.

The PSD associated with the ARW can be described as follows [13].

$$S_\Omega(f) = ARW_C^2 \tag{D.7}$$

where ARW_C is the measurement noise coefficient. Substituting (D.7) into (D.5) and performing the integration yields

$$\sigma^2_{A,ARW}(\tau) = \frac{\mathrm{ARW}_C^2}{\tau} \tag{D.8}$$

from which it can be inferred, that the Root Allan Variance associated with the ARW noise will be identifiable by a slope of $-1/2$ when plotted in the Root Allan Variance plot. The numerical value of ARW_C in units of $[^\circ/\sqrt{s}]$ can be extracted from the Root Allan Variance. ARW_c is the variance coefficient at $\tau = 1s$ which intersects the $-1/2$ slope appurtenant to the ARW noise process. The process of extracting the numerical value of ARW_C will be addressed later in this section.

Rate Random Walk

The RRW noise process is a noise process of uncertain origin. The RRW noise process manifests itself as a slow change in the gyroscope bias. The PSD associated with the RRW noise process is as follows

$$S_\Omega(f) = \left(\frac{\mathrm{RRW}_C}{2\pi}\right)^2 \frac{1}{f^2} \tag{D.9}$$

Where RRW_C is the RRW noise process coefficient. Substitution of (D.9) into D.5 and performing the integration yields

$$\sigma^2_{A,RRW}(\tau) = \left(\frac{{\mathrm{RRW}_C}^2}{3}\right)\tau \tag{D.10}$$

from which it can be inferred, that the Root Allan Variance associated with the RRW noise will be identifiable by a slope of $+1/2$ when plotted in the Root Allan Variance plot. The numerical value of RRW_C in units of $[^\circ/\sqrt{s^3}]$ can be extracted from the Root Allan Variance. RRW_c is the variance coefficient at $\tau = 3s$ which intersects the $+1/2$ slope appurtenant to the RRW noise process. The process of extracting the numerical value of RRW_C will be addressed later in this section.

Bias Stability

The output from a non calibrated gyroscope will not be centered around zero but contain some offset. This offset is normally refereed to as bias and is normally considered a deterministic value that can simply be subtracted from the gyroscope output to center the output. The method of identifying the bias is normally to sample the gyroscope output for a prolonged period of time and calculating the sample mean. This however is somewhat of a approximation as the bias will change both with temperature and over time. As the bias inherently drifts over time the length of the sample period will influence the value of the bias.

Considering the AV algorithm works by dividing the samples of the gyroscope output into clusters and finding the variance between cluster averages the AV method is excellent to

investigate the stability of the gyroscope bias. For short cluster time τ the variance of the cluster averages will be quite large due to ARW. As the cluster time τ is increased the variance between cluster averages decrease and the bias estimate becomes more and more accurate. At some point the Root Allan Variance plot curve will start increasing again due to RRW. The minimum point of the Root Allan Variance plot curve is thus a measure of how stable the output of the gyroscope is over time and is measured in $[^\circ/s]$. The value represents the best bias stability that can be expected, assuming that the bias is calculated based on the characteristic time τ as defined by the minimum point of the Root Allan Variance plot.

Estimation Accuracy

The Allan variance analysis is based on a finite number of samples and thus a finite number of clusters can be formed. In this way the number quality of the estimate is proportional to the number of independent clusters of a given length that can be formed. The quality of the variance estimate increases as the number of clusters are increased.

The percentage error ζ in estimating the Root Allan Variance σ_A when clusters containing M data points from a data set with N collective points can be calculated as follows

$$\zeta = \frac{1}{\sqrt{2\left(\frac{N}{M} - 1\right)}} \tag{D.11}$$

The equation can be used in the process of designing the data acquisition method. Consider a data set containing 100,000 data points. If cluster sizes of 10,000 data points are used the percentage error in estimating $\sigma_A(10,000s)$ is approximately 23.5%. On the other hand the error in estimating $\sigma_A(1,000s)$ is approximately is only 7.1%. In this way it can be seen that uncertainty grows with the correlation time τ. A rule of thumb, a minimum of 9 clusters must be formed, which corresponds to an percentage error of 25%, unless the estimate loses it meaning.

To investigate the presence of a random process, which is know to manifest at $\tau = 100,000s$, in a gyroscope output within an error of 25%, a minimum of $900,000$ seconds must be obtained. Quite obviously the required length (in time) of the data set can therefor grow to extreme lengths depending on the process to be investigated. In the example from above, the $900,000$ seconds sample length corresponds to a data set with a length of 10.4 days. As illustrated the error equation can be used as a guideline when a designing the data acquisition method. In the following section a small example on how to use the error equation to estimate the required capture length.

Simulation Data Example

In this section use of the AV method is addressed using simulated data. The objective is of this section is to provide the reader with a clear understanding of the considerations related to the use of the AV method. The decision to use simulated data enable complete control of the noise processes in the considered data set and thus clear evaluation of the results provided by the analysis.

The data used in this section is generated using the model designed in chapter 5 on page 39. In models the stochastic characteristics of the project gyroscope ADIS16265 [3] and features the ARW and RRW noise processes. In the following the typical values for $\text{ARW}_{C,TYP}$ and $\text{RRW}_{C,TYP}$, as identified in chapter 5 has been used.

$$\text{ARW}_{C,TYP} \triangleq 0.068^\circ/\sqrt{s} \qquad \text{and} \qquad \text{RRW}_{C,TYP} \triangleq 0.00029^\circ/\sqrt{s^3} \tag{D.12}$$

The first objective is to evaluate the required data length (in time) which provides a decent level of accuracy when estimating the noise processes. In this example quite a lot of information regarding the expected noise processes, are present and the overall shape of the Root Allan Variance plot can therefor be estimated. As only the ARW, RRW noise have been modeled, slopes of -1/2 and 1/2 will will dominate of the Root Allan Variance plot. The expected shape of the Root Allan Variance plot is illustrated in figure D.5.

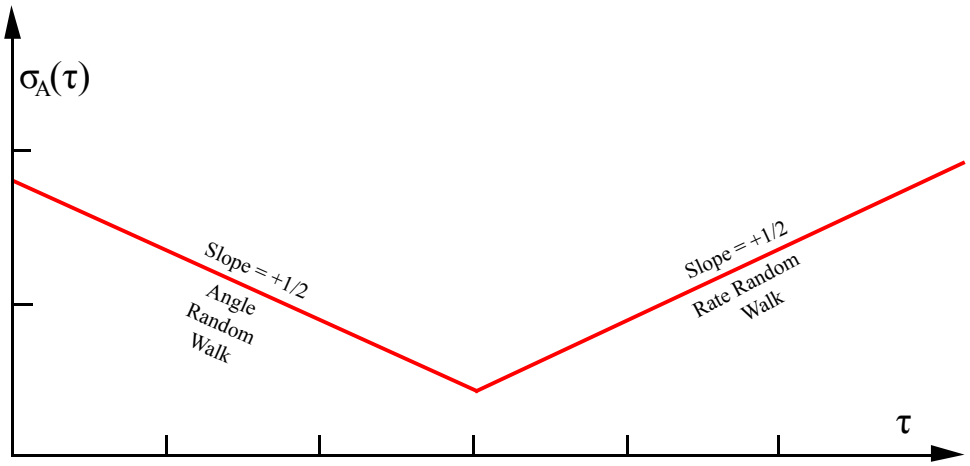

Figure D.5: Expected Root Allan Variance Plot - AV method

From figure D.5 it is quite clear, that the RRW is the process that manifests at the highest τ value. This means that if the RRW process is to be estimated with a specific error percentage, the required capture length can be inferred from this error percentage using equation (D.11). In this example the actual value of the RRW noise process is know and the required capture length could thus be calculated directly. However in most situations only guiding information regarding the noise processes are present. Therefor this example as if nothing where known about the noise processes in the captured data.

