Autonomous Submersible Robot : Sensor Characterisation and Testing

Jordan Kiro Cvetanovski

THE AUSTRALIAN NATIONAL UNIVERSITY

Robotic Systems Lab

A thesis submitted for the partial fulfillment of the requirements for the degree of

Bachelor of Engineering at
The Australian National University

Supervisors:
Samer Abdallah ,Prof. Alexander Zelinsky Dr. David Wettergreen

Australian National University
Canberra, ACT 0200 AUSTRALIA

October 2000

Typeset in Palatino by TEX and LATEX 2_ε.

Except where otherwise indicated, this thesis is my own original work.

Jordan Kiro Cvetanovski
2 October 2000

Acknowledgements

My experience at RSL was unforgettable. The place was filled with energy and teamwork. Along the way, I have received help from almost every member of the lab. Thank you to everyone for allowing me to learn so much about what engineering is all about

In particular, I would like to thank the following people:

Dr. Thomas Brinsmead - Whose help was paramount in this project. Thank you Thomas for putting up with all my silly questions and more importantly for providing intelligent answers.

Thank you to all my supervisors, Alexander Zelinsky, Samer Abdallah and David Wettergreen for allowing me to have this unique opportunity to work in this highly energetic research environment. Special mention must go to David Wettergreen who always found the time to look over my paper and thesis drafts and provide valuable feedback despite being out of the country.

To all the guys at the lab working late nights, thank you for your company during these trying times. Particular thanks go to Chris Gaskett and Tim Edwards for providing invaluable advice. I would also like to express my gratitude towards Harley Truong, Seungjun Oh and Sunil Rao with their help and assistance. My very good friends, Orson Sutherland, James Constable and Vladimir Baltovski, get a token mention for their support and friendship.

Special thanks go to my family for their invaluable support and encouragement during the past few months.

Abstract

This thesis is a part of a continual effort to develop a state estimation system for an autonomous underwater vehicle at the RSISE in the ANU. This thesis presents an experimental method to determine the dynamic characteristics of the inertial sensors that are included in the sensor suite of the AUV. The particular parameters calculated were the offset bias and scale factors of the rate gyroscope and accelerometer. A least squares estimation algorithm is proposed to provide these parameters from the available "ground truth" and sensor data. The drift characteristics of derived parameters such as velocity and position from raw sensor data are also investigated. The results indicate that the experimental method was successful in accurately calculating the bias and scale factor parameters. Results from static experiments with the sensors are also presented.

Considerable advance has also been made in developing the necessary algorithms necessary to complete the software aspect of the state estimation system. Preliminary tests on these algorithms have also been carried out.

Contributions

As part of my project, over the last 4 months I have completed the following works.

- Successfully designed and implemented an experimental method for dynamic testing of the inertial sensor parameters that are crucial for a state estimation system. Investigation was also carried out on the drift properties of the sensor model state estimates. This testing provided considerable in-roads into the full understanding and modelling of the inertial sensors.
- A least-squares method was proposed for acquiring the scale factor and bias values for the inertial sensors. This method was proven to be successful in calculating these values.
- Extensive knowledge was gained in both signal processing techniques and robot kinematics in order to obtain the desired results
- Helped identify and correct the errors (both in the software and hardware) of the state estimation system components. This was due to both static and dynamic testing and analysis of the observed sensor behaviour.
- The software necessary for running the Extended Kalman Filter was completed. The top level state estimation system module was written. Initial testing of the EKF on the WAM experiments has been initiated.
- Background theory of the state estimation system for the Kambara and some of the results presented in this thesis have been published in a paper that was accepted for the Australian Conference of Robotics and Automation '99

Contents

Chapter 1

Introduction

With the increasing intensity in allocation of resources to space exploration, the need has developed for a better understanding and more effective preservation for something that is closer to earth, our oceans. Covering 70 % of the Earths surface, the ocean is still a relatively under-utilised resource. It is well documented that the oceans hold vast food, energy and mineral resources as well as affecting weather patterns. However, we still do not know enough about our oceans to be able to utilise or more importantly preserve these natural resources. The need for preservation is evident closer to Australia's shore in places such as the Great Barrier Reef, off of Australias NorthEast coast. These vast areas of coral reef are showing evidence of the negative impact stemming from increasing tourism and destructive sea-life such as the Crown of Thorns Starfish.

Over the last two years, the Robotic Systems Laboratory (RSL) at the Australian National University has been involved in the construction of an autonomous underwater vehicle (AUV) named "Kambara", an aboriginal word for crocodile. The vehicle is autonomous in the sense that it does not require a human operator to control it. Occasional supervisory commands such as 'hold current position' and 'swim after target' may be provided. Some of the applications envisaged for the AUV are mapping out underwater regions, following and observing marine life, and inspection and maintenance of underwater structures.These applications propagate in the fields of marine biology, marine geology, underwater inspection and assistance. The main aim of the Kambara project is to further research particularly in the area of underwater vehicle control and vision systems.

Figure 1.1: The Crown-of-Thorns Starfish

1.1 Approaches

The tasks above are currently performed in two different manners. The first is the use of human divers and the second is one that utilises various underwater vehicles. In the case of cataloguing reefs for example, divers are required to perform broadscale surveys of coral densities and censuses of corals and marine life that form the basis for status reports. Currently, visual transect information is recorded using underwater video cameras held above the reef as diver follows a prescribed path [Williams et al., 1999]. The inherent high costs and scope for improvement in efficiency of using human labour, raise the need for a higher level of automation.

At the moment, a relatively large number of underwater robots are used around the world. However, the majority are of the remotely operated vehicles (ROV) variety. Remotely operated vehicles differ from autonomous vehicles in the sense that they require a human operator. Currently, ROVs are primarily used in oceanic research, mining, sea floor surveying, search and rescue, as well as naval defence applications. These vehicles are limited in these applications when a need for control of fine manouvers is required with respect to the underwater objects of interest. This is due largely to the fact that there is a limitation to the remote operators ability to apply the limited amount of useful feedback supplied by the vehicle to effectively close the control loop reliably and efficiently. Given the nonlinear nature of the underwater environment, operator experience and fatigue also comprimise the effective control of the vehicle. This amounts to a costly investment in time of an experienced operator and support staff to control an ROV, let alone perform the tasks envisiged for the Kambara. These costs see the diminishing economic advantage of ROVs over that of conventional human divers.

The Kambara project and other autonomous vehicle projects around the world seek to address these problems. That is to say that autonomous vehicles such as Kambara are built with the aim of relieving the burden, difficulties and costs associated with conventional ROVs, whilst expanding their uses with scientific, commercial and military aplications in mind.

1.1.1 Our AUV - Kambara

Kambara is self-sufficient, with onboard computing, sensing, and power resources. The Kambara comprises of five thrusters and two cylindrical enclosures mounted on an open aluminium cylindrical frame. The five thrusters enable roll, pitch, yaw, heave, and surge manoeuvres. It is therefore underactuated and not able to perform direct sway (lateral) motion. The frame is 1.2m long, 1.5m wide and has a height of 0.9m. The assembled vehicle displaces approximately 120 litres of water.

The upper enclosure houses a computer system, frame grabber, analog-to-digital converters and communications equipment. Also in the upper enclosure is a sensor suite consisting of a triaxial accelerometer, compass, biaxial inclinometer, rate-gyroscope and tem-

Figure 1.2: **Kambara**

perature sensor. The lower enclosure contains lead-acid batteries, power distribution and charging circuitry, a depth sensor and a leak sensor. Stereo cameras are mounted on the front frame. The bulk of the processing power is present on-board.

1.2 Project Description & Contributions

Work on the inertial navigation system (INS) of the Kambara was commenced by [Beswick, 1998]. The purpose of the INS was to provide adequate sensor data to estimate the current state of the AUV in real time. The state of the vehicle describes an estimation of position, attitude, linear velocity and angular velocity. This state information is then utilised by the motion control system to navigate the vehicle.

In order to achieve this, Beswick proposed the use of a Kalman filter, incorporating a linear model of the AUV and the sensors. This linear model however ignored hydrodynamic added mass and gyroscopic acceleration terms. Therefore, the linear Kalman filter was deemed insufficient and to improve performance nonlinear modeling was required. Betlehem improved the model to a more accurate nonlinear one and incorporated it into an extended Kalman filter [Welch and Bishop, 1995]. Integral to the Kalman filter are a process model and the sensor model.

The process model is a set of equations that describe the current *state* of the AUV as a function of the *state* at a previous time and the control inputs. The process model will not be dealt with in this thesis. Details on the process model are provided in [Betlehem, 1999].

This thesis will instigate an investigation in the properties of the sensors that will be included in the sensor model of the state estimation system. In order to improve the accuracy of the sensor model, it is imperitive to evalute sensor characteristics such as bias and scale factor properties. It is also of interest to investigate the ability to estimate the state parameters by using only the sensed output without the use of filtering techniques such as the Kalman filter. This thesis aims to evaluate such characteristics of the sensors and also to observe the ability of using the inertial set of sensors as a stand-alone inertial navigation system in calculating state parameters such as position, velocity and attitude of the sytem.

1.2.1 Project Objectives

The main focus of this project is to determine the characteristics of the sensors under dynamic conditions. In order to do this, an experiment was required in order to test the sensors dynamically for their scale factor, offset biases and drift rates. The data gathered from these experiments is also to be used in investigating the drift properties of the integrated values of the inertial sensors. All the while, constant progress was to be made on the implementation of the Extended Kalman Filter. A summary of the aims is outlined:

- Devellop an experimental method that will enable the calculation of accurate bias and scale factor values for the inertial sensors.
- Calculate the bias offset value for both of the inertial sensors
- Calculate the scale factor for the accelerometer and rate gyroscope
- Compare the performance of the experimentaly calculated bias and scale factor with those provided by the manufacturer.
- Investigate the drift properties in calculating higher order parameters such as position and velocity.
- Compare the drift properties in estimating attitude from the rate gyroscope and TCM2 inclinometer/compass
- Develop the algorithms neccessary for the eventual implementation of the extended Kalman filter on the AUV
- Identify any problems with the sensors or hardware related to the sensors

These objectives have all been met and the results are outlined in the remainder of the thesis. A brief description of each chapter follows.

1.3 Outline

Chapter2 This chapter describes the dynamic experiment that was employed. It presents the hardware that was neccessary to conduct the experiment and it also outlines the problems/limitations to the experiment. A description of the sensors is also included.

Chapter3 This chapter is concerned with the dynamic testing of the sensors. It proposes an iterative least squares method designed to process the data gathered from the experiments in order to calculate the bias and scale factor for both the inertial sensors. Analysis of the drift characteristics of calculated position, velocity and attitude is also presented.
Chapter4 Presents the results from the static testing of the sensors.
Chapter5 Presents an overview of the software used to drive the Kalman filter algorithm.
Chapter6 Outlines a number of conclusions as well as future work required for the completion of the sensor testing phase as well as the state estimation system.
A description of the supporting material in the Appendices follows:

Appendix A Outlines the format of the information contained in the output files from the WAM
Appendix B Describes in detail how the MotionPak rate gyroscope functions
Appendix C Contains a certificate of the MotionPak data sheets
Appendix D Holds a diagram of the WAM kinematic description
Appendix E Contains Pro Engineer calculated data for the sensor suite
Appendix F Contains the derived Compass/Inclinometer sensor equations.
Appendix G Illustrates the transient magnetic field sources in the WAM experiment room
Appendix H A description of the type of tests used in the analysis
Appendix I Holds a flow-chart description of the files execution process for analysing the *Accelerometer*
Appendix J Contains a portion of the Matlab code that was used to implement the filters for differentiation.
Appendix K The results from the experiments in optimising the weighted average filter window are contained here
Appendix L Holds a flow-chart description of the files execution process for analysing the *Rate Gyroscope*
Appendix M Holds a flow-chart description of the files execution process for analysing the *Inclinometer/Compass*
Appendix N This appendix has a detailed description of the functions of the individual modules(software) within the State Estimation System.
Appendix O Contains a description of the directory structure of the files provided in the CD-ROM with this thesis
Appendix P Contains a series of Pro Engineer Drawings that were used to manufacture the mount sensor suite on the WAM

Chapter 2

Experimental Setup

In order to evaluate the characteristics of the inertial sensors, an experiment was conducted that enabled the investigation of the dynamic behaviour of the sensors.

The aim of the dynamic experiments was to determine sensor parameters such as dynamic biases and possible scale factor nonlinearities. The particular values calculated included the zero offset bias and the sensitivity of the triaxial accelerometer and rate gyroscope. These values are crucial to the accuracy of the sensor model and hence the state estimation system as a whole as these parameters will be included in the Kalman filter algorithm. Other sensor characteristics such as static and dynamic noise properties could also be investigated.

The analysis of sensor behaviour such as drift properties from parameters integrated directly from the sensor data will also prove important in understanding the limitations of the sensor information and provide possibilities for the improvement of the sensor model. If the values taken from direct integration from the sensors are to be deemed accurate enough, then less emphasis will need to be placed on refining the process model parameters of Kambara.

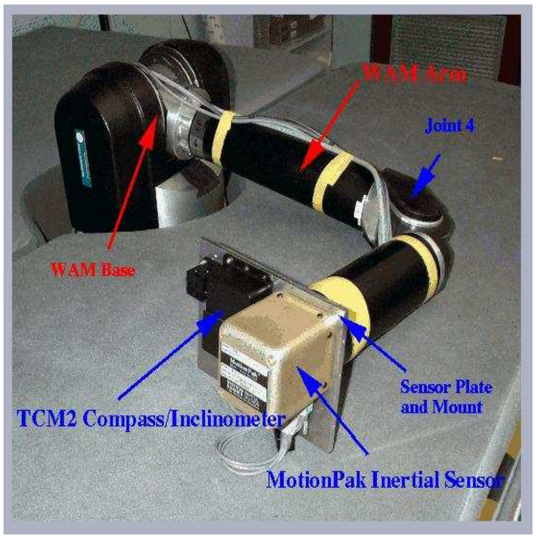


Figure 2.1: Sensor Suite mounted on the WAM

In order to perform the above tasks, we needed "groundtruth" data, that is data that is taken to accurately represent the bahaviour of the system. By attaching the sensors to a 7 DOF manipulator arm called the Whole Arm Manipulator (WAM) at the RSL, accurate measurements of the Cartesian position[1] and attitude of the sensors[2] were obtained. This was made possible by hav-

[1]a resolution of 0.5 mm
[2]Euler Angles

ing an accurate model of the WAM arm and hence calculating the forward kinematics of the system at every time step. From the WAM information provided, it was possible to calculate accelerations and angular rates for comparison with the raw sensor output. The following sections will describe the hardware used in the experiments and their limitations. The first section will concentrate on the sensors used and the second section will clarify issues concerned with the WAM.

2.1 Hardware/Software Interface

The figure below outlines the hardware and software interface for the components included in the dynamic experiments on the WAM. More information on this aspect of the experiment is provided in Chapter 5. Information on the function of the individual sensors is presented in the section that follows. The format of the output files and sampling frequencies are provided in appendix K.1.

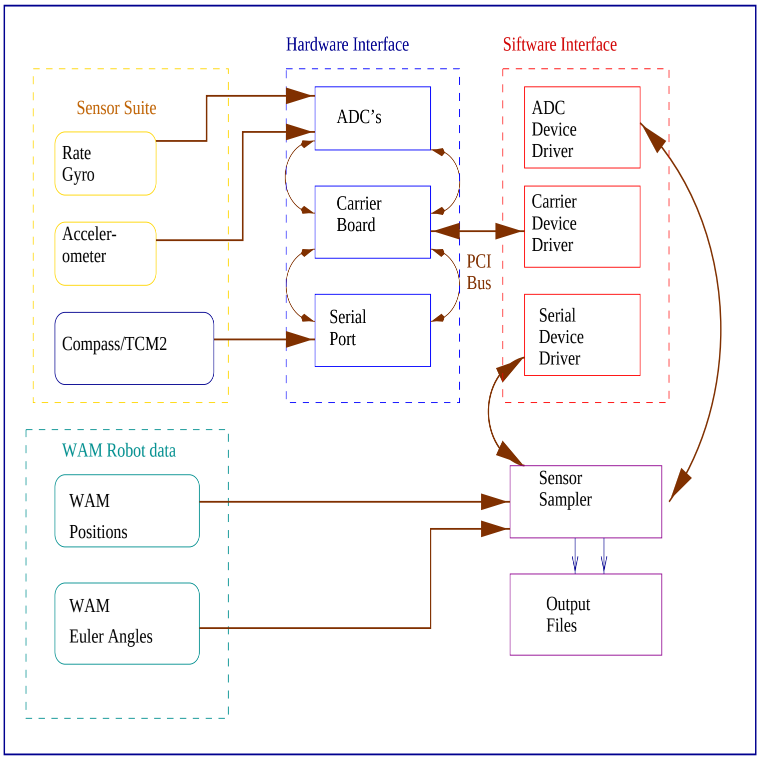


Figure 2.2: This diagram shows the Hardware/Software Interface of the SES

2.2 Hardware - Sensors

2.2.1 The Sensor Suite

Kambara is equipped with a sensor suite that includes:

- a **SensorTechnics depth sensor** model PTE2005G1A;
- a **Systron Donner MotionPak** model MP-GDDDQBBB-100;
- and a **Precision Navigation TCM2 compass unit** integrating a biaxial inclinometer and magnetic compass model TCM2-50.