From literature [13], it is know that the RRW noise process, in medium grade MEMS gyroscopes, typically manifests at cluster time of 10,000 seconds ($\tau(10,000s)$). Along with the desired estimation error percentage this somewhat qualified guess can be used to estimate the required capture length. In this example the estimation error percentage is set to 20%. The required estimation length can be estimated using the following equation

$$20\% = \frac{1}{\sqrt{2\left(\frac{N}{10{,}000\text{S}} - 1\right)}} \Rightarrow N = 13.5 * 10000\text{s} = 37.5\text{h} \tag{D.13}$$

The above equation describes that the theoretically required session length (in time), which ensures that a percentage error of 20% in estimating random process with characteristic time of 10,000 seconds is at least 37.5 hours. In order to have some room for inaccuracies in the measurements and be well on the correct side of 20% the session length is set to 48 hours.

The simulated sensor is sampled at 100Hz for 48 hours i.e. N = 17.280.000 samples. In the figure D.6, a 10 second sample of the simulated data is illustrated.

The MATLAB® file `simallanexample.m` script used to generate and process the data can be viewed on the project CD. In figure D.7 the Root Allan Variance plot related to the simulated data is illustrated.

From figure D.7 it is clear that the estimated noise coefficients are extremely close to those used in the simulation. Figure D.7 shows, that the AV analysis method is a valid tool in relation to identifying and characterizing random noise processes in gyroscope data. In can also be inferred that the MATLAB® script used to plot and identify the value of the noise

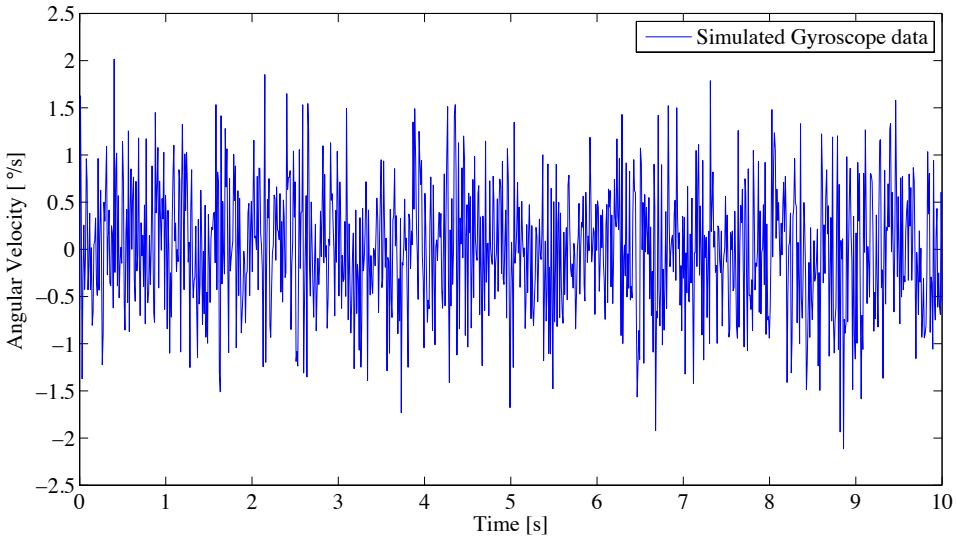

Figure D.6: 10 seconds of simulated data - AV method

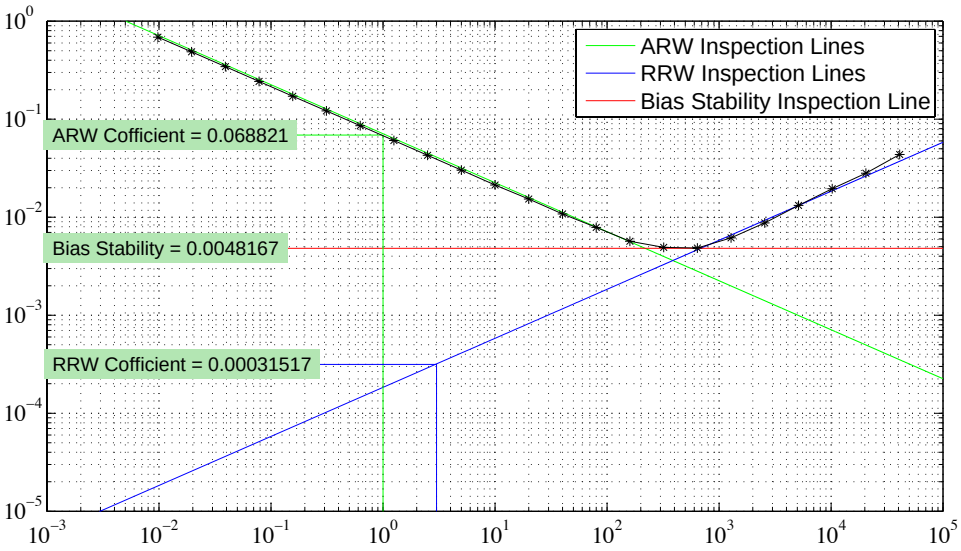

Figure D.7: Root Allan Variance plot of simulated data - AV method

process coefficients is sufficient. The script requires that the user manually reads of the Root Allan Variance plot and the accuracy of the coefficient is thus partially determined by the users ability to correctly read the chart. This means of implementation, based on the simulation results, does not seam to induce high levels of inaccuracy.

Appendix Conclusion

In this appendix the AV analysis method has been described and illustrated. The AV algorithm and implementation in MATLAB® has been presented in detail as well as the mathematical connection between the AV method and the PSD. The validity of the AV method has been substantiated by simulating the output from the project gyroscope containing both ARW and RRW noise processes. The simulated data has been processed using the AV algorithm MATLAB® implementation and the identified noise processes where found to be identical to those used in the process of generating the simulation data. To this extent the

implementation of the AV method is considered valid.

Appendix E

Measurement Noise Analysis

In this appendix the characteristics of the gyro-board output signal is analyzed in detail. The objective of this appendix is to evaluate the characteristics of the gyro-board measurement noise.

Data Acquisition

The test setup used for this sample session is quite crude in design. The main objective of the appendix is to capture of the gyro-board output signal for a short period of time. The test setup consists of the gyro board being placed on a damped base as illustrated in figure E.1.

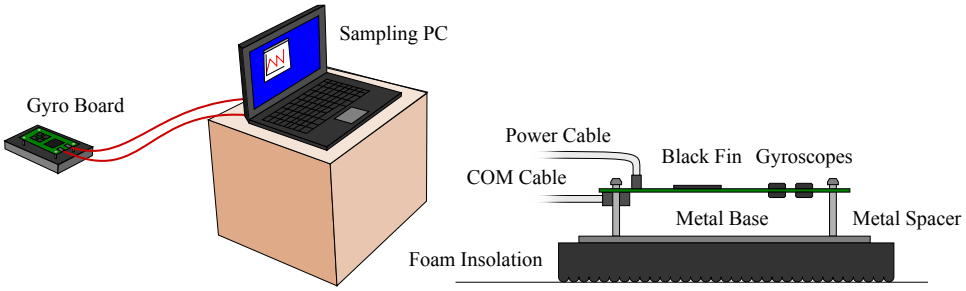

Figure E.1: Measurement noise Test setup - AAU

The test was performed in the UAV Lab at Aalborg University. The following is a step-by-step description of the test procedure.