The TCM2 produces a digital RS-232 output. Therefore, it is interfaced with a serial port module. The other sensors produce analog signals which are converted into a digital form through an analog-to-digital converter (ADC) module. The sensor output interfacing is discussed in detail in chapter 6 of [Betlehem, 1999].

2.2.2 Sensor Mounting

To accomplish the dynamic testing for the sensors, a suitable mounting was needed to be designed so that the sensors could be attached to the WAM arm. The same mounting is also be eventually fixed in the enclosures of the Kambara. Important considerations in the design of the sensor plate were rigidity and minimisation of sensor vibration so as to minimise additive senor noise. A suitable adaptor plate used to attach the sensor suite onto the wrist of the WAM arm has also been designed and manufactured (see Appendix P).

The components included on the sensor mounting were:

- The MotionPak module
- The TCM2 Compass/Inclinometer
- A DC-DC Converter

As the depth sensor measures pressure in the the Kambara, it must be in contact with the water outside the enclosure. Investigation of this sensor out of the water environment would provide limited insight in its properties, such as sensitivity to acceleration or vibration. However, the inconvenience of designing an additional sensor mounting to the existing one was deemed excessive. As a result, the depth sensor properties are yet to be investigated in future underwater experiments as it is to be mounted into the front-end cap of the lower enclosure in the Kambara. The depth sensor will not be mentioned for the remainder of this thesis. For more information on this sensor or the physical characteristics of the sensor mount see [Betlehem, 1999].

Two changes have been made to the previous sensor mount.

- The Summit Instruments accelerometer [3] is no longer included in the sensor model

[3] See Betlehem [Betlehem, 1999] for reasons for omitting this sensor from the INS.

and hence was not attached to the sensor plate.

- A new DC-DC converter (Model - Type)has been purchased and installed on the sensor plate. Experimentation on the effects of the possibility of added noise by the new DC-DC switch mode power supply on the MotionPak sensors has not been carried out. These effects can be easily investigated by examining the noise standard deviations from the MotionPak sensors with and without the DC-DC converter.

2.2.3 MotionPak-An Inertial Sensor Package

The Systron Donner MotionPak is a 6 DOF inertial sensing system. It contains three orthogonally mounted linear servo accelerometers, three micromachined quartz angular rate gyroscopes, and a temperature sensor. These are all to be used in the state estimation system. Inertial sensors are used in applications where rotational and linear movements are to be measured without reference to external coordinates. The summary data sheet from the manufacturer is provided in appendix C.

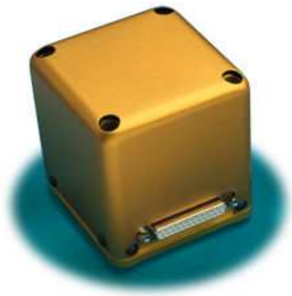


Figure 2.3: The MotionPak Sensor Module

2.2.3.1 Triaxial Accelerometer

A triaxial accelerometer measures acceleration in three mutually perpendicular directions. On Kambara, the three axes are approximately aligned with the body-fixed frame $\{B\}$ .The MotionPak accelerometer superimposes gravity on the sensor output. As the gravity term must be subtracted from the accelerometer output so that a true indication of acceleration can be given, the accelerometer readings can hence be no more accurate than the attitude estimates coming out of the Kalman filter. This is because the attitude needs to be accurately known for subtracting gravity completely. The attitude estimate in turn will depend on the rate-gyroscope, compass and inclinometer sensor readings.

2.2.3.2 Triaxial Rate Gyroscope

Gyroscopes measure rotational values without reference to external coordinates. The triaxial rate gyroscope on the MotionPak measures angular velocity in three mutually perpendicular directions. The operating principle of these sensors can be split into the following groups:

- Mechanical
- Optical
- Micro Electro-Mechanical Systems (MEMS)

The oldest and one that more intuitive is the rotating disc gyroscope, which belongs to the mechanical group. One of today's high precision gyros is still the mechanical, rotating gyro, however, it is dependent on linear accelerations because of its mechanical measuring principle.

The optical group comprises the fibre optic and laser types. These use the Sagnac effect (named after its French discoverer), which, when the sensor is turned, results in a difference in transit time between two light waves passing through the same optical path but in opposite directions). Therefore, optical gyros do not depend on acceleration as gyros in the mechanical group. This is one of their biggest advantage.

The MotionPak rate gyroscope belongs to the MEMS family. There are several types of MEMS rate gyros. The MotionPak includes a Quartz Rate Sensor (QRS) that uses a vibrating quartz tuning fork (see Figure B) to sense rate, acting as a Coriolis sensor. Coupled to a similar fork is a pickup to produce the rate output signal. This type of gyroscope is also referred to as a dual mass gyroscope. The advantages of these kind of systems is their cancellation of thermal drifts and accelerations as well as improved vibration sensitivity. Experiments conducted by [Betlehem, 1999], have also demonstrated this low correlation of the rate gyroscope output with respect to temperature. For more information on the operational theory of the rate gyroscope sensor see appendix B.

Specifications on the rate gyroscope sensor are given in the table below. Testing on these sensors was aimed mainly at high speed applications (50-500 deg/sec). Our application for the Kambara however, focuses on low-speed manoeuvres. This is an added incentive to investigate the performance of the sensors under lower rates and accelerations.

The sensor will produce a signal output in volts proportional to the rate of rotation sensed. Speed of rotation is normally given in degrees(or radians) per second, therefore, it is necessary to determine a scale factor that will transform raw voltage readings into the correct degrees per second units. The importance of determining a correct scale factor is also highlighted in chapter 4 which describes the dynamic experiments.

2.2.4 TCM2 Compass/Biaxial Inclinometer

The TCM2 compass module possesses a biaxial inclinometer and a triaxial magnetometer. The purpose of this module is to measure attitude. The biaxial inclinometer has no mechanical moving parts and it uses a fluid filled tilt sensor, which is an angle sensing device that uses gravity as a reference to measure the orientation of the compass.

There are three major limitations to this module:

- Diminished performance under acceleration
- Limited tilt range (± 50∘)
- Sampling Speed

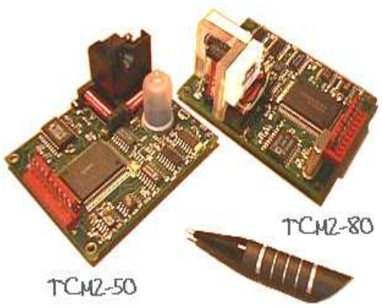


Figure 2.4: **TCM2-50 Module**

The first point raises from the fact that the tilt sensor is based on the principle of a carpenter's level-liquid and a bubble. Therefore, the tilt sensor will be inaccurate in situations where there are sudden changes in heading, which would cause the fluid in the inclinometer to slosh. The settling time for the sloshing fluid is 300msec. Acceleration exerted on the fluid will also cause it to tilt due to inertial effects, causing a decrease in accuracy. A relationship between acceleration and tilt error is provided by the manufacturer in the TCM2 user's manual. Reference to how this equation can be integrated into the SES will be made again in Chapter 3.

The derived equations provide a basic model that is most accurate when the TCM2 undergoes little or no acceleration. These equations are not presented here due to lack of space[4]. Problems with attitude measurements for high accelerations can be avoided by compensating the inclinometer equations and filtering the triaxial magnetometer readings directly [Smith et al., 1999]. It was not possible to test such a model in this experiment due to transient magnetic field disturbances (see section 2.5.4.2)

2.3 The WAM

A WAM (Whole Arm Manipulator) was used to aid in the investigation of the dynamic behaviour of the sensors. Our WAM is a Barett WAM robot, the commercial model of the arm built in MIT. It is a cable driven robotic arm with seven DOF. Appendix D shows the frame references for the WAM configuration. We see that the inertial frame on the WAM at the base of the robot has the {Z} frame pointing up i.e. opposite to the direction of the gravity vector. To obtain the results from the dynamic experiments presented in Chapter 3, a considerable amount of frame manipulation and transformation was needed. The next section outlines a description of the frames in the WAM/sensor plate configuration. This section is followed by a description of two different forms of attitude representation that were necessary to obtain the desired results.

2.4 Frame Description

The representation of the sensor plate/WAM system will consist of two coordinate systems. An inertial reference frame {I} will be used to describe the position and attitude of the sensor plate. The position and Euler angle output from the WAM are both with respect to the inertial frame. A local body-fixed frame, frame {B} is used for specifying the parameters such as angular velocity and acceleration. When implementing the EKF, it will be convenient to attach the body-fixed frame to the center of mass of the plate. However, when investigations of sensor characteristics are carried out, frame {B} will be attached to the volumetric centre

[4]See appendix F for a discussion on these equations

of the respective sensor[5].The axes of frame {B} will be alligned to that of the accelerometer and rate gyroscope axis. See Figure 2.5 for a graphical description of the frames. For more information on the frames allocated to the WAM see appendix D.

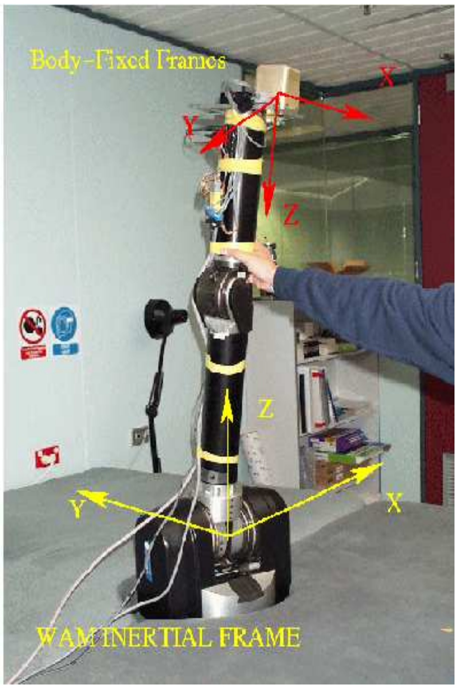


Figure 2.5: **Reference Frames - Inertial and Body reference frames**

2.5 Attitude Representation

There are different ways of representing the attitude (or orientation) of the AUV. Two common forms are Z-Y-X Euler angles and quaternions. There are several disadvantages to Euler angle representation. The first is that all Euler angle representations contain singularities. The second problem in using Euler angles for attitude representation is in computation time. That is, the calculation of six trigonometric functions are required for every computation of the transformation matrices as defined by [Craig, 1986]. The use of quaternions avoids the above problems. It uses a fourth parameter to eliminate the problem of evaluating trigonometric functions and it also eliminates the problem of singularities. The update of the quaternion from the sensor and process models will be updated recursively. This will involve a sequence of multiplications and additions. The two representations of attitude are described below. The relevant linear and angular transformations are also given for

[5]This is to minimise effects such as Coriolis effects on acceleration and to accurately compare position estimates without having to correct for the offsets. More information will be provided in the process model section.

each case. For more background on these results see [Craig, 1986] for the Euler section and [Fjellstad and I.Fossen, 1994] for the quaternion section.

2.5.1 Euler Angles

The attitude/orientation of frame {B} in frame {I} is commonly specified in terms of ZYX Euler angles. These are:

- Yaw angle ψ which is the angle of rotation of frame B about the Z-axis
- Pitch angle θ - a rotation about the Y-axis
- Roll angle ϕ - a rotation about the X-axis

With these Euler angle parameters (represented by the vector $\Theta(\psi, \theta, \phi)$) it is possible to obtain matrices that convert parameters stated in terms of Frame {B} into inertial reference frame parameters. In particular, it will be necessary to convert linear accelerations, velocities and angular velocities expressed in terms of frame {B} into inertial acceleration,velocity and Euler angle parameters respectively.

2.5.1.1 Linear Velocity and Acceleration transformation

Let the sensor plates position relative to the inertial frame be represented by the vector $\mathbf{r}(x, y, z)^T$ and the velocity of the sensor suite in the body fixed frame is represented by the vector $\nu(u, v, w)^T$. The sensor suite's velocity and acceleration flight path relative to the inertial coordinate system is then given by the transformations:

$$\dot{\mathbf{r}} = R(\Theta)\nu \Leftrightarrow \nu = R^{-1}(\Theta)\dot{\mathbf{r}} \tag{2.1}$$

$$\ddot{\mathbf{r}} = R(\Theta)(\dot{\nu}) \Leftrightarrow \dot{\nu} = R^{-1}(\Theta)\ddot{\mathbf{r}} \tag{2.2}$$

where $R(\Theta)$ is a linear transformation matrix which is related through the functions of the Euler angles. Note the exclusion of the Coriolis componentsof acceleration. This is because the distance between the body-fixed frame and the point that is investigated (i.e. the sensor position) is negligible as frame {B} is attached to volumetric centre of the MotionPak sensor for the dynamic testing of the inertial sensor. The expression for $R(\Theta)$ is given by[6]:

$$R(\Theta)\nu = \begin{pmatrix} c\theta c\psi & s\theta s\phi c\psi - c\phi s\psi & s\theta c\phi c\psi + s\phi s\psi \\ c\theta s\psi & s\theta s\phi s\psi + c\phi c\psi & s\theta c\phi s\psi - s\phi c\psi \\ -s\theta & c\theta s\psi & c\theta c\psi \end{pmatrix} \tag{2.3}$$

The matrix $R(\Theta)$ from 2.3 is orthogonal. As a consequence, the inverse rotation matrix can be computed as $R(\Theta)^{-1} = R(\Theta)^T$.

[6]Note that c and s represent cosine and sine functions

2.5.1.2 Angular Velocity Transformation

The body-fixed angular velocity vector $\omega = (p, q, r)^T$ cannot be integrated to obtain actual angular coordinates since $\int_0^t \omega(\tau)d\tau$ has no immediate physical meaning. Instead, one can obtain the rate of change of the Euler angles from the angular velocity by applying the angular transformation matrix $U(\Theta)$. The orientation of the body-fixed reference frame {B} with respect to the inertial frame is given by:

$$\omega = \mathbf{U}^{-1}(\mathbf{\Theta})\dot{\mathbf{\Theta}} \tag{2.4}$$

where

$$\mathbf{U}^{-1}(\mathbf{\Theta}) = \begin{pmatrix} 1 & 0 & -s\theta \\ 0 & c\phi & c\theta s\phi \\ 0 & -s\phi & c\theta c\phi \end{pmatrix} \tag{2.5}$$

$$\mathbf{U}(\mathbf{\Theta}) = \begin{pmatrix} 1 & s\phi t\theta & t\theta c\phi \\ 0 & c\phi & -s\phi \\ 0 & s\phi/c\theta & c\phi/c\theta \end{pmatrix} \tag{2.6}$$

Note that $\mathbf{U}(\mathbf{\Theta})$ is undefined for a pitch angle of $\theta = \pm 90^\circ$ and that $\mathbf{U}^{-1}(\mathbf{\Theta}) \neq \mathbf{U}^T(\mathbf{\Theta})$.

2.5.2 Quaternions

Consider a quaternion $\mathbf{q} = (\eta, \epsilon^{\mathbf{T}})^{\mathbf{T}}$ where η is a scalar and ϵ is a 3 vector. The transformations to convert the velocity and angular velocity both expressed in terms of frame {B}, into inertial velocity and quaternion parameters are given by the linear transformation **R(q)** and the angular transformation **U(q)** as defined in [Fjellstad and I.Fossen, 1994].The transformation matrices **R(q)** and **U(q)** have full rank and the kinematics equations contain no singularities or trigonometric functions.