1. Turn on Sampling PC
2. Connect data and power cables to gyro-board
3. Connect data and power cables to Sampling PC
4. Securely position the gyro-board on the foam isolation pad.
5. Start the gyro-board sampling software on the Sampling PC
6. Verify that the gyro-board is providing samples by taking a few samples
7. Initiate the measurement by requesting two minutes equivalent to 12,288 samples from the sampling software prompt

8. Wait two minutes
9. Verify the that the requested amount of samples has been sampled by looking at sampling software prompt
10. End of test session

Session Data

In this section the raw results from the capture session preprocessing are presented. The preprocessing implements both evaluation of the sampled data and unit conversion from bits to $^\circ/s$.

The data files have been processed using the script file `datafilt.m` as described in appendix A on page 99. By applying the script to the acquired data a measure of the data quality is obtained. The result of the processing was the no samples where removed from either capture session indicating that no faulty samples where present in the data, and that no data packets where lost. In figure E.2 the first second of the data captured on the 14 April is illustrated.

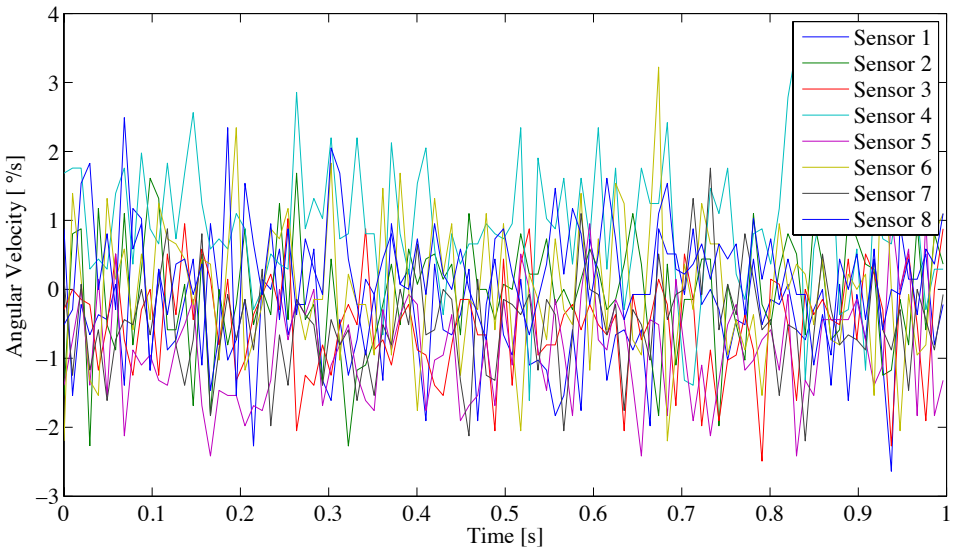

Figure E.2: Gyro-board output 14 April - AAU Test

In the following the data processing related to the captured data is presented.

Data processing

In the following management of the captured data and the results of the processing are presented.

Measurement Noise basic properties

In this section the basic properties the gyro-board measurement noise is investigated. The basic properties of the noise is the sample mean, standard deviation, peak value and root mean square value. The MATLAB® file `signalcharacteristics.m` implements the computation of all the following parameters and can be found on the project CD.

Sample Mean

The sample mean taken on a data set provides an unbiased estimate of the mean of the data set. Given a set of sampled angular rate values from the gyro-board, i.e. $\{z_i(1), \ldots, z_i(N)\}$ the sample mean $\bar{z_i}$ can be calculated as follows

$$\bar{z_i} = \frac{1}{N} \sum_{k=1}^{n} z_i(k) \tag{E.1}$$

where $\bar{z_i}$ is the sample mean for the individual gyroscopes in the gyro-board. MATLAB® implements equation (E.1) through the `mean()` function, that requires a vector of data and supplies the sample mean of that vector. In this case the vector is a column vector containing the angular rate samples of the eight individual gyroscopes in the gyro-board.

Standard deviation

The standard deviation σ is a measurement of variability that provides information on how much the measurement values varies relative to the sample mean. A low standard deviation indicates that the majority of sample points tend to be close to the sample mean and a high standard deviation indicates the sample points are spread out over a larger range. Given a set of sampled angular rate values from the gyro-board, i.e. $\{z_i(1), \ldots, z_i(N)\}$ the standard deviation σ_i can be calculated as follows

$$\sigma_i = \sqrt{\frac{1}{N} \sum_{k=1}^{N} (z_i(k) - \bar{z_i})^2} \tag{E.2}$$

where σ_i is the standard deviation for the individual gyroscopes in the gyro-board. MATLAB® implements equation (E.2) through the `std()` function that requires a vector of data and supplies the standard deviation of that vector. In this case the vector is column vectors containing the angular rates from the eight gyroscopes.

Peak values

The peak values are the highest and and lowest values contained in the data set. Essentially maximum and minimum observed values in the data set. MATLAB® implements a method of finding the maximum and minimum values of a given data set. The maximum and minimum can be found using the functions `max()` and `min()` respectively. Both functions require a vector of data and provides the maximum and minimum of this vector respectively.

Root Mean Square value

Root Mean Square (RMS) value is a statistical measure of the magnitude of a varying quantity and is often used when the data set contains both positive and negative values. Given a set of sampled angular rate values from the gyro-board, i.e. $\{z_i(1), \ldots, z_i(N)\}$ the RMS value $RMS(z_i)$ can be calculated as follows

$$RMS(z_i) = \sqrt{\frac{z_i(1)^2 + z_i(2)^2 + \ldots + z_i(n)^2}{n}} \tag{E.3}$$

$$= \frac{\sqrt{z_i(1)^2 + z_i(2)^2 + \ldots + z_i(n)^2}}{\sqrt{n}} \Rightarrow \frac{\|z_i\|_2}{\sqrt{n}} \tag{E.4}$$

MATLAB® does not directly implement calculation of the RMS value. MATLAB® does however implement a function, that enable the computation of the RMS value using equation (E.4). The 2-norm or Euclidean norm is implemented in MATLAB® through the function `norm()` that if no additional fields are supplied provide the 2-norm of the input vector.

Results

In table E.1 the collective results in relation to the basic properties of the measurement noise $v(t)$.

Gyroscope	Mean	Std	Max	Min	RMS
1	-0.56031	0.79799	2.75811	-3.17595	0.79796
2	0.10571	0.81730	3.04447	-3.62219	0.81861
3	-0.46951	0.76365	3.39991	-3.12023	0.81865
4	0.84696	0.91002	3.91494	-3.41106	0.91143
5	-0.90311	0.77111	3.02765	-2.61337	0.91143
6	0.24428	0.94069	3.27220	-3.17468	0.94100
7	-0.52403	0.75907	2.72183	-3.21223	0.94101
8	0.36220	0.79537	2.93450	-3.14608	0.94106
Mean value		0.81940			0.88514

Table E.1: noise analysis results

Distribution Properties of Measurement noise

In this section the distribution and spectral properties the measurement noise $v(t)$ are investigated. These properties include the distribution of the data, sample autocorrelation function and Fast Fourier Transform (FFT).

Histogram

Histograms plot the frequency of values in a data set and be used to estimating the probability density function of the underlying process. Based on the appearance of the histogram conclusion about the distribution of the sampled process can be made. Most recognizable is the the normal (or Gaussian) distribution, that result in a bell-shaped histogram indicating that the distribution is indeed Gaussian.

MATLAB® implements a method from generating the histogram through the function `histfit()`. The function function requires a vector of data and supplies the histogram for the data set. The function has a optional field that describe the number of discrete intervals (bins) that are used in the histogram. A large number of bis provides a smoother histogram. The function in addition fits a standard Gaussian density function to the data. The resulting histogram for the three sessions are illustrated in figures E.3 - E.4. The approximated mean has been subtracted from the data set so the data cluster around zero instead of the real mean value.

From figure E.3 - E.4 it can be seen that the distribution of the sampled data approximately resembles a Gaussian distribution. This implies that the measurement noise $v(t)$ can be assumed to be Gaussian noise.

Sample Autocorrelation Function

Correlation is the mutual relationship between two or more observed random variables. The sample autocorrelation on the other hand refers to the correlation of an observed random variable with its own past and future values. The sample autocorrelation is a useful method of finding hidden repeating patterns in a signal, such as determining the presence of a periodic signal which has been buried under noise, or identifying the fundamental frequency of a signal.

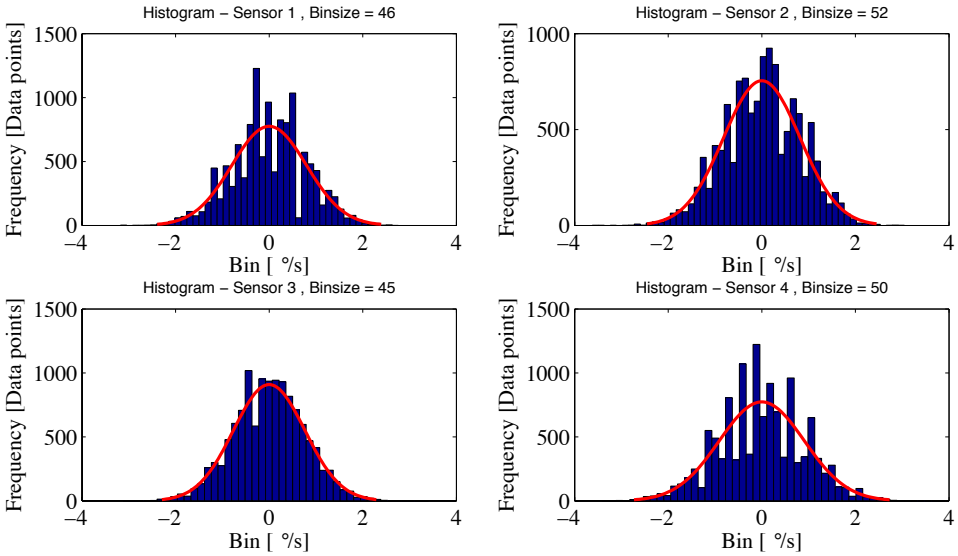

Figure E.3: Sensor 1 - Sensor 4 Histogram 14 April - AAU Test

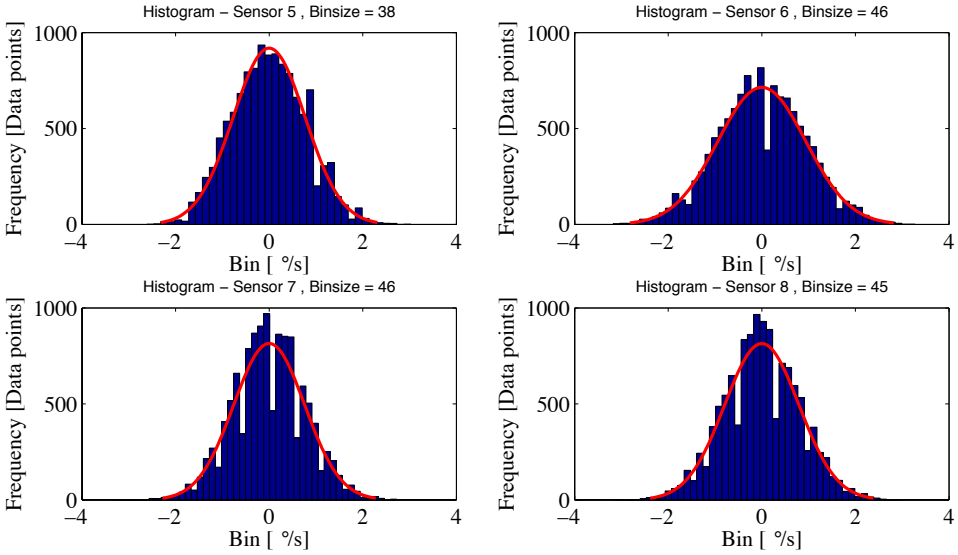

Figure E.4: Sensor 5 - Sensor 8 Histogram 14 April - AAU Test

Given a set of sampled angular rate values from the gyro-board, $\{z_i(1), \ldots, z_i(N)\}$ an important measure is the sample autocorrelation coefficients, which describe the correlation between samples points at with different separation in time. A set of autocorrelation coefficients arranged as a function of separation time (lag) is called the Auto Correlation Function (ACF). The plot of the ACF as a function of lag is also called a correlogram. From the correlogram it is possible to investigate whether the data points are random or not. The autocorrelation coefficients should be near-zero for randomness for all time lags other than zero.

MATLAB® implements a method for calculating the autocorrelation through the function `autocorr()`. Different methods of calculating the ACF exist and the method implemented by `autocorr()` works by first removing the sample mean of the input vector and then normalizing the sequence such that the ACF at lag zero is unity. The means of calculating the ACF implemented by `autocorr()` is as follows.

$$r_i(k) = \frac{c_i(k)}{c_i(0)} \qquad c_i(k) = \frac{1}{n}\sum_{t=1}^{n-k}\left(z_i(t) - \bar{z}_i\right)\left(z_i(t+k) - \bar{z}_i\right) \tag{E.5}$$

Where $r_i(k)$ is the correlation coefficient for gyroscope at lag k, $c_i(k)$ and $c_i(0)$ is the auto covariance at lag k and zero respectively. The auto covariance at lag zero is the variance. In figures E.5 - E.6 the resulting correlograms for the data are illustrated.