2.5.2.1 Linear velocity and Angular Transformations

The relevant transformations to obtain the velocity and acceleration in frame {I} from those in frame {B} are given by:

$$\dot{\mathbf{r}} = \mathbf{R}(\mathbf{q})\nu \tag{2.7}$$

$$\ddot{\mathbf{r}} = \mathbf{R}(\mathbf{q})\dot{\nu} \tag{2.8}$$

where

$$\mathbf{R}(\mathbf{q}) = \begin{pmatrix} 1-2\epsilon_2^2-2\epsilon_3^2 & 2(\epsilon_1\epsilon_2-\eta\epsilon_3) & 2(\epsilon_1\epsilon_3+\eta\epsilon_2) \\ 2(\epsilon_1\epsilon_2+\eta\epsilon_3) & 1-2\epsilon_1^2-2\epsilon_3^2 & 2(\epsilon_2\epsilon_3-\eta\epsilon_1) \\ 2(\epsilon_1\epsilon_3-\eta\epsilon_2) & 2(\epsilon_2\epsilon_3+\eta\epsilon_1) & 1-2\epsilon_1^2-2\epsilon_2^2 \end{pmatrix} \tag{2.9}$$

2.5.2.2 Angular Velocity and Acceleration Transformations

To obtain the rate of change in a quaternion from the angular velocity in the body-fixed frame one can use the following relationship:

$$\dot{\mathbf{q}} = \frac{\mathbf{1}}{\mathbf{2}}\mathbf{U}(\mathbf{q})\omega \tag{2.10}$$

and similarly for angular acceleration:

$$\ddot{\mathbf{q}} = \frac{\mathbf{1}}{\mathbf{2}}\mathbf{U}(\mathbf{q})\dot{\omega} \tag{2.11}$$

The angular transformation matrix is given by:

$$\mathbf{U}(\mathbf{q}) = \begin{pmatrix} -\epsilon_1 & -\epsilon_2 & -\epsilon_3 \\ \eta & -\epsilon_3 & \epsilon_2 \\ \epsilon_3 & \eta & -\epsilon_1 \\ -\epsilon_2 & \epsilon_1 & \eta \end{pmatrix} \tag{2.12}$$

2.5.3 Experimental Method

The experiment was composed of several phases. The first phase which was the preliminary phase needs to be performed only once. The second phase needs to be carried out only when the system is reset and the final phase is relevant every time a new experiment is run.

Phase One:

- calibrate the TCM2 compass sensor to its experimental workspace. The method for calibrating the compass is outlined in the TCM2 user's manual. By performing the user calibration procedure, the TCM2 module is allowed to identify the major sources of the local magnetic anomalies and subsequently cancel out their effects when measuring the earth's magnetic field for computing compass heading. This method only aids the measurements in an environment that does not contain dynamic magnetic fields.
- use Pro Engineer to accurately calculate the transformation matrix[7] from its joint number 4[8] to the body fixed frame. The transformation matrix from the body-fixed frame {B} to the elbow joint J4 is required so that the program that performs the kinematic

[7] see section 2.5.1
[8] To identify the joint on the WAM see appendix D

calculations for the motion of the arm is able to output the correct position and attitude of the center of mass of the inertial sensor. [9]. The next step is to input the transformation matrix R_B^{J4} matrix into the C++ code that performs the forward kinematic calculations. See apendix[E] for details on these parameters.

Phase Two:

- Perform a cold start on the WAM. This procedure is used to initialise the frame referencing for the WAM. The cold start should be performed carufully as errors due to the "coldstart" will propagate through to position and attitude estimates.(i.e. axis might be offset from the expected ones)

Phase Three:

- Execute the *Sensor Sampler* and input the number of seconds that the experiment is intended to run for. Once the sensor sampling code has executed, the sensor plate was handled manually. The output consists of three files. For information on the contents of these files see appendix K.1

2.5.4 Experimental Limitations

2.5.4.1 Automation

It was desirable at the early stages of the project to investigate the possibility of automating the experiment so far as to control the motion of the WAM manipulator. Useful experiments would be motions that involved constant velocity or acceleration which is otherwise impossible when controlled manually. This exercise was deemed futile however as the WAM introduces a lot of noise through it "jerky" movements when automatically controlled.

To achieve a certain velocity path for the WAM it is also important to have accurate joint torque measurements. Torque ripple effects however introduce a noisy joint torque measurement.

2.5.4.2 Magnetic Distortion

It was stated earlier in section 2.2.4 that it is also possible for the compass/inclinometer configuration to make use of magnetometer values so that accuracy would be improved under non-zero acceleration. However, static tests performed in different position of the WAM in its workspace showed frequent magnetic field distortions through the activation of the appropriate flag. This was found to be largely due to the transient magnetic sources present in the vicinity, such as the transformer unit that powers the WAM. These transient magnetic anomalies made the investigation of the compass model that uses magnetometer readings for attitude correction impossible. Appendix G gives an illustration of the workspace and the major sources of high magnetic field distortion.

[9]which also where the body-fixed frame is attached

Chapter 3

Dynamic Sensor Testing

This chapter presents the dynamic experiments used to investigate different properties of the sensors. These include scale factor, bias and drift properties. The chapter begins with a description of the method employed in calculating the scale factor and bias followed by a brief description on the concept drift. The results from the dynamic experiments are presented for the accelerometer, rate gyroscope and TCM2 inertial sensors in separate sections.

3.1 Bias and Scale Factor

In this section a least squares method is proposed in calculating the offset bias and the scale factor(*sf*) through offline testing of dynamic experimental data. In utilising this method to obtain accurate results, two assumptions were made.

- Zero offset bias is defined to be the sensed output when the input is zero.[1] Under static conditions, this bias maintains at a constant value. Therefore, it is assumed that the *offset bias* remains uncorrelated with time or other dynamic inputs. Thermal effects on the bias are also ignored[2]. Hence this implies:

$$\sum b = b_{zero-offset}$$

- The scale factor, *sf* on the other hand is considered to be a linear transformation of the sensed values. Therefore, the scale factor remain constant for the duration of the experiment.

In combining the two assumptions, we obtain a linear transformation for the sensed values M_{sensor}. This transformation is represented by:

$$R_{WAM} = M_{sensor} S_{sf} + b \tag{3.1}$$

[1]The motionPak accelerometer however, will still register the acceleration component due to gravity even under static conditions.

[2]See further down in this section for investigation of thermal effects

where $\mathbf{R}_{WAM}$ is the "real" value that the sensor is to represent in the desired units, and $\mathbf{M}_{sensor}$ is the raw value read from the sensor (eg. Acceleration in unit Volts for the MotionPak). $\mathbf{S}_{sf}$ represents the scale factor used to scale sensor readings to the appropriate units, while $\boldsymbol{b}$ is the bias value of the sensor.

Representing equation 3.1 in vector form:

$$\mathbf{R_{WAM}} = \Gamma \begin{bmatrix} S_{sf} \\ b \end{bmatrix} = \begin{bmatrix} \mathbf{M_{sensor}} & \mathbf{1} \end{bmatrix} \begin{bmatrix} S_{sf} \\ b \end{bmatrix}$$

$$\begin{bmatrix} S_{sf} \\ b \end{bmatrix} = \begin{bmatrix} m_1 & 1 \\ m_2 & 1 \\ m_3 & 1 \\ \vdots & \vdots \\ m_n & 1 \end{bmatrix}^{\dagger} \begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ \vdots \\ r_n \end{bmatrix} \tag{3.2}$$

The components of the vector $\mathbf{R_{WAM}}$ $(r_1 \ldots rn)$ represent the n data points that reflect the real behaviour of the system. Thus, in the case of acceleration, these are the real accelerations derived from the WAM Cartesian positions of a particular axis. The $(m_1 \ldots m_n)$ on the other hand are the corresponding sensor values[3]. Note that equation 3.2 is to be applied to one axis at a time[4].

An alternative equation to 3.1 would be one that includes a temperature correlated bias. This equation would add an extra temperature vector **T** to the Γ matrix in equation 3.1 and a scale factor k_{temp} that when multiplied by temperature, would add the excess bias correlated to thermal changes. Given that the experiments spanned a short period of time, the temperature would not be given time to change sufficiently to represent a change in the components of the **T** vector. This means that two of the three columns in Γ would become linearly dependent and hence the least squares method would not be feasible. Regardless, the accelerometer and rate gyro in the MotionPak sensor module are expected to be zero [5] as determined experimentally by [Betlehem, 1999]. However, if one chooses to include the temperature component in the equation, then an experimental method would have to be devised whereby the temperature vector would demonstrate different states[6]. This improvement to the method presented here would offer a generic dynamic testing method that could be employed in testing all inertial sensors.This includes those with sensitivity to thermal variations.

[3]That is to say that r_i and m_i correspond to the acceleration(or angular velocity) at time i.

[4]Matlab was used in implementing this equation. The Matlab command pinv() was used to implement the pseudo-inverse of the matrix

[5]See section 2.2.3.2

[6]eg. By attaching a variable heating element of some sort to the sensor during the dynamic experiment

3.2 Drift

It is desirable to estimate the scale factor and biases as accurately as possible or otherwise to model their behaviour in order to obtain an accurate value of the higher order parameters. For example, given that the Kambara does not hold a sensor that measures velocity, it would have to be calculated from our sensed accelerations. In which case, the bias must be removed entirely from the raw sensor values. If a residual bias is present in the measured value after scaling, then this bias would be integrated with time and causing a *drift* behaviour of the resulting velocity with respect to the true velocity. This phenomenon is demonstrated in section 3.4.4.1.

An intuitive analogy to this drift property is the case where a person's sensory perception is suddenly stripped away (eg. sight, hearing, touch etc). However, this person is only able to sense accelerations with a small error present. On this basis, if the person was made to walk x meters, he/she would most certainly overshoot or underestimate the mark due to the error being added over time. The longer the time period, the larger the resulting error and the smaller the initial error $bias$ the longer it takes the drift value to reach a certain value. If however, this person was given an extra sensor, say sight, then he/she would be able to minimize the error further by navigating through recognising objects in the path to the set mark. This introduces the concept of *land-mark navigation*. Surface vehicles normally fuse GPS data to correct for their drift properties. Unfortunately, GPS does not work very well under water and Kambara does not posses sonar sensors. Therefore, the Kambara must rely on its reduced set of sensors to optimise drift behaviour in its state estimates. In certain situations however, Kambara's stereo cameras could be utilised to provide $noisy$ distance estimates which could potentially be fused with the inertial information to correct for position drift.

Minimising drift properties relies on the success of accurately estimating bias and scaling properties of each sensor. The results from the following sections will demonstrate that the method outlined above in section 3.1 can be used reliably in accurately determining the offset bias values.

3.3 Experiments

Several experiments of differing merit were conducted. A list of these is given in Appendix H. In investigating the accelerometer bias experiment number 3 in Appendix H was used. While experiment 9 was used in the investigation of the rate gyro and inclinometer. The same experiment was used for both sensors so that comparisons could be made on their respective estimates of attitude. The type of experiment was limited to one that involved low accelerations as the performance of the inclinometer is degraded severely when undergoing high acceleration or rapid changes in direction. These observations are outlined in the following sections.

3.4 MotionPak Accelerometer Investigation

Method

As outlined in section 3.1, true acceleration values that represent the state of the sensor are needed in order to calculate the bias and scale factor parameters. However, as accelerations are not directly available from the WAM, they have to be calculated from Cartesian position that are relative to the inertial frame $\{I\}$. This means that differentiation of position needs to be performed in order to calculate the acceleration. Also, the resulting accelerations will need to be translated to the body-fixed frame $\{B\}$ and the gravity component for each vector needs to be added so as to yield real accelerations that can be compared to the sensed accelerations from the MotionPak given in the body-fixed frame $\{B\}$. Figure 3.1 is included to give a high-level representation of the process. A crucial step in this process is the differentiation of the WAM data. The differentiation techniques are discussed in section 3.4.1. Section 3.4.2 presents the resulting bias and scale factor values from the dynamic experiment.

3.4.1 Filtering WAM Data

Butterworth Differentiator

To obtain values for acceleration and angular velocity from Cartesian positions and Euler angles, a 4th-order Butterworth *differentiator*[7] filter with a cutoff of 30Hz was implemented. This differentiator effectively acts as a differentiator followed by a low pass filter[8]. Figure 3.2 presents the position, velocity and acceleration output from the Butterworth differentiator. Velocity is obtained by passing the position through one differentiator, whereas acceleration is obtained by differentiating the position twice. Of particular concern is the added noise due to the differentiator in the acceleration values. Despite the low pass filtering of the Butterworth differentiator, a considerable amount of noise remained in the acceleration signal. The source of the added noise is mainly due to the noise ripple introduced through differentiation of the data, and the fact that the differentiator is of a relatively low order.

Weighted Average Filter

The accuracy of the estimates for the correcting parameters (offset bias and scale factor) will depend largely on the actual accuracy of the "ground truth" data. In the case where the accelerations are differentiated from position, the added noise will reduce the effectiveness of the iterative least squares estimation algorithm outlined in section 3.1. To improve the noise characteristics of the resulting accelerations, an acausal triangular weighted average

[7] The Matlab code used to implement the filters is given in appendix J

[8] Even though linear phase filters introduce less errors for low frequencies, experimentation with FIR filters by using the "remez()" function on Matlab did not show a marked improvement in the noise characteristics of the derived acceleration

filter with an optimal window size of 90[9] points was applied to the differentiated data. The effect of including the weighted average filter(WAF) on the correcting parameters will be considered in section 3.4.4.2.

3.4.1.1 Results: Ground Truth acceleration, velocity and position

The resulting output from the differentiators and the weighted average filter is the ground truth accelerations in $\frac{m}{s^2}$ with respect to the inertial frame. As mentioned in section 2.2.3, the accelerometer includes the gravity component in its sensed readings. In order to compare the ground truth data and the accelerometer data correctly, the acceleration due to gravity must be added to the ground truth data[10]. Once this step is completed, the least-squares method can be applied to obtain the scale factor and bias for each axis of the sensor.

3.4.2 Bias and scale factor values for accelerometer

Equation 3.2 was applied to the WAM accelerations and the sensed accelerations over a 20 second interval(i.e. 2000 samples). The results for the scale factor and bias are shown in table 3.1.

$\mathbf{ScaleFactor}_{experiment}(\frac{\frac{m}{s^2}}{V})$	$\mathbf{Bias}_{experiment}(\frac{m}{s^2})$
2.611203	0.036094
2.660585	0.056003
2.670453	-0.033754

Table 3.1: Scale Factor and Bias values to convert MotionPak raw values to $\frac{m}{s^2}$ - derived experimentally

$\mathbf{ScaleFactor}_{manufacturer}(\frac{\frac{m}{s^2}}{V})$	$\mathbf{Bias}_{manufacturer}(\frac{m}{s^2})$
2.61604	-0.024545
2.62023	-0.036916
2.61117	-0.0189

Table 3.2: Scale Factor and Bias values to convert MotionPak raw values to $\frac{m}{s^2}$ - manufacturer's data

[9] The optimal window size was determined by calculating the scale factor and bias for different window lengths. It was found that the noise covariance was minimized where the window size included 90 points. See Appendix K for further information

[10] Or conversely, subtracted from the accelerometer

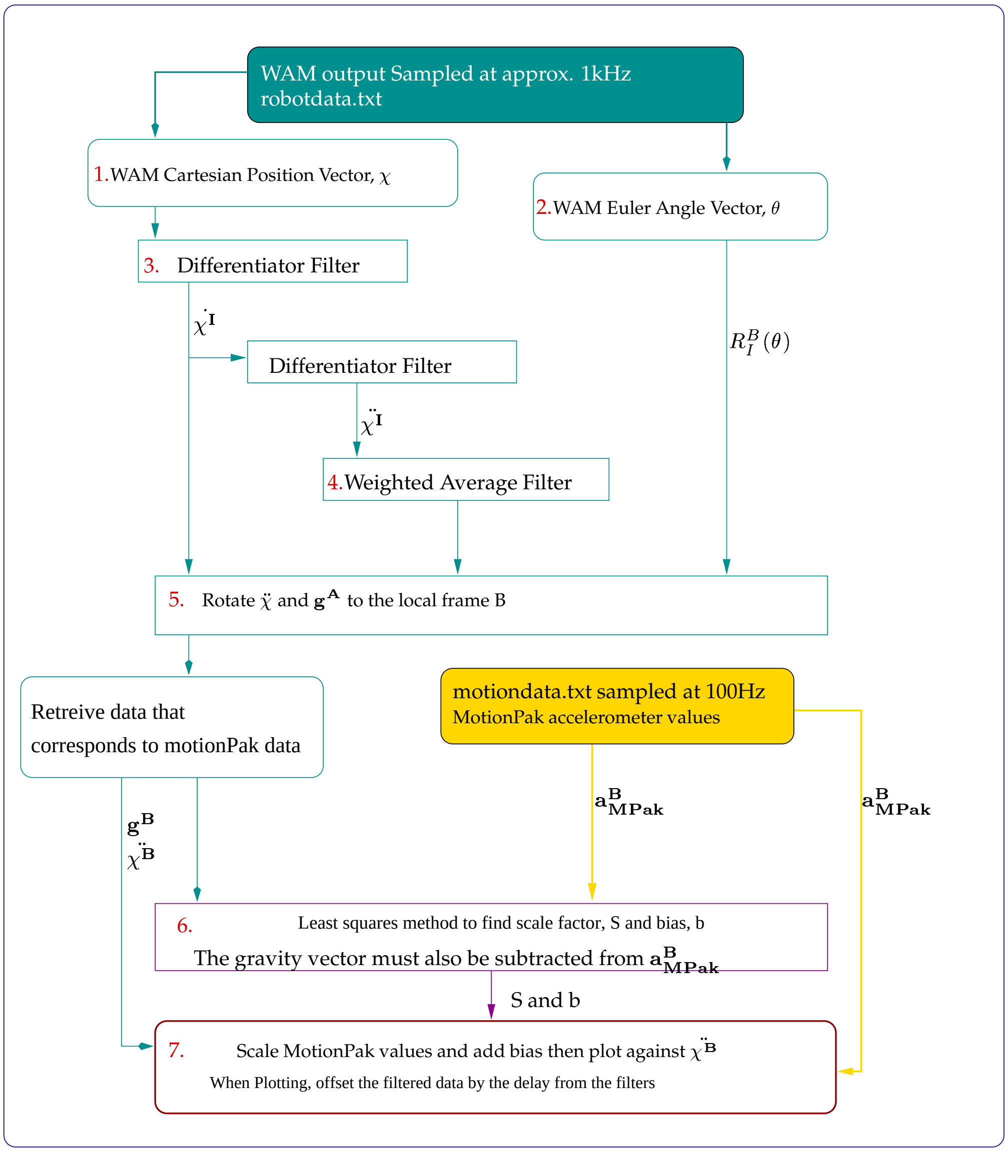


Figure 3.1: Method for obtaining acceleration from the WAM "groundtruth" positions and comparison with the MotionPak sensed accelerations.Cyan boxes indicate manipulation of WAM data, whereas Gold framed boxes indicate operations concerned with the sensor output

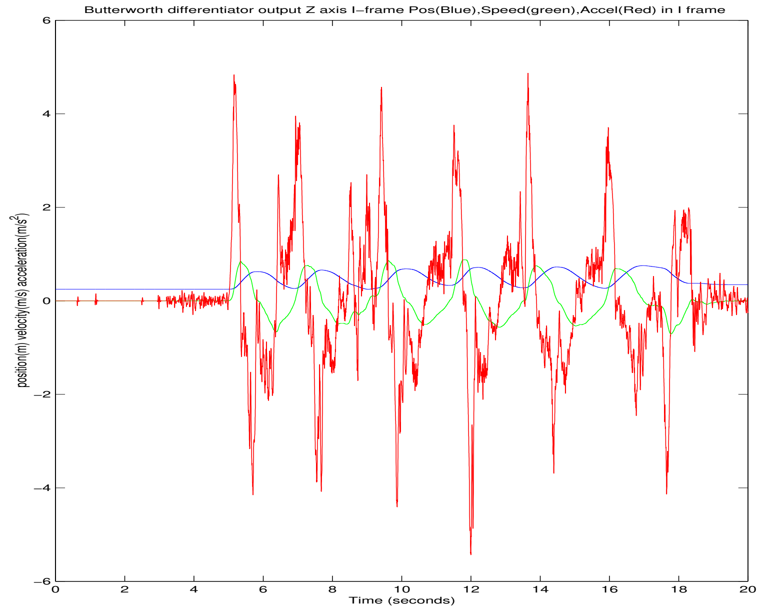


Figure 3.2: Output from Butterworth differentiating filter with 20Hz cutoff frequency applied on WAM robot position to obtain velocity and then applied obtain acceleration

Δ**ScaleFactor**$_{experiment-manufacturer}(\%difference)$	Δ**Bias**$_{experiment-manufacturer}(\frac{m}{s^2})$
-0.18	0.06064
1.7	0.093
2.27	-0.0148

Table 3.3: Difference in scale factor and bias between experimental and manufacturer's results. i.e. $\Delta ScaleFactor_{experiment-manufacturer} = \frac{SF_{experiment} - SF_{manufacturer}}{SF_{manufacturer}}$ and $\Delta \mathbf{Bias}_{experiment-manufacturer} = Bias_{experiment} - Bias_{manufacturer}$

3.4.3 Accelerometer output vs Ground Truth Accelerations

Figure 3.4 plots the acceleration as calculated from the WAM in the body-fixed frame superimposed on to the MotionPak acceleration (sensor output) after using the values in Table 3.1. The graph demonstrates a qualitative compatibility of the data and primary verification of the validity of the scale factor and bias values.

Figure 3.5 is a plot of the dynamic noise component of the accelerometer. It is a plot of the residual between the sensed and true acceleration resembling a white noise characteristic with a zero mean. This reflects the noise distribution of the accelerometer sensors in static

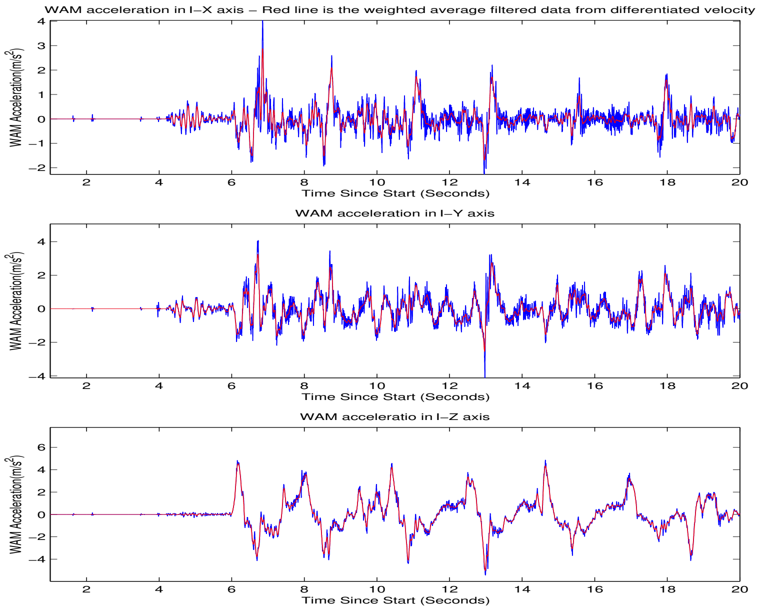


Figure 3.3: The effect of the weighted average filter with an 90 point window

tests [11]. This is a further indication of the validity of the scale factor and bias calculated through the experiment. The covariances of the three axes in the dynamic and static case are given below table 3.4.

Dynamic σ^2	Static σ^2
0.011	0.000022
0.017	0.000022
0.009	0.000186

Table 3.4: *Dynamic and Static Covariance Values. Note that the dynamic covariance is larger in value. This is due to the added differentiation and experimental errors in the "groundtruth" values.in* $\frac{m}{s^2}$

3.4.4 Drift Properties of Velocity and Position

By integrating the sensed accelerations with time and offsetting for the initial condition, it is possible to attain both velocity and position. It is of interest to know the level of precision that can be reached in estimating these values over a short period of time by using solely the accelerometer information. The integration method used was highly unsophisticated. The data points were simply added over time. Brief incursions into other higher order in-

[11] See Figure4.1

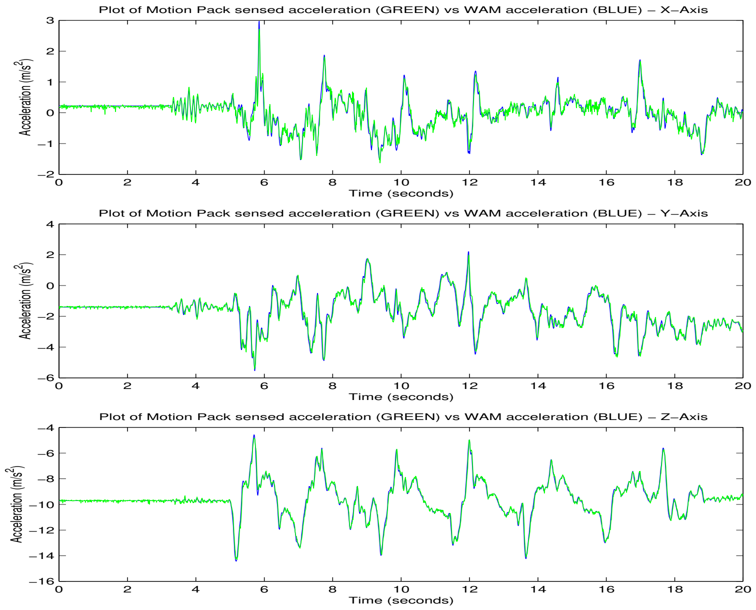


Figure 3.4: WAM accelerations Plotted on MotionPak Acceleration after scaling and correction for bias

tegration techniques[12] showed no improvement on the overall result. This is most likely due to the high sampling time of the sensor (100Hz). Better integration techniques could aid in cases where the sampling period is higher, hence a larger change between the present and the next value. Figure 3.6 demonstrates a flow-chart view of the steps taken to derive position and velocity. The next two sections outline the results of position and velocity drift.

3.4.4.1 Velocity Drift

Velocity in the three orthogonal directions in the body-fixed frame derived from a single integration of the sensed acceleration plotted against the real velocities are shown in Figure 3.7. We see that the drift rate on the velocity is almost completely corrected. This is especially evident in the Z and Y-Axis velocity plots, where the two plots are virtually overlapping. This indicates that the scale factor and bias values are extremely close to the actual values.

The difference between the calculated and real velocity value is plotted in figure 3.8. This plot illustrates the dynamic noise properties of the sensors. The plot suggests a random-walk property which is obtained when integrating white noise[13].

[12] 4^{th} order Runga-Kutta method

[13] See figure 3.5

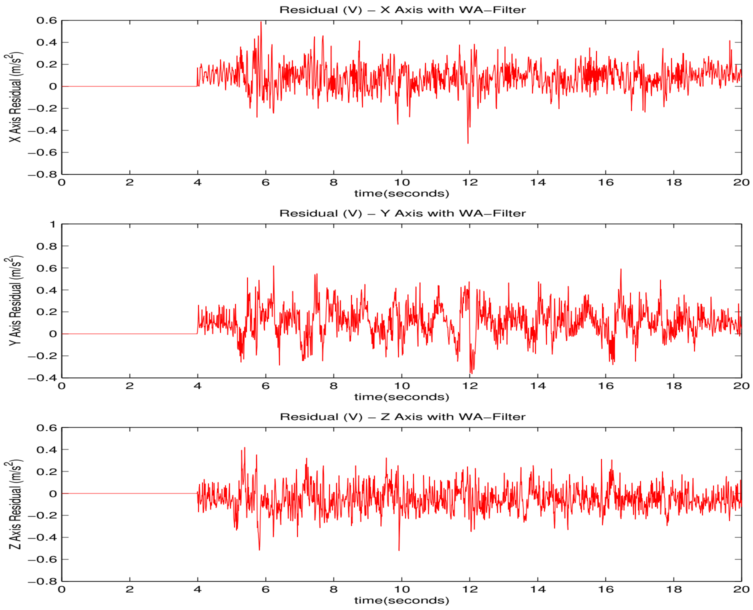


Figure 3.5: The residual of WAM acceleration and the motionPak scaled sensed acceleration. *note the white noise-like properties of the residual*

3.4.4.2 Position Drift

This section demonstrates drift properties of position. The red line in Figure 3.9 illustrates the position of the sensor plate in the inertial frame {I}. More importantly, it illustrates the superiority of the experimentally derived bias and scale factor values over the values provided by the manufacturer. This confirms the need for dynamic off-line testing of the inertial sensors. The manufacturer's values often differ from sensor to sensor. This is mainly due to manufacturing errors. For example, a different impedance at the sensor output could produce a slight voltage drop across the output, causing a difference in the offset bias. Other factors[14] could also contribute to a differing bias and scale factor to that provided by the manufacturer.

Figure 3.10 shows the difference between the integrated position (in the inertial frame) from sensor data and the raw position values recorded from the WAM. This is an important plot as it demonstrates that using the triangular weighted average filter to refine the WAM accelerations further minimised drift, which indicates a higher precision in the bias and scale factor estimation.

[14]for example circuitry between the raw sensor output and the final sample such as some kind of a filter

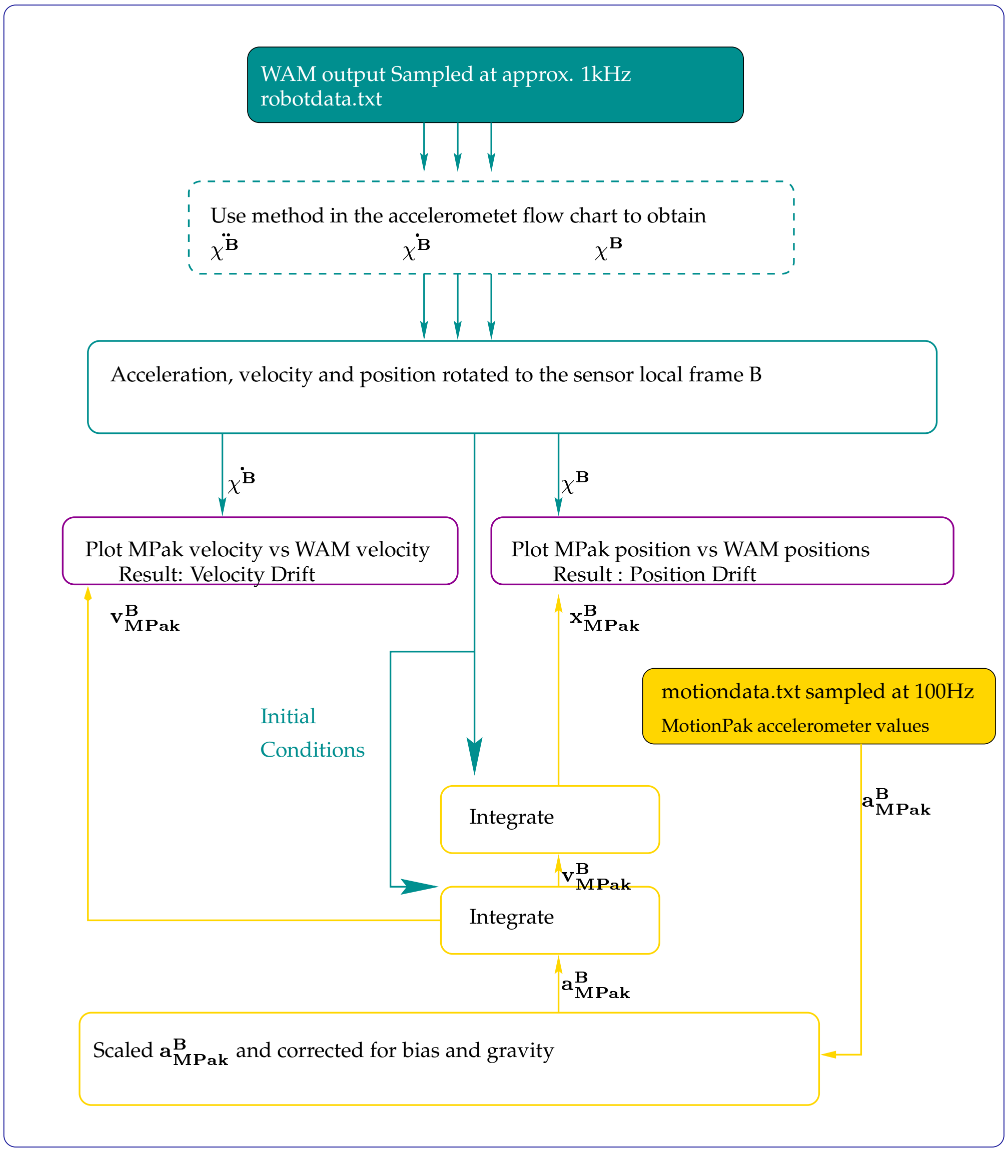


Figure 3.6: This flowchart demonstrates the method by which the drift rates for position and velocity were obtained

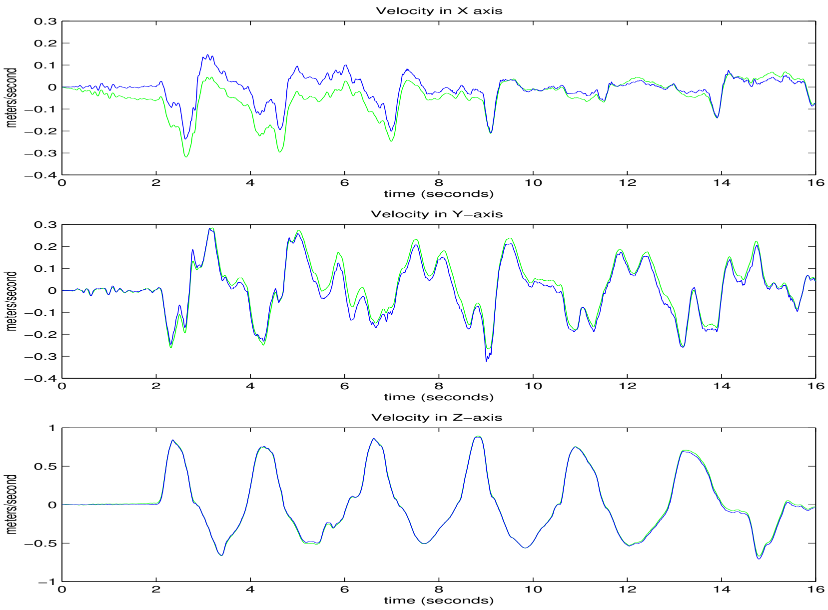


Figure 3.7: Plot of Velocity (Green) scaled with experimental scale factor and bias and the real velocity (Blue).