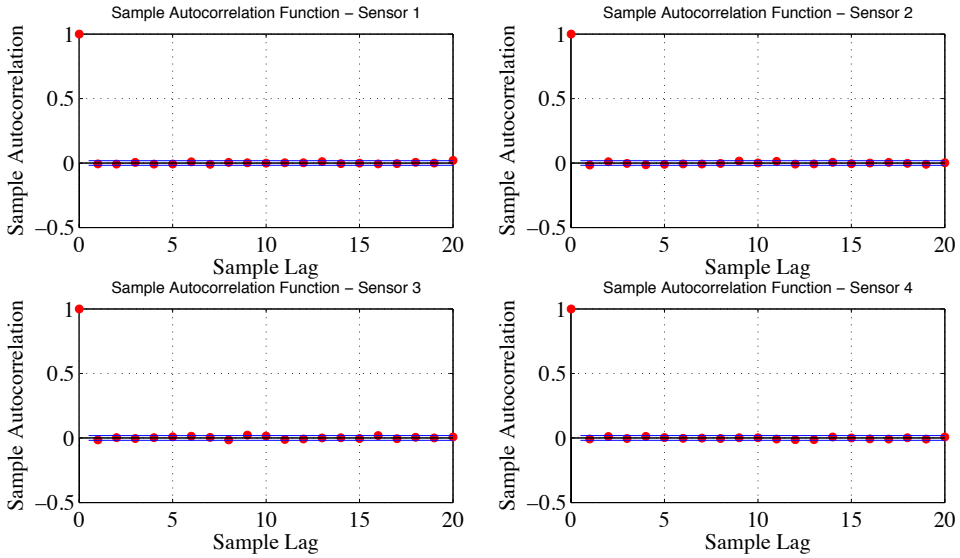

Figure E.5: Sensor 1 - Sensor 4 Correlograms 14 April - AAU Test

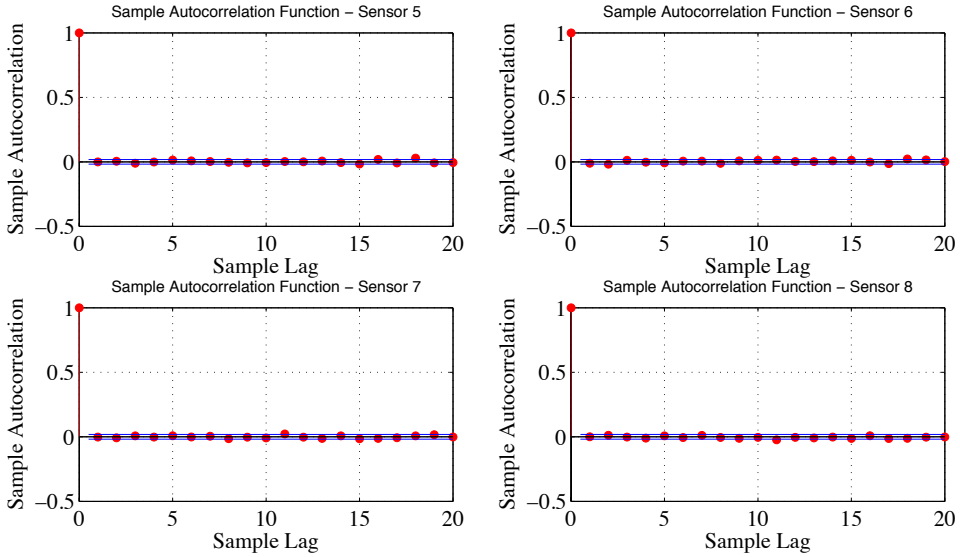

Figure E.6: Sensor 5 - Sensor 8 Correlograms 14 April - AAU Test

If the measurement noise is indeed white noise the individual samples muse be uncorrelated in time. This means that the exception is that $r_i(k) = 0$; for $k < 0$. The 95% confidence limits for the correlogram can calculated as $-1/n \pm 2/\sqrt{n} \approx \pm 2/\sqrt{n}$. As the sampled data consists of 12,288 samples the confidence limits are [-0.18;0.18]. From figures E.5 - E.6 it can be seen that for $r_i(k) \approx 0$; for $k < 0$.

Fast Fourier transform

The FFT is a method of rapidly calculating the Discrete Fourier Transform (DFT) of a data series. The DFT describes the relations between a sampled signal with a finite number of points in the time domain and a sampled spectrum with a finite number of points in the frequency domain. The DFT decomposes series of data points into components of different frequencies supplying the datasets frequency spectrum. The FFT computes the DFT and produces exactly the same result as evaluating the DFT, the only difference is that the FFT is much faster. MATLAB® implements a method for calculating the FFT through the function `fft()` that uses a Cooley-Tukey algorithm.

The amplitude spectrum of a sampled data set is the absolute value of it's Fourier transform. The amplitude spectrum of white noise is constant over a broad frequency band and should thus have a flat frequency spectral density, which means the magnitudes should be equal and constant on each frequency components. To visualize the DFT, the absolute values of the DFT is plotted and a periodogram that plots the absolute value of the Fourier transform. The first half of the frequency range, from 0 to the Nyquist frequency $F_s/2$), is sufficient to identify the component frequencies in the data, since the second half is just a reflection of the first half.

```
1 Fs = 50;                                          % Sample frequency [Hz]
2 y = abs(fft(data));                               % Amplitude spectrum
3 f = (0:length(data)/2-1)*(Fs/length(data));       % Frequency range
4 plot(f,y(1:length(data)/2))
```

In figures E.7 - E.8 the resulting periodograms for the data are illustrated.

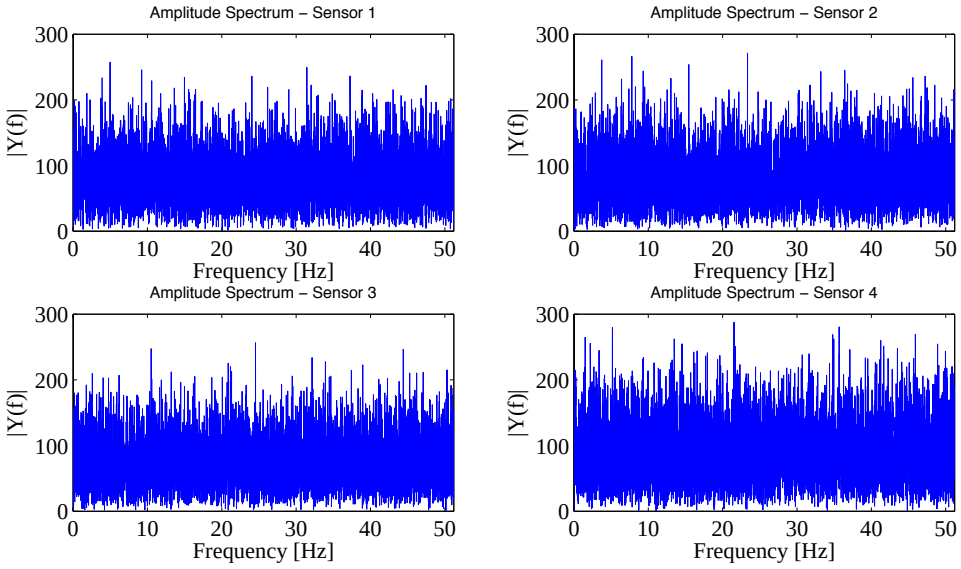

Figure E.7: Sensor 1 - Sensor 4 Amplitude Spectrum 14 April - AAU Test

From figures E.7 - E.8 is can be seen that the amplitude spectrum for the captured data is quite flat in all frequencies. This is typical for a signal containing mostly white noise, as white noise contain an equal amount of all frequencies.

Appendix Conclusion

In this section the characteristics of the gyro-board measurement noise $v(t)$ have been investigated. In the first part of the appendix the sample mean, sample standard deviation,

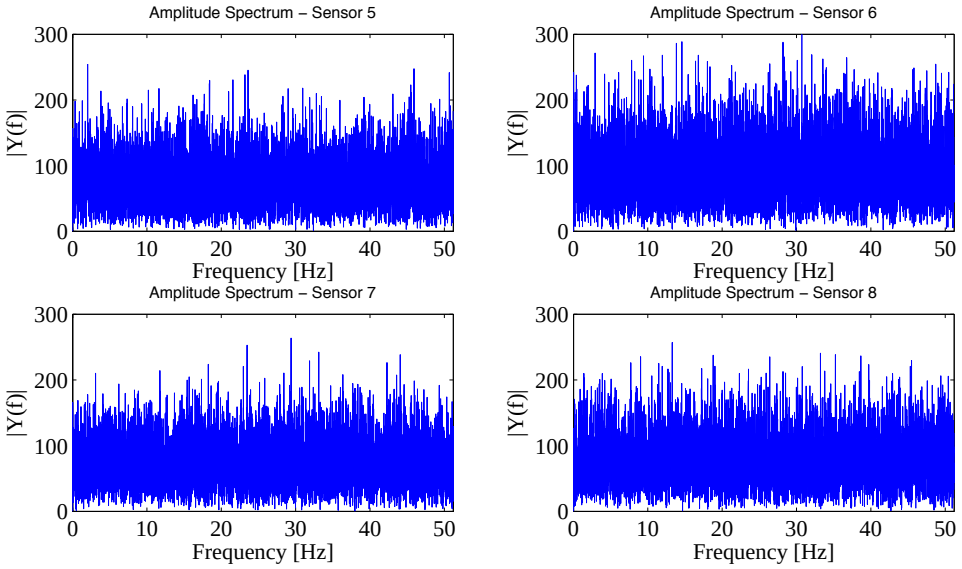

Figure E.8: Sensor 5 - Sensor 8 Amplitude Spectrum 14 April - AAU Test

minimum and maximum values where identified. In the second part of the appendix the distribution properties of the data where investigated as well as the correlation properties and spectral amplitude of the data.