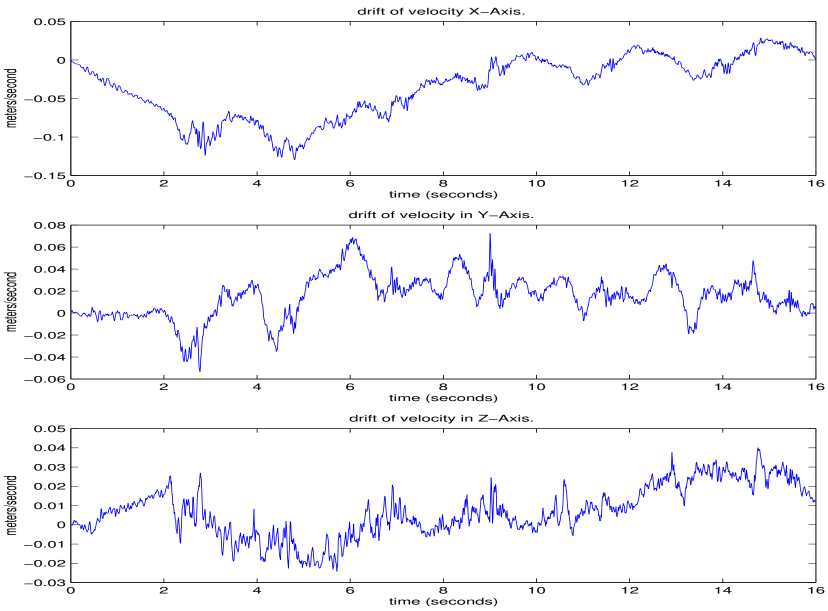


Figure 3.8: Drift of Velocity (Blue) scaled with experimental scale factor and bias. This plot was obtained by subtracting the estimated velocity from the real WAM velocities during the time interval.

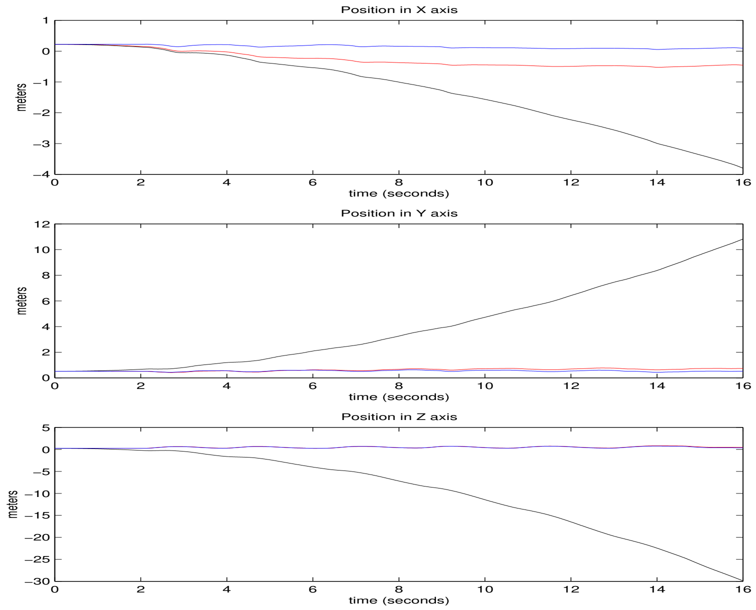


Figure 3.9: Plots of Position over 16 seconds. The Blue line is the real position from the WAM. The Black line represents the derived position by using the manufacturer's data. The Red line for position was obtained by using the scale factor and bias values calculated in the experiment. Note that the position calculated from the experimental results (Red Line) demonstrates a lower drift than the black line over all and especially in the Z-axis , where the Red line almost overlaps the Blue line.

3.5 Rate Gyroscope Bias and Scale Factor

The flow chart 3.11 demonstrates the steps involved in calculating the bias and scale factor for the rate gyroscope. The "groundtruth" angular rates were obtained from the Euler angle information provided by the WAM. Equation 2.12 was used to calculate the angular velocities. The relevant algorithms developed to achieve this are presented in appendix L. Note that a quaternion approach was taken in calculating the rates so as to avoid problems with singularities in the Euler data.

3.6 Attitude Quaternion and Euler Drift

The attitude of the AUV is one of the state parameters that is to be provided by the State Estimation System for the Kambara. As it stands, the sensor model for the EKF uses rate gyro sensor information to calculate the attitude quaternion for the vehicle. This section aims to provide some insight into the properties of the attitude value derived from raw sensor data.

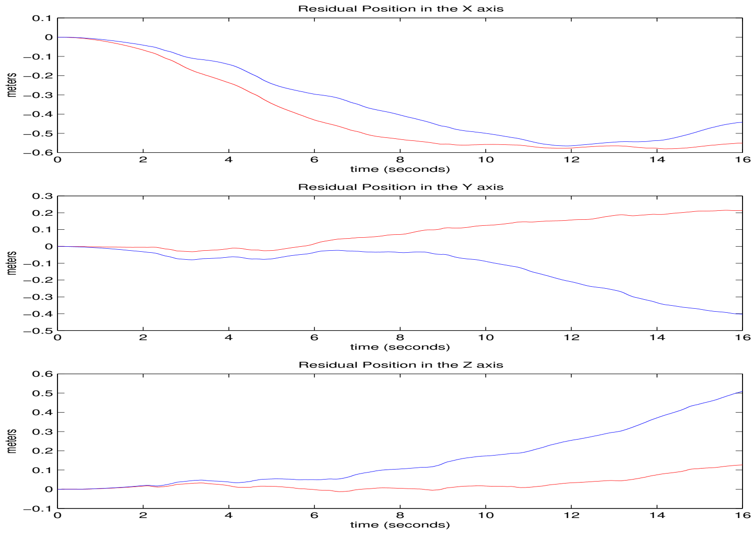


Figure 3.10: This is a plot of the drift value of Position. The difference between the two lines is that they were scaled and corrected for bias with different values. The line in Red uses the scale factor obtained when the derived WAM accelerations were passed through the weighted average filter. The correcting parameters for the blue line were calculated by using the noisy acceleration data derived from only the Butterworth differentiators the WAM

To demonstrate the attitude drift of the estimates, the Euler transformations presented in 2.5.1 will be used to obtain the rate of change in the Euler angles. This can then be integrated to provide the Euler angles at each time step. The method by which this was achieved is presented in figure 3.12. The resulting Euler angles derived from the sensor data plotted against the real angles are shown in figure 3.14. The maximumum drift rate determined experimentally over longer periods of time was found to be approximatelly 4 degrees per hour.

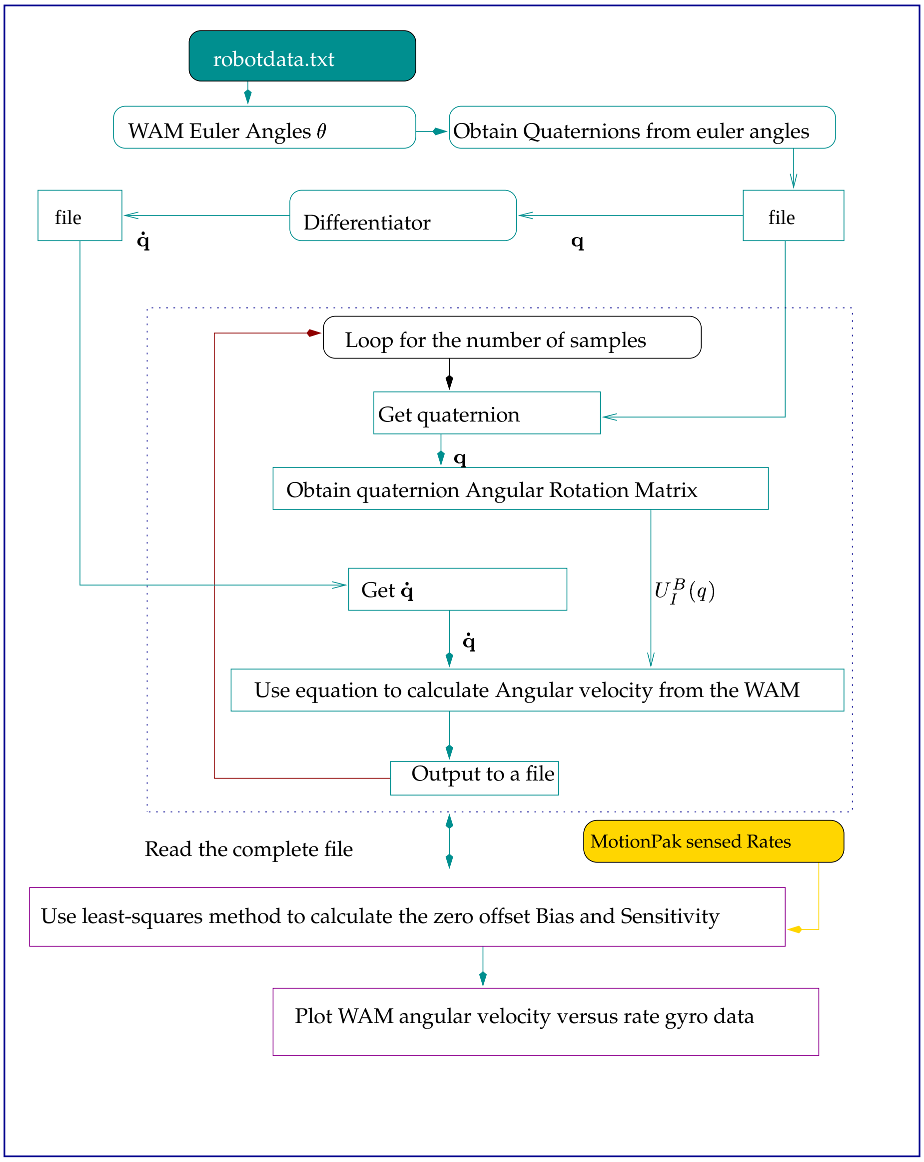


Figure 3.11: Method for obtaining angular velocity from the WAM "groundtruth" attitude data and comparison with the MotionPak sensed rates. Cyan boxes indicate manipulation of WAM data, whereas Gold framed boxes indicate operations concerned with the sensor output

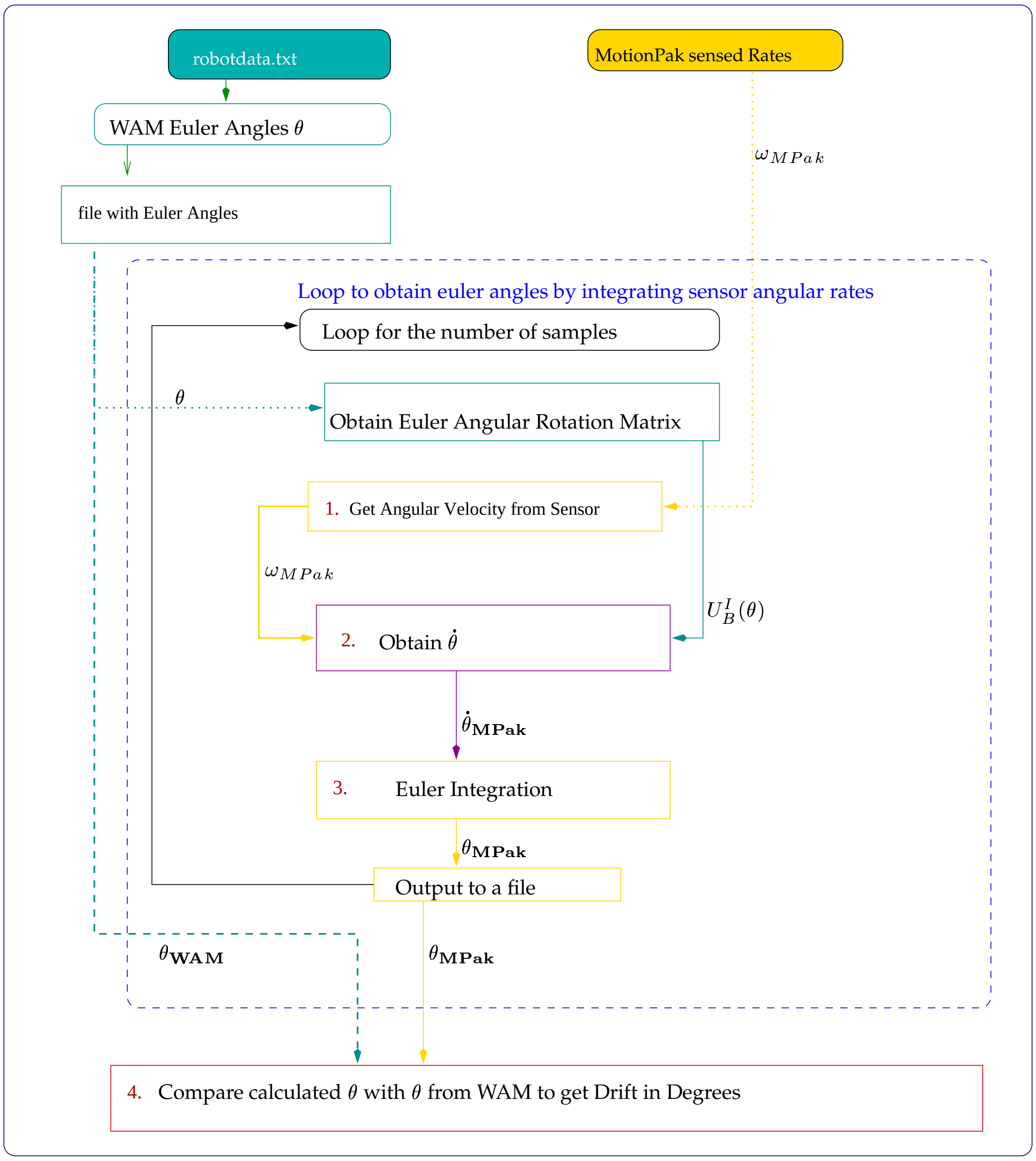


Figure 3.12: Method for obtaining Euler Drift from angular velocity

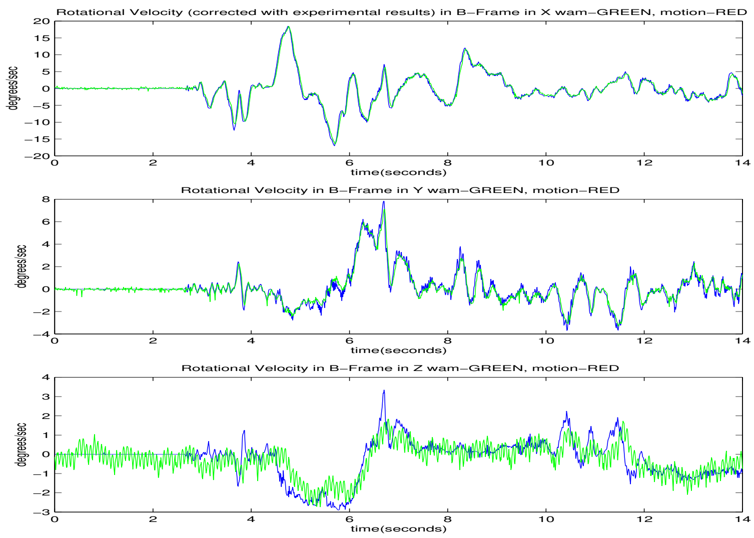


Figure 3.13: Rate Gyroscope Sensor Data with scale factor and bias implemented experimental calculations versus the WAM - ground truth angular rate data. Note the virtual overlap of the two plots.