The measurement noise $v(t)$ was found to have a approximate Gaussian distribution, a fairly flat amplitude spectrum, and only weakly correlation between adjacent samples indicating that the measurement noise $v(t)$ may be approximated as Gaussian white noise.

Appendix F

Allan Variance Noise Analysis

In this appendix the random noise processes of the project gyroscopes are characterized using the AV method. The appendix presents both the results of the AV analysis and the process of acquiring the necessary data. This includes all relevant considerations and the used test setup.

Data Acquisition

The section describes the process of acquiring the necessary data, including the relevant considerations and the used test setup.

Considerations

The objective of this test is to capture data from the project gyroscopes for a prolonged period of time. As presented in appendix D the error ζ, describes the percentage error in estimating the Root Allan Variance σ_A when clusters containing M data points from a data set with N collective points are considered.

$$\zeta = \frac{1}{\sqrt{2\left(\frac{N}{M} - 1\right)}} \tag{F.1}$$

Using this equation it is possible to estimate the session length (in time) required to achieve a specific error percentage. Based on figure B.1 the required sample session length can be estimated. From figure B.1 it can be seen that the Root Allan Variance curve increases after cluster times of approximately 1000 seconds. As the RRW noise process is not identifiable form the figure, it can be inferred that a significantly higher cluster time is required to adequately estimate the RRW noise coefficient. A somewhat qualified guess wold be that the RRW noise will manifest at a cluster time of 10000 seconds ($\tau = 10000$). As the cluster time of interest has been identified, the last unknown parameter in equation (F.1) is the desired estimation error percentage.

In this project the estimation error percentage is set to 20% as several identical gyroscopes are considered during the test. As several Root Allan Variance plots of identical gyroscopes will be generated it is assumed that the any errors in the estimate will be evident. The

required estimation length can be estimated as follows

$$20\% = \frac{1}{\sqrt{2\left(\frac{N}{10000} - 1\right)}} \Rightarrow N = 13.5 * 10000 = 37.5 \text{ hours} \tag{F.2}$$

In order to have some room for inaccuracies in the measurements and be well on the correct side of 20% the session length is set to 48 hours. The last parameter to be determined is the sample frequency.

The on-board sampling hardware of the gyro-board samples the eight MEMS gyroscopes at 2048Hz which is quite excessive in relation to long capture sessions. In relation to the AV analysis 48 hours are required in order to provide and adequate estimating of the RRW noise process. Using the standard 2048Hz sampling frequency will produce a staggering amount of data, which will complicate both the data handling and management. Therefor sampling frequency is lowered by under-sampling the original 2048Hz frequency to provide a more manageable amount of data. In relation to this appendix and under sampling factor of 20 is implemented in the gyro-board sampling software. The resulting sampling frequency F_s is thus given as follows

$$F_s = 2048 \cdot \frac{1}{20} = 102.4 \tag{F.3}$$

Using this sampling frequency F_s will result in 102.4Hz ·48 hours = $17.6947 \cdot 10^6$ samples which based on the groups experience is known to be manageable in MATLAB®.

Test-setup and Procedure

The test setup used for the capture session is quite crude in design. The test setup consists of the gyro-board being placed on a damped base as illustrated in figure F.1.

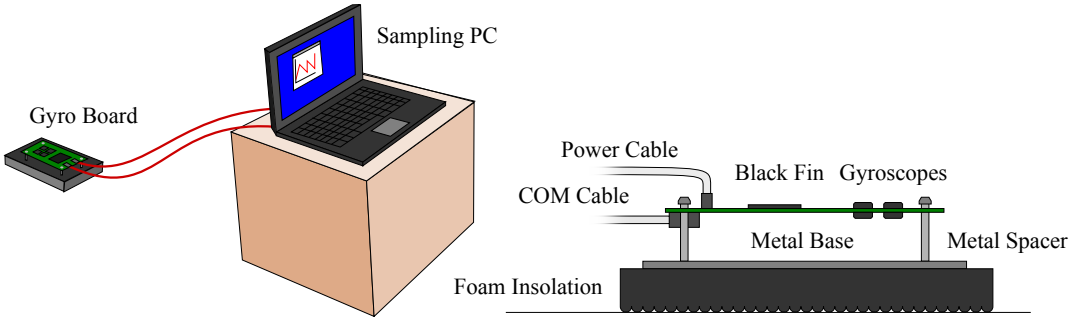

Figure F.1: AV Test setup - AAU

A total of two tests where performed to acquire two sets of data. The tests where performed in the UAV Lab at Aalborg University. The following is a step-by-step description of the used test procedure.

1. Turn on Sampling PC
2. Connect data and power cables to gyro-board
3. Connect data and power cables to Sampling PC
4. Securely position the gyro-board on the foam isolation pad.
5. Start the gyro-board sampling software on the Sampling PC
6. Verify that the gyro-board is providing samples by taking a few samples
7. Initiate the test by requesting 48 hours = 172800 samples from the gyro-board sampling software prompt

8. Leave the room and return after 48 hours
9. Verify the that the requested amount of samples has been sampled by looking at gyro-board sampling software prompt
10. End of test session

Session Data

In this section the raw results from the capture session and the results of the preprocessing are presented. In figure F.2 the first 1 second of data captured in the period 15 April - 18 April is illustrated. In figure F.3 the first 1 second of data captured in the period 22 April - 25 April is illustrated. The two data files have been processed using the script file