Alternative method:

An alternative method in obtaining the Euler angle drift is by using quaternions. Although more complicated, an attempt at this method highlighted an implementation problem when a quaternion approach is taken[15]. This problem was due to computational errors in calculating the quaternion. One of the properties of a quaternion is that its magnitude must always equal one. However, it was found that in calculating the quaternion, the norm slid slightly off one [16]. This meant that it had to be normalised for each sample. It was noticed, however, that noise was introduced in the normalisation of the quaternions at each time step.

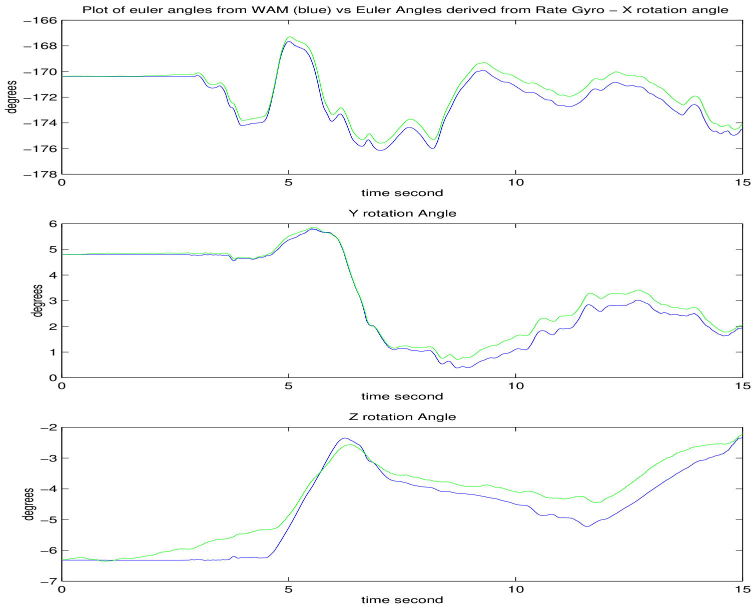


Figure 3.14: Rate Gyro Euler Drift Rates - Slow Movements/accelerations

3.7 Compass/Inclinometer Investigation

As discussed in Chapter 2, the Kambara sensor suite includes TCM2 module, which contains an inclinometer and a compass that can be used to determine the attitude of the vehicle. This section is concerned with the dynamic properties of the TCM2. The results will show the introduction of noise under acceleration & will also propose a method by which the TCM2 rate gyroscope readings can both be used in order to estimate the attitude of the vehicle.

[15]see equations 2.7 in section 2.5.2. These equations can be used to obtain the rate of change of a quaternion, which could then be integrated to obtain the quaternion parameters. Although observing drifts in the resulting quaternion parameters is not as intuitive as observing drift rates in Euler angles as rotations in Euler space are easier to conceptualise

[16]If this is not the case, then the resulting rotation matrices seize to be orthogonal

Method of Analysis

Figure 3.15, shows a block diagram of how the compass/inclinometer output was investigated. This figure is followed by a discussion on the individual blocks in the flow chart.

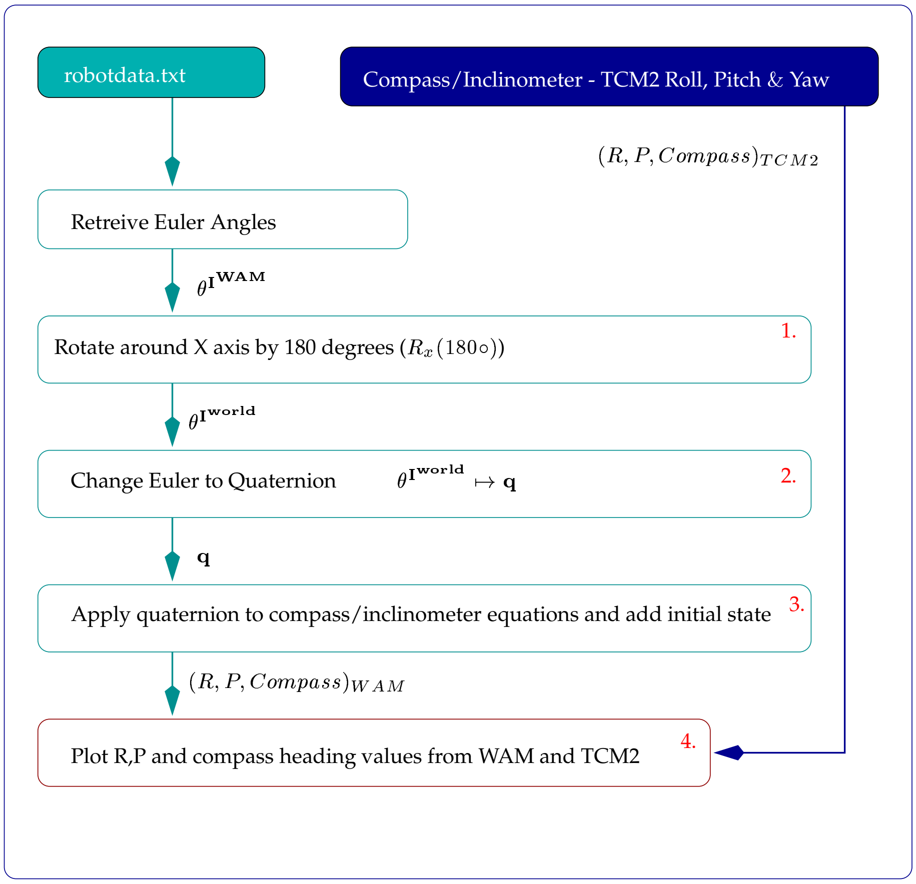


Figure 3.15: Flow Chart for Investigation of TCM2 compass output

In order to observe the performance of the sensor output, that is the roll, pitch and compass heading values, we need to have values of the same parameters that represent the real behaviour of the sensor. This can be achieved by mapping the Euler angle values from the WAM on to the three parameters presented by the TCM2.

In his thesis, Betlehem [Betlehem, 1999] has presented equations that give the roll, pitch and compass heading values from a given quaternion. These equations are also presented here.[17]

In order to obtain the ground truth values for the TCM2 output, the Euler angles from the

[17]Note that the final equation for the compass heading in Betlehems thesis was incorrect, however the derivation and the equations for P and R are correct. The corrected equations are provided in appendix **??**

WAM had to be converted to quaternion representation so that the sensor model equations in appendix F could be applied. However, there are two factors that had to be considered before this could be achieved.

- The first is in the derivation of these equations, the gravity vector was assumed to point in the positive Z direction in the inertial frame. Appendix D and also figure 2.5, however shows the Z axis from inertial frame at the base of the robot as pointing up, opposite to the gravity vector. Therefore, the Euler angles calculated by the WAM are with respect to the inertial I frame of the robot, and these angles need to be rotated by 180 degrees about the X-axis [18]. Once this operation is completed, the resulting Euler angles can be used to obtain the corresponding quaternion representation.
- The second point to note is that the inclinometer has a limited tilt range of ± 50 degrees. For a tilt angle larger than 50 degrees the inclinometer values saturate.[19] Therefore, it is desirable to use an experiment where the inclinometer remains within this range. In conducting the experiments to obtain the results presented here, care was taken not to saturate the inclinometer.

3.7.1 Results

The results highlight two major conclusions. The first is a confirmation of the correctness of the equations presented above for obtaining attitude estimates, and the second is concerned with the performance of the inclinometer when subjected to varying accelerations.

From Figure 3.16 it can be seen that the actual sensed tilt values for R and P follow the calculated true values from the WAM. This demonstrates that the derived equations that map quaternion estimates into the corresponding sensor parameters, produce the correct results. This result provided assurance in including the equations in the sensor model of the state estimation system.

3.7.1.1 TCM2 response when undergoing acceleration

Another point that can be deduced from Figure 3.16 follows from the discussion in section 2.2.4 in Chapter2. It was reasoned that due to the nature of the inclinometer it was expected that accelerations other than that of gravity would introduce errors in the output. This phenomenon is demonstrated in figure 3.16 by the noise associated with the inclinometer readings. Figure 3.17 shows the correlation between the residual of the two values and the absolute acceleration measurement. The graph shows that the peaks in the residual occured at times of high accelerations and more importantly in rapid changes in direction, previously referred to as sloshing.

[18] See step 1. in the flow-chart in Figure 3.15

[19] For this reason the atan function will be used as opposed to the atan2 function for calculating tilt values P and R.

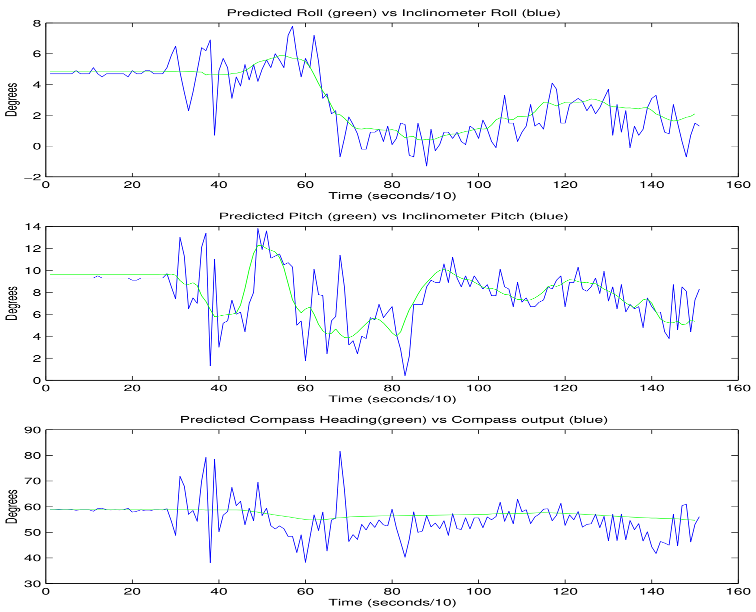


Figure 3.16: Inclinometer and Compass comparison between actual and real data - Slow Movements/accelerations

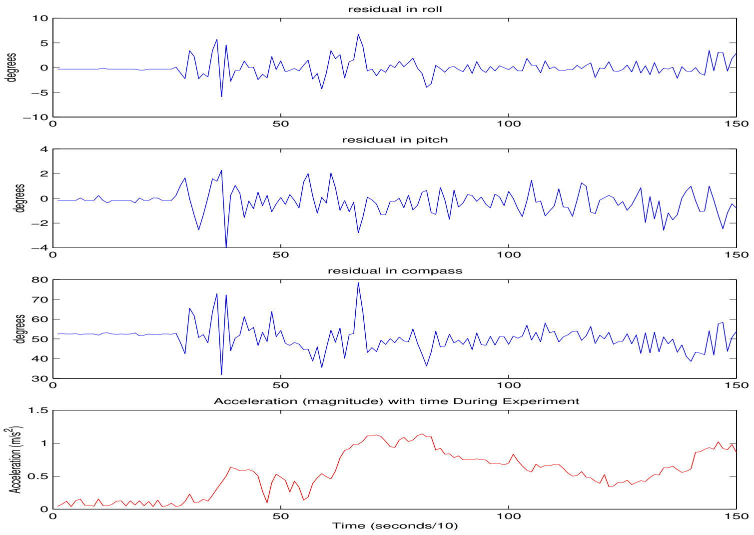


Figure 3.17: Inclinometer/Compass Residual and magnitude of Acceleration - Slow Movements/accelerations

3.7.2 Drift Properties of the Compass/Inclinometer

The attractive quality about an inclinometer and compass configuration is the fact that attitude can be measured with the same accuracy consistantly in a non accelerated state. This means that the inclinometer does not drift with time when there are zero accelerations. This is favourable as in the case where attitude estimates are required over longer periods of time, the inclinometer can be used to reset the attitude drift that is calculated from the rate gyro during times when the vehicle is at rest.

As explained in the previous section, it was possible to obtain quaternion estimates by integrating the rate gyroscope output directly. However, it was noted that due to bias characteristics in the sensor output, a drift was introduced in the resulting attitude estimate. Intuitively, this means that if the sub was to remain still for a period of time, and we were to obtain attitude information from the rate gyroscope, due to the drift in these estimates, we would have a representation of the vehicle as being upside down over a certain period of time.

It is not difficult to imagine the implications of such a case. Therefore, there is a trade off evident here. On the one hand. the rate gyro provides noise free and relatively accurate measurements over short periods of time, and on the other, the inclinometer is more reliable in its attitude estimates over the long term, although, producing noisy measurements when undergoing acceleration.

This trade off could potentially be modelled. That is, if the drift rates from the rate gyroscope were accurately known or an upper bound established, and similarly, if the inclinometer errors with respect to acceleration could be modelled, then we could determine a point of operation where the TCM2 would provide more accurate results than the rate gyro and vice-versa[20]

If the noise properties of both sensors were assumed to have white-Gaussian properties, then the KF would be able to automatically switch between the two methods of attitude estimation. However, given that some of the axis for the rate gyro were shown to have coloured noise properties, and that the inclinometer has a variable noise component that is related to the acceleration, it is necessary to add the model that was suggested above. This could be implemented in the EKF in the way off increasing the covariance (to a large value) of the source of attitude estimation that needs to be ignored. This will serve in effect as a 'switch' between the two methods of obtaining the attitude estimate.

[20] An equation that relates acceleration to tilt error is given by the manufacturer. For more information see the user's manual. This could be incorporated into the Kalman filter

Chapter 4

Static Sensor Testing

This chapter presents an examination of long term low frequency errors caused by sensor bias offset and drift. Long term logging sessions, under zero input conditions were conducted with the accelerometer and rate gyroscope sensors. The long term sessions were conducted several times for a period of 8 to 16 hours at each time. Special care was taken to isolate the proximate area where the experiments were conducted from disturbances due to human activity. The data from the sensors was sampled at a period of 1 second and the data was dumped to an ASCII data file at the completion of each experiment. The results are plotted and discussed in the following sections.

4.0.3 Objectives

The aim of these experiments is to determine whether a time correlated bias drift is present in any of the sensors.

4.0.4 Long Term Bias Behaviour

In investigating the rate-gyroscope's dynamic properties, it was noticed that there was a considerable day-to-day variation in the resulting bias for the rates measured in the X-axis. Ideally, zero-input conditions to both the inertial sensors should produce constant, non-varying sensor outputs. Static experiments have been carried out on the MotionPak sensors previously by Betlehem[Betlehem, 1999], however these tests resulted in a constant bias offset for the duration of the experiments. The plots presented in this chapter however, show this not to be the case for all of the sensors. Further investigation highlighted some problems with the hardware which was overlooked in previous tests. These problems, once rectified could have an effect on the bias values calculated through experimentation in Chapter 3.

Figure 4.1 is a plot of the accelerometer and rate gyro output for a period of 8 hours. The accelerometer plots (first 3 plots) show a constant bias for the duration of the experiment. The important results, however, can be found in the 4th graph which illustrates the X-rate sensor output. We see that despite the static conditions, this sensor starts at a lower value (of approximately 1V) and slowly settles to a higher value (of approximately 1.06 V) over a period of 2 hours. This behaviour was also prevelent in successive long term tests. This in effect provides an answer to the variable day-to-day offset bias of the X-Axis rate values which were noted during dynamic testing. From previous discussion in section 3.1, the bias value should be constant and insensitive to dynamic excitations. Therefore, in effect, the static offset value should equal the offset value calculated dynamically. Figure 4.1 however demonstrates a correlation of bias with time in the sensor that calculates angular velocity in the X-axis. This means that depending on the time the dynamic experiments were conducted once the sensors were activated, the bias values for the X-axis rate sensor differed (increased) the longer the time period. Note that this effect is isolated only in this particular sensor output.

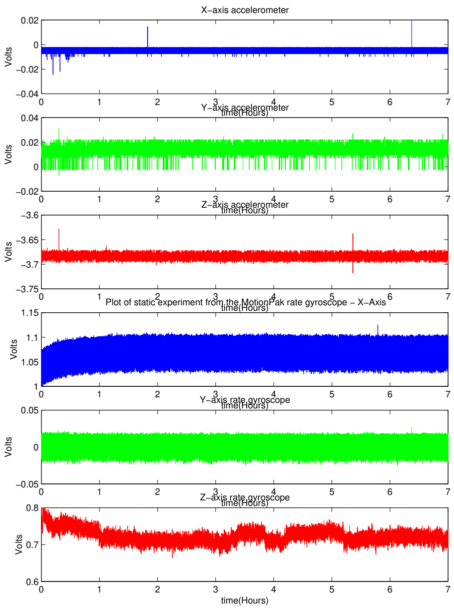


Figure 4.1: Plot of long-term static results. Note that the X-axis on the rate gyroscope takes approximately 2 hours to settle to a constant value.

The next logical step was to identify possible causes for this effect. Temperature effects were the first candidate for investigation. After logging ambient and temperature values from the MotionPak's own sensor, it was decided that there was no correlation present between changes in temperature and the behaviour of the sensor. This is coupled with the fact that the rate sensor is manufactured to be insensitive to temperature differences (see Appendix B).

Once temperature effects were deemed negligible, the next source of investigation was to look at the low pass filter which is used to filter out high frequency noise in the analog sensed data. The low pass filter was made up of a simple RC filter with a frequency cutoff of 50Hz(i.e half the sampling frequency of the MotionPak sensor so as to avoid aliasing). To isolate filtering effects, the raw sensor data was sampled without the low-pass(LP) filter. The results from the static tests without the LP filter showed a 1V drop in the previous X-axis offset bias value from the rate gyroscope[1]. This would indicate that in including the LP filter, an excess voltage drop was produced causing the DC offset of the output channel to increase.

Solution To Problem

After closer investigation of the analog-to-digital converter (ADC), it was ascertained that the cause of this unexplained excess bias in the sensor was due to a current flowing back from the ADC to the sensor causing a voltage drop across the series resistor of the LP filter. To rectify the problem, the resistor was omitted from the configuration. This is in line with the ADC documentation that suggests that an effective LPF can be approximated by using the output impedance of the sensor and a capacitor across each off the differential inputs. This problem was encountered late in the development of the thesis, therefore only preliminary experiments with the improved ADC configurations were able to be conducted. Figure **??** represents the results from the long-term static tests once the problem was rectified. We see from the 4th graph in the sequence that the time correlated bias drift in the X-axis of the rate gyroscope is no longer present. This means the hypothesis presented earlier stating that the temperature effects were discountable was correct and that the problem was solely in the wiring of the sensor/LPF/ADC circuit.

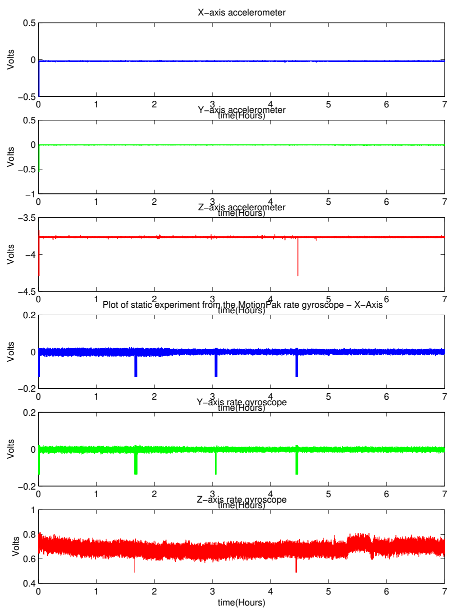


Figure 4.2: Plot of the accelerometer and rategyro output after the LPF has been modified. Note the X-axis on the rate gyro is constant for the duration of the experiment

[1]The other axes were found to exhibit the similar bias characteristics as before

4.0.5 Short Term Noise Behaviour

Implications on the Results
The problem discussed above is considered to have the following implications:

- Considerable change in the bias value was evident in the X-Axis rate gyroscope output compared to the results shown in Chapter 3 on the static tests. After close consideration on the effects on the individual channels, it was discovered that the majority of the current and hence the voltage drop ocurred on the channel that was sampling this particular axis of the MotionPak rate gyroscope sensor. This means that the results for bias calculated through the dynamic experiments presented in this thesis will not be valid for this axis of the rate gyro. Thus, possible changes to other bias values could have ocurred, however, are expected to be negligible. It is therefore necessary to carry out the dynamic experiments again so that new biases can be calculated[2].

- The dynamic effects such as the effects on the scale factor are considered to be negligible.

Behaviour from axis to axis for both sensors also varied. These undesired effects contribute a negative effect on the signal and hence the accuracy of the information that can be extracted from it.

Investigation of rate-gyro Z-axis output

The Z-axis rate sensor in both figures 4.2 and 4.1 shows a low frequency component in its static noise properties. This drift property of the bias does not seem to be periodic. The Z-axis angular rate output is reproduced in figure 4.3.

The effect of the small variations illustrated in figure 4.2 in the bias values has been shown to have consequential effects on the drift properties of the integrated yaw Euler angle, which is directly related to the Z-axis rate gyro values (see section 3.6 figure 3.14). Therefore, to minimize these drift effects on higher order values, it is desirable to identify the source of the problem.

Due to time limitation, only limited headway has been achieved in regards to this problem. Under the old circuit configuration, the raw sensor values were passed through a RC LPF, and similarly, in figure 4.2 the capacitor and impedance of the sensors, combined to produce an effective LPF. Given that the Z-Axis behaviour is prevalent in both of these cases, a common cause to the problem could be the inclusion or the characteristics of the LPF. However, when sampling was carried out directly from the raw output (i.e. bypassing the LPF), it was noticed the the Z-axis rate gyro output possessed a low frequency, periodic noise component that was not evident in any of the other output signals (see Figure 4.4). It is therefore possible that the LPF is attenuating the lower frequency noise component in the signal. This

[2]Note that this problem does not change the method of investigation presented in this thesis. It will only impact the final results. The programs developed for the dynamic experiments presented here can be applied to subsequent experiments concerning the investigation of these sensors

issue must be investigated further in future testing of the sensors. Eliminating this problem will lead to improved drift properties of the attitude estimates.

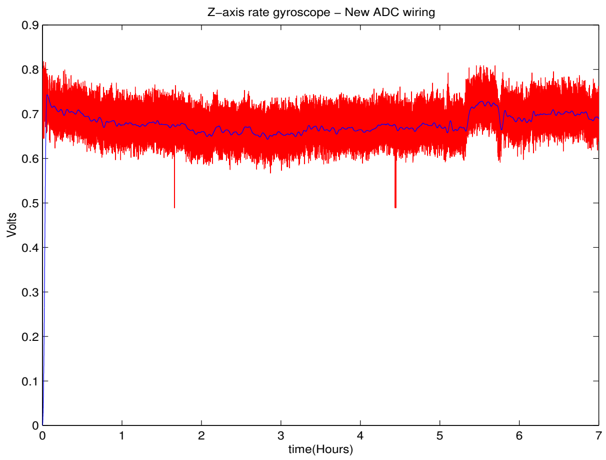


Figure 4.3: Plot of Rate Gyro Angular Velocity (RED) for the static experiment. Blue represents the low-pass filtered signal to highlight the drift property

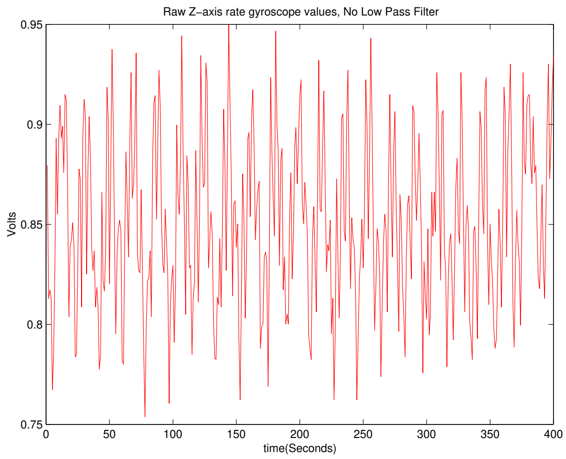


Figure 4.4: This is a short term plot of the Z-axis angular rate gyro. Note the low frequency oscilations. These could have effected the LPF output of the sensor

Chapter 5

Extended Kalman Filter

As stated in the introduction, a Kalman filter will eventually be used to perform the task of estimating the state of the Kambara. The Kalman filter is a set of mathematical equations that provides a recursive least squares solution to the problem of estimating state variables by fusing the information from the available sensors and a process model equations that reflect the dynamics of the vehicle. It is also adaptive to varying situations, such as the varying accuracy of the sensors under certain conditions[1].

The Kalman filter, when optimised, should provide better state estimates than those calculated directly from the sensor model. The level of this improvement in state estimation over an Inertial Navigation System (that uses only sensor information) is to be investigated in this chapter and in future work. The results from this analysis will prove relevant in the implementation of the reinforced learning control system that is to be tested on Kambara [Gaskett et al., 1999]. This control system is to be one which learns to control the AUV through experience of the real world. Therefore, no explicit model of the vehicle is to be given. Hence, it is desirable (from an experimentation point of view) to use only the sensor model state information in testing the control system. However, in doing so, it is also important to understand the limits of the sensor model over time. The results presented in Chapter 3 will provide insight into these limitations (eg by way of drift rates of the derived values)[2].

The tradeoffs that are to be considered in the choice of a state estimations system are summarised as follows:

- Calculating parameters that are required by the EKF for the non-linear process model (such as hydrodynamic drag coefficients) will be difficult from an experimental stand point.

[1]As mentioned in the section on the Inclinometer performance degradation undergoing acceleration or tilt angles above $\pm 50^\circ$. The Kalman filter in this case is able to adapt to these dynamic characteristics of the sensor and other components of the system model

[2]The accuracy from deriving state parameters(such as velocity and position) directly from the sensors was demonstrated in Chapter 3

- Fine-tuning the Kalman Filter parameters will prove to be a time-consuming exercise. This might mean that the state estimation system development will lag behind the other research efforts concentrated on Kambara, thereby potentially delaying testing in "real" environments.
- Once optimised, the EKF will provide better results than the INS(sensor model).
- The sensor model is easier to implement and testing can begin sooner.
- The sensors will provide more inaccurate information of the state over time on a stand-alone basis.

In light of these trade offs, the testing conducted off-line (such as that proposed in this chapter) will indicate the level of precision in state estimation that is to be gained by furthering research in the Kalman Filter. If this gain is considered to be outweighed by the factors outlined above, then focus is to shift to either improving the range of the available sensors or optimising the current configuration.

5.1 The discrete Kalman filter

Integral to the Kalman filter are the process & sensor models. The process model is concerned with the dynamics of the vehicle whereas the sensor model is a set of equations that relate sensor output to the state variables. The Kalman Filter requires the models incorporated to be linear. However, the models developed for the Kambara are highly nonlinear [Betlehem, 1999]. To deal with this problem, it is necessary to implement a modified Kalman filtering algorithm called the extended Kalman filter (or EKF). A critical component of this algorithm is the calculation of Jacobian matrices so that the nonlinear equations can be linearised. The calculation of such Jacobian matrices will be carried out once for each modification to either the sensor or process models by executing the "jacobian.m" Matlab file. Only the basic equations of the Kalman filter are presented here. For detailed descriptions on the Kalman Filter and the EKF algorithm see [Welch and Bishop, 1995] or [Brown and Hwang, 1985].

5.1.1 The Discrete Kalman Filter Algorithm

The Kalman filter addresses the general problem of trying to estimate the state $x \in \Re^n$ of a discrete-time controlled process that is governed by the linear stochastic difference equation

$$x_{k+1} = A_k x_k + B u_k + w_k, \tag{5.1}$$

with a measurement $z \in \Re^m$ that is

$$z_k = H_k x_k + v_k, \tag{5.2}$$

The random variables w_k and v_k represent the process and measurement noise (respectively). They are assumed to be independent (of each other), white, and with normal probability distributions

$$p(w) \sim N(0, Q), \tag{5.3}$$

$$p(v) \sim N(0, R), \tag{5.4}$$

The n matrix A in the difference equation 5.1 relates the state at time step k to the state at step $k+1$, in the absence of either a driving function or process noise. The n matrix B relates the control input $u \in \Re^l$ to the state x. The m matrix H in the measurement equation 5.2 related the state to the measurement z_k.

The properties of the noise processes are dictated by their covariance matrices **Q** and **R**. The parameters of the measurement noise covariance matrix, **R** are governed by the covariance properties of the sensors. These values can be calculated experimentally[3]. Assuming white-Gaussian properties of the sensor noise, the experimentally derived covariances would correspond to the diagonal elements of the matrix. In the case of **Q** (given by 5.4),however, the choice is less deterministic. This noise source is often used to represent the uncertainty in the process model (5.1).

A proposed method for estimating the process covariance matrix, Q

A possibility to to get "ball-park figures" for **Q** is possible through off-line experiments with available ground truth data. This can be achieved by passing the "ground truth" values through the process model equations and then obtaining the covariance from the residual of the results and the corresponding ground truth values. Even though this is feasible for cases where ground truth data is available, it would prove rather difficult to carry out the process for a nonlinear environment that the Kambara will operate in. This means that the process noise, **Q** might need to be changed dynamically during filter operation in order to adjust to different dynamics [Welch and Bishop, 1995].

In either case, whether or not a rational basis for choosing the parameters is applied, often, the optimisation of the filter performance can be obtained by "tuning" the filter parameters **Q** and **R**. This tuning is usually performed off-line.

To implement the Kalman filter algorithm in the Kambara, the EKF equations were developed in C++. The top level state estimation algorithm was developed by myself as well as the Matlab file used to create the Jacobian matrices necessary to perform the measurement

[3]A static experiment similar to the one conducted in chapter 4 can be used to determine the covariance of each sensor

and time update equations in the EKF. The final EKF algorithm was finally tested on the WAM experimental data. For a description of the algorithms and hardware related module necessary for the implementation of the EKF refer to appendix N.

In order to perform the tests, a process model had to be created for the WAM system. However, the aim of these experiments was to test the performance of the sensor model as it is the important component that will later be integrated into the final state estimation system. In view of this, the process model time updates were "ignored" in a sense by "injecting" enough uncertainty into the process model via the selection of $\mathbf{Q}$[4]. Preliminary testing of the Kalman filter algorithms has been carried out. The results from these experiments are as of yet inconclusive, and hence are not presented here.

5.2 Chapter Summary

This chapter presented the issues that are to be considered in the final implementation of the state estimation system. It described how insights gained from the dynamic experiments in Chapter 3 can be used to decide on the amount of future resources that should be allocated to the implementation of the EKF on the AUV.

The code necessary to perform the EKF algorithm has also been implemented and tested. During the process of ensuring that the developed algorithms functioned correctly, considerable insight was gained into the overall function of the EKF. Further work needs to be carried out however in optimising computational load and the speed of the state vector updates. These can be achieved in the future by conducting further experimentation with the existing EKF on the WAM.

[4]This can be achieved by allocating very high values (as large as computationally possible) to the components of $\mathbf{Q}$

Chapter 6

Conclusions and Future Work

The initial aims of the project were to carry out dynamic characterisation of the inertial sensors included in the AUV and to implement an off-line version of the EKF. Both of these aims have been successfuly completed through this project. In carrying out these aims, a considerable amount of knowledge has been gained in the fields of discrete signal processing, Kalman filtering and sensor characterisation. This project would not have been possible without the help of Kambara team members and other members of the RSL labs.

The list bellow outlines the contributions that were made through the duration of the project. Issues that need to be addressed in the future related to each contribution are also included in the list.