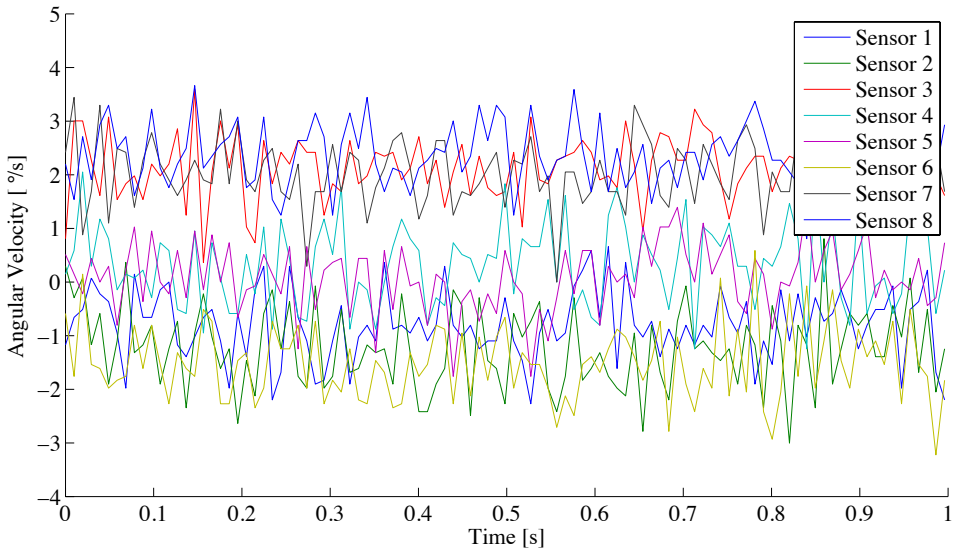

Figure F.2: gyro-board Output 15 April - 19 April - AAU Test

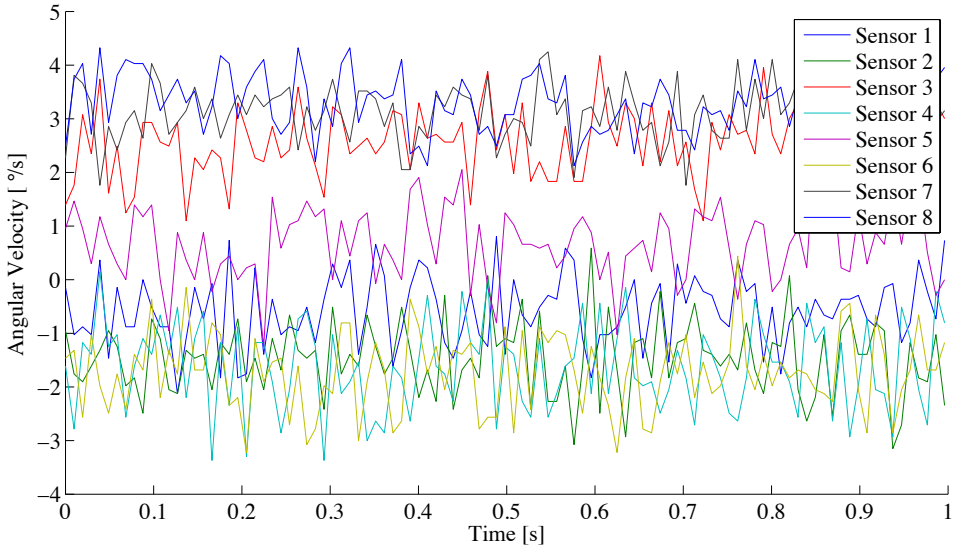

Figure F.3: gyro-board Output 22 April - 25 April - AAU Test

`datafilt.m` as described in appendix A on page 99. By applying the script to the acquired

data a measure of the data quality can be obtained. The result of the processing was the no samples where removed from either capture session, indicating that no faulty samples where present in the data, and that no data packets where lost.

During the capture sessions the temperature of the gyro-board was monitored, by sampling the internal temperature of the fourth sensor. In figure F.4 the temperature during the 15 April - 18 April capture session is illustrated. In figure F.5 the temperature during the 22 April - 25 April capture session is illustrated.

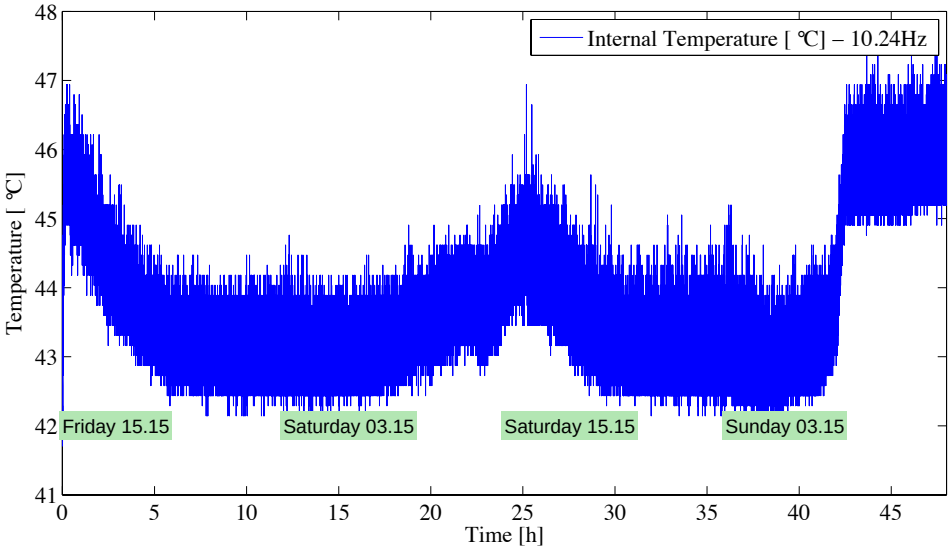

Figure F.4: Sensor 4 Temperature 15 April - 18 April - AAU Test

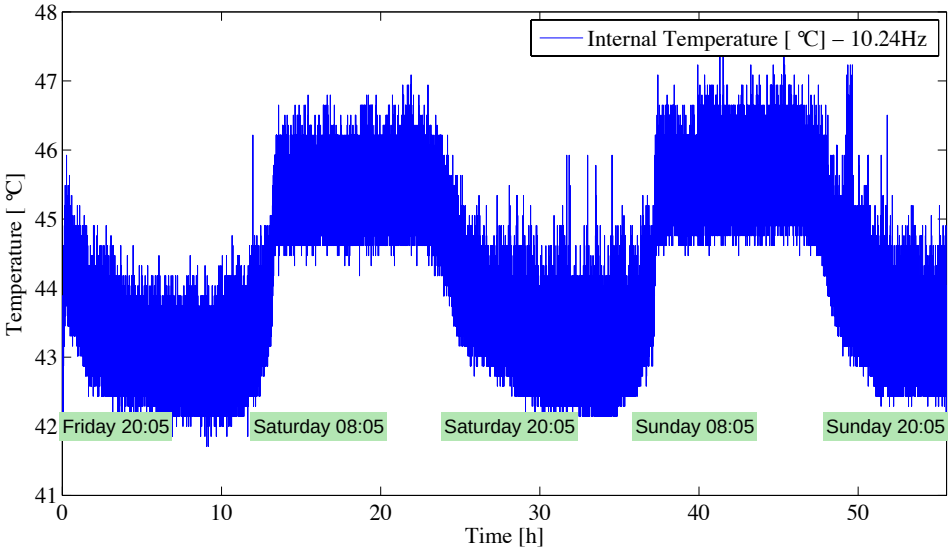

Figure F.5: Sensor 4 Temperature 22 April - 25 April - AAU Test

From figures F.4 - F.5 it can be seen that the internal temperature of the gyroscope is maintained within a narrow band. The somewhat periodical fluctuations are believed to be caused by the room temperature variation.

Data processing

In this section the acquired data is processed with the AV method MATLAB® script file `allan.m`, as described in appendix D. In figure F.6 the Root Allan Variance plot of the eight sensors, based on the 15 April - 18 April data, are illustrated. In figure F.7 the Root Allan Variance plot of the eight sensors, based on the 22 April - 25 April data, are illustrated.

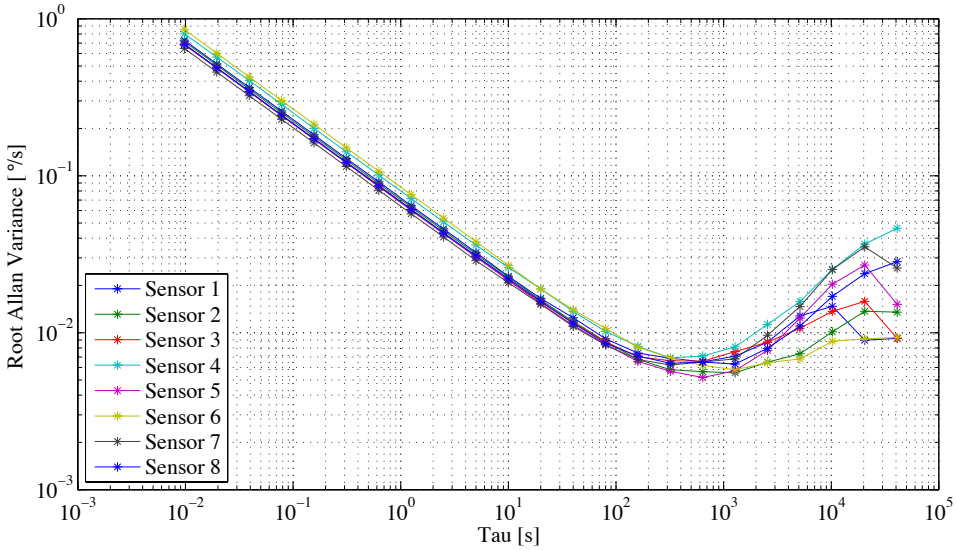

Figure F.6: Root Allan Variance, gyro-board Output 15 April-18 April - AAU Test

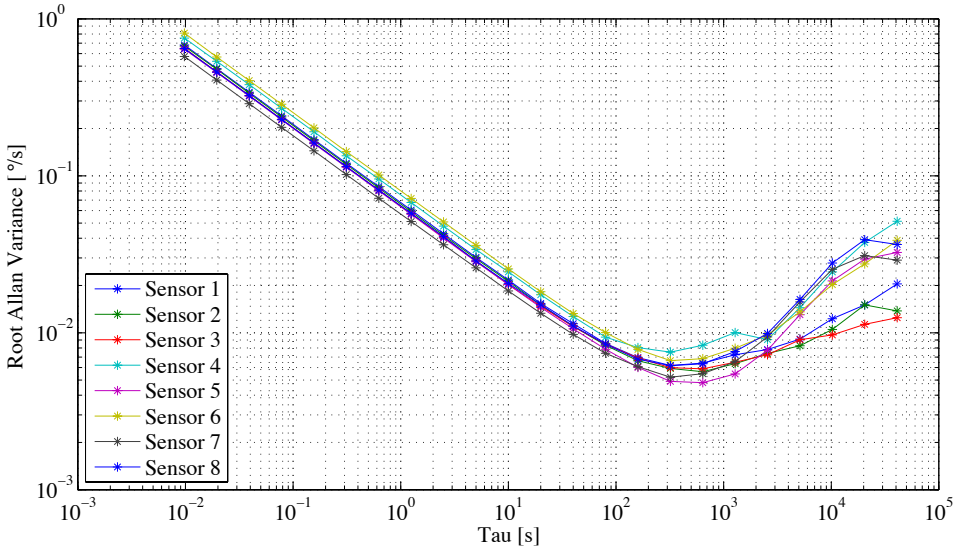

Figure F.7: Root Allan Variance, gyro-board Output 22 April-25 April - AAU Test

Figures F.6 - F.7 illustrates the collective Root Allan Variance plots. Based on the figures it is obvious that the individual sensors, as expected, has similar noise characteristics. It can also be seen that the dominant noise processes in the data are the ARW, RRW.

Using the MATLAB® script files `aauallan1.m` and `aauallan2.m` the ARW, RRW and Bias stability noise coefficients for the individual sensor have been identified. The plots used in the process of identifying the values of the noise components can be viewed on the project CD.

Appendix Conclusion

In this appendix the random noise processes in the project gyroscopes have been investigated using the AV method. The analysis was conducted using data captured from the gyro-board during two 48 hour sample sessions. The used sampling frequency was F_s =102.4Hz. The captured data was processed using a data preprocessing script and found suitable for further processing. Subsequently the data, from the individual sensors, where processed using the AV method and the individual Root Allan Variance where plotted. Based on these plots the ARW and RRW where found to be the predominant noise processes in the gyroscopes.

By inspecting the Root Allan Variance plots the noise coefficients where estimated. In addition to the ARW and RRW coefficients, the gyroscopes Bias stability where also estimated. The collective coefficients, from the two sessions respectively, are listed in table F.1 and F.2.

Gyroscope	Bias Stability $^\circ/s$	ARW $^\circ/\sqrt{s}$	RRW $^\circ/\sqrt{s^3}$
1	0.00674	0.07118	0.00032
2	0.00551	0.06882	0.00018
3	0.00674	0.06434	0.00027
4	0.00746	0.07874	0.00039
5	0.00533	0.06654	0.00028
6	0.00589	0.08422	0.00015
7	0.00631	0.06221	0.00033
8	0.00631	0.06654	0.00029
Mean value	0.00629	0.07032	0.00028

Table F.1: Noise coefficients, gyro-board Output 15 April - 18 April - AAU Test

Gyroscope	Bias Stability $^\circ/s$	ARW $^\circ/\sqrt{s}$	RRW $^\circ/\sqrt{s^3}$
1	0.00610	0.06434	0.00020
2	0.00589	0.06434	0.00019
3	0.00610	0.06434	0.00022
4	0.00746	0.07361	0.00043
5	0.00515	0.06015	0.00028
6	0.00674	0.08144	0.00034
7	0.00533	0.05437	0.00032
8	0.00631	0.06434	0.00035
Mean value	0.00613	0.06587	0.00029

Table F.2: Noise coefficients, gyro-board Output 22 April - 25 April - AAU Test

From table F.1 and F.2 it can be seen that the noise coefficients found during the two individual capture sessions are convincingly similar. It can also be noted that the sensors have almost identical noise coefficients, which is expected as the sensors are identical and operating under identical conditions. Based on the fact, that a simple method of identifying the noise coefficients i.e. `plotallan.m` has been used, the results are quite impressive. The use of the plotting tool and the inherent inaccuracy in the method induce a level of in accuracy of the estimates, but for this project the achieved values are considered adequate.

Several places in the thesis, what is considered to be typical values for the ARW and RRW coefficients are used in simulations. Based on the results presented in table F.1 and F.2 these typical values are set as follows.

$$\text{ARW}_{C,TYP} \triangleq 0.068^\circ/\sqrt{s} \qquad \text{and} \qquad \text{RRW}_{C,TYP} \triangleq 0.00029^\circ/\sqrt{s^3} \tag{F.4}$$

The values are based on the mean value for the two capture sessions and are considered to be sufficiently good estimates of the actual values. Should a higher estimate accuracy for the individual sensors be required, different sampling frequencies, longer capture sessions, and automated coefficient identification scripts should be implemented. The capture length versus sample frequency contains the inherent tradeoff of the AV method. To correctly capture the high frequency noise processes a high sampling frequency must be used, but to capture the low frequency noise processes a prolonged capture session must be used. This obviously results in large amounts of redundant data that do not provide additional information and increase data processing time.

The solution to this problem is to divide the Root Allan Variance computation into several windows, that focus on different pars of the collective plot. Using this approach, high frequency noise processes can be sampled using a high sampling frequency and low frequency processes can be sampled using a lower frequency. This approach is in contrast to the approach seen most often in literature, though it is specified as the standard way of testing rate sensing senors in the IEEE standard [12]. In this project a crude method of reading of the Root Allan Variance plot has been use with acceptable results. The use of this approach however induces inaccuracy in the results as the accuracy becomes dependent on the users ability for correctly read of plots. From a the perspective of eliminating this problem it would be advisable to develop a automated method of reading of the plots.

In order to use the noise coefficients as standard deviations for the measurement noise $v(t)$ and process noise $w(t)$ in simulations, the coefficients must be scaled. This scaling implemented as follows done as follows, and is used throughout the project simulations.

$$\sigma_{ARW} = \mathrm{ARW}_{C,TYP} \cdot \sqrt{102.4} \tag{F.5}$$

$$\sigma_{RRW} = \mathrm{RRW}_{C,TYP} \cdot \sqrt{F_s} \tag{F.6}$$

Where $\mathrm{ARW}_{C,TYP}$ and $\mathrm{RRW}_{C,TYP}$ are the noise coefficients typical values and F_s is the simulation sampling frequency. This conversion will be used throughout the project wherever the ARW and RRW coefficients are used as measures of standard deviation.

This page intentionally left blank

www.es.aau.dk

Department of Electronic Systems
Section of Automation and Control
Aalborg University
Fredrik Bajers Vej 7
DK - 9220 Aalborg Øst
Denmark
Tel: (+45) 96 35 87 02
Fax: (+45) 98 15 17 39

List of Corrections