- An experiment was successfully carried out on the WAM that enabled dynamic testing of the inertial sensor characteristics that are to be used in the AUV. A method for carrying out the experiments and their limitations has been documented.
- Problems with the sensor sampling code were identified and corrected during the experimentation process. These included timing issues that produced quantization noise and along with problems with reading the temperature sensor in the MotionPak sensor module.
- The relevant signal processing and dynamic equations were applied in processing the experimental data from the WAM in order to obtain values that could represent the "ground truth" state of the sensors. The accuracy of the derived higher order values was paramount in obtaining the correct characteristics of the sensors.
- Filtering techniques were developed to optimise the accuracy of the resulting offset bias and scale factor sensor parameters. The effect of applying an acausal weighted average filter on differentiated data during off-line analysis was investigated and shown to improve the resulting sensor scaling components.
- A least-squares method was proposed for acquiring the scale factor and bias values for the inertial sensors. This method was proven to be successful in calculating these values. An extension to the experimental method is suggested where the estimation

method for bias and scale factor is made more generic and includes temperature effects. This method can be used in the future where inertial sensor characteristics are to be investigated.

- Analysis was carried out in examining the correctness of state parameters such as position, velocity and attitude calculated from the sensor output. The resulting drift on these values was shown to exhibit better properties that those calculated by using the manufacturer's data for bias and scaling.

- Static tests were carried out where I arrived at two important results, both of which were overlooked in previous sensor testing and could have led to unexpected if not detrimental results in the final implementation of the state estimation system in the AUV. As discussed in Chapter 4, further dynamic testing will have to be carried out in the future. This can be achieved by following the methods described in this thesis. The results should indicate similar scale factor values and lower biases. Drift properties might also improve as a consequence of the corrected circuit.

- Faults with the sensor equations developed for the compass/inclinometer have been identified and corrected. These functions have also been optimised in terms of necessary computations in the final implementation of the kalman filter by simplifying the trigonometric equations.

- The software necessary for running the Extended Kalman Filter was completed. The top level state estimation system module was written in C++ and a Perl program was developed in order to add flexibility in any future changes to the sensor or process models.

- The Kalman filter algorithm was tested iteratively with a simple one dimensional problem.

- A dynamic model was developed for the WAM/sensor plate system that can be used to test the AUV sensor model and real time properties such as timing and computational load of the Kalman filter code. Future effort should focus on this aspect. That is, to ensure that the EKF algorithm is tuned and the resulting values improve upon the drift values presented in Chapter 3. This would complete the dynamic characterisation of the complete inertial sensor model.

- A paper was submitted and accepted in the Australian Conference for Robotics and Automation (ACRA) to be held in August,1999 (see [J. Cvetanovski et al., 2000])

Bibliography

Beswick, 1998. Beswick, D. (1998). Inertial navigation system for an autonomous underwater vehicle. Technical report, Australian National University.

Betlehem, 1999. Betlehem, T. (1999). Autonomous submersible robot: State estimation system. Technical report, Australian National University.

Brown and Hwang, 1985. Brown, R. G. and Hwang, P. Y. C. (1985). *Introduction to Random Signals and Applied Kalman Filtering*. John Wiley Sons, second edition.

Craig, 1986. Craig, J. (1986). *Introduction to Robotics Mechanics and Control*. Addison Wesley Publishing Company, second edition.

Fjellstad and I.Fossen, 1994. Fjellstad, O. and I.Fossen, T. (1994). Position and attitude tracking of auvs: A quaternion feedback approach. *IEEE Journal of Oceanic Engineering*, 19.

Gaskett et al., 1999. Gaskett, C., Wettergreen, D., and Zelinsky, A. (1999). Reinforced learning applied to the control of an autonomous underwater vehicle. In *Australian Conference on Robotics and Automation*.

J. Cvetanovski et al., 2000. J. Cvetanovski, a. D. W., Brinsmead, T., Abdallah, S., and Zelinsky, A. (2000). A state estimation system for an autonomous underwater vehicle. Technical report, Australian National University.

Smith et al., 1999. Smith, R., Frost, A., and P.Probert (1999). A sensor system for the navigation of an underwater vehicle. *The International Journal of Robotics Research, Vol.18*.

Welch and Bishop, 1995. Welch, G. and Bishop, G. (1995). An introduction to the kalman filter. Technical report, UNC-CH Computer Science Technical Report 95-041, http://www.cs.unc.edu/ welch/kalman/kalman.html.

Williams et al., 1999. Williams, S., Majumder, S., Rosenblatt, J., and Durrant-Whyte, H. (1999). Autonomous transect surveying of the great barrier reef. In *Australian Conference on Robotics and Automation*.

Appendix A

Output format from WAM Experiment

File	Sensor	Data	Units	Range
robotdata.txt	WAM	Time Stamp	WAM Sample Period	
	WAM	Cartesian Position	meters[1]	1m
	WAM	Euler Angles	$^\circ$	$\pm\pi$
compassdata.txt	Inclinometer	Roll and Pitch	$^\circ$	$\pm\pi$
	Compass	Compass(yaw)	$^\circ$	0, 2π
	Magnetometer	Magnetic fields	μT	
		Temperature	$^\circ$	
	Warning Flags	Magnetometer Saturated	Boolean	
		Inclinometer Saturated	Boolean	
		Magnetic Field Distortion	Boolean	
motiondata.txt	Accelerometer	Acceleration	Volts	$\pm$7.5V
	Rate Gyroscope	Angular Velocity	Volts	

Table A.1: Data contained in WAM and Sensor Sampler output files

Appendix B

The Quartz Rate Sensor

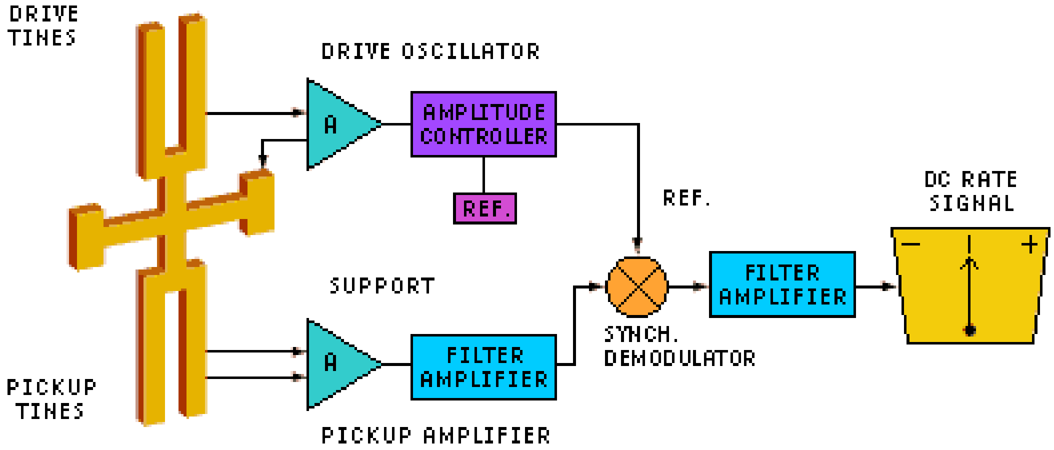


Figure B.1: Quartz Rate Sensor System

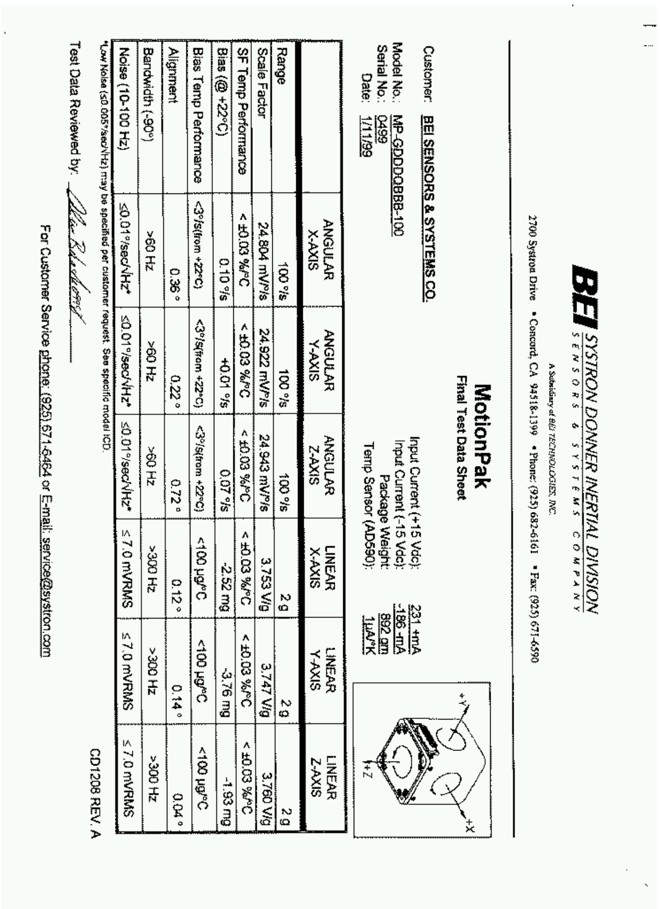
The GyroChip uses a vibrating quartz tuning fork to sense rate, acting as a Coriolis sensor, coupled to a similar fork as a pickup to produce the rate output signal. Each comprised of a pair of tuning forks, the GyroChip's along with their support flexures and frames are batch fabricated from thin wafers of single-crystal piezoelectric quartz. The piezoelectric drive tines are driven by an oscillator to vibrate at a precise amplitude, causing the tines to move toward and away from one another at a high frequency. This vibration causes the drive fork to become sensitive to angular rate about an axis parallel to its tines, defining the true input axis of the sensor.

Vibration of the drive tines causes them to act like the arms of a spinning ice skater, where moving them in causes theskater's spin rate to increase, and moving them out causes a decrease in rate. For vibrating tines ("arms"), an applied rotation rate causes a sine wave of torque to be produced, resulting from "Coriolis Acceleration," in turn causing the tines of the Pickup Fork to move up and down (not toward and away from one another) out of the plane of the fork assembly.

The pickup tines thus respond to the oscillating torque by moving in and out of plane, causing electrical output signals to be produced by the Pickup Amplifier. Those signals are amplified and converted into a DC signal proportional torate by use of a synchronous switch (demodulator) which responds only to the desired rate signals. The DC output signal of the GyroChip is directly proportional to input rate, reversing sign as the input rate reverses, since the oscillating torque produced by Coriolis reverses phase when the input rate reverses.

Information curtesy of Systron Donner.

MotionPak Manufacturer's Data Sheets



BEI SYSTRON DONNER INERTIAL DIVISION
SENSORS & SYSTEMS COMPANY
A Subsidiary of BEI TECHNOLOGIES, INC.
2700 Systron Drive • Concord, CA 94518-1399 • Phone: (925) 682-6161 • Fax: (925) 671-6590

MotionPak
Final Test Data Sheet

Customer: BEI SENSORS & SYSTEMS CO.

Model No.: MP-GDDDQBBB-100
Serial No.: 0499
Date: 1/11/99

Input Current (+15 Vdc): 231 +mA
Input Current (-15 Vdc): -186 -mA
Package Weight: 892 gm
Temp Sensor (AD590): 1µA/°K

	ANGULAR X-AXIS	ANGULAR Y-AXIS	ANGULAR Z-AXIS	LINEAR X-AXIS	LINEAR Y-AXIS	LINEAR Z-AXIS
Range	100 °/s	100 °/s	100 °/s	2 g	2 g	2 g
Scale Factor	24.804 mV/°/s	24.922 mV/°/s	24.943 mV/°/s	3.753 V/g	3.747 V/g	3.760 V/g
SF Temp Performance	< ±0.03 %/°C	< ±0.03 %/°C	< ±0.03 %/°C	< ±0.03 %/°C	< ±0.03 %/°C	< ±0.03 %/°C
Bias (@ +22°C)	0.10 °/s	+0.01 °/s	0.07 °/s	-2.52 mg	-3.76 mg	-1.93 mg
Bias Temp Performance	<3°/s(from +22°C)	<3°/s(from +22°C)	<3°/s(from +22°C)	<100 µg/°C	<100 µg/°C	<100 µg/°C
Alignment	0.36 °	0.22 °	0.72 °	0.12 °	0.14 °	0.04 °
Bandwidth (-90°)	>60 Hz	>60 Hz	>60 Hz	>300 Hz	>300 Hz	>300 Hz
Noise (10-100 Hz)	≤0.01°/sec/√Hz*	≤0.01°/sec/√Hz*	≤0.01°/sec/√Hz*	≤ 7.0 mVRMS	≤ 7.0 mVRMS	≤ 7.0 mVRMS

*Low Noise (≤0.005°/sec/√Hz) may be specified per customer request. See specific model ICD.

CD1208 REV. A

Test Data Reviewed by: [signature]

For Customer Service phone: (925) 671-6464 or E-mail: service@systron.com

Figure C.1: This is a scanned copy of the manufacturer's data sheet for the accelerometer and rate gyroscope

WAM Kinematics

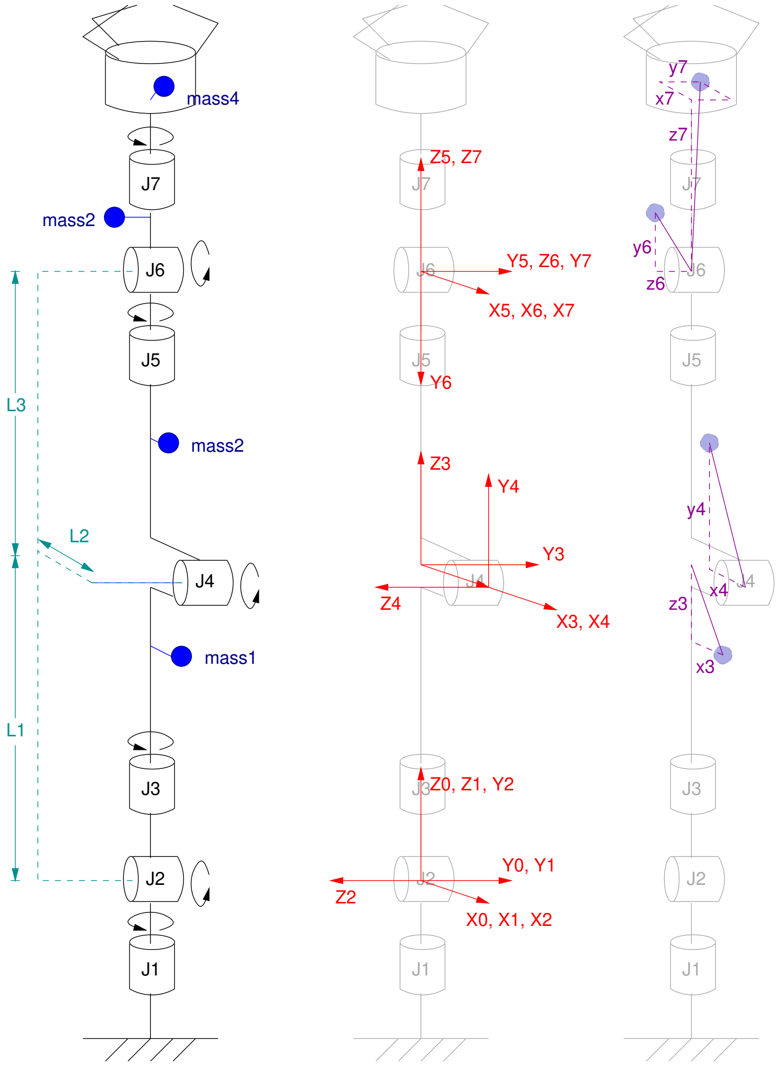


Figure D.1: WAM Kinematics

Appendix E

Pro Engineer Parameters

Inertia data for sensor suite 10/5/00

All distances in metres

elbow to compass rel to elbow

Distance =3D 0.435648;

dx=3D -0.124500, dy=3D 0.413500, dz=3D =0.0575000.

elbow to gyro rel to elbow

Distance =3D 0.450429; dx=3D -0.0990000, dy=3D 0.437000, dz=3D = -0.0460000.

elbow to sensor suite rel to elbow

Distance =3D 0.422049; dx=3D -0.0840600, dy=3D 0.413080, dz=3D = -0.0205950.

sensor suite to gyro rel to sensor suite

Distance =3D 0.0379577; dx=3D 0.0254050, dy=3D -0.0149400, dz=3D = -0.0239200.

sensor suite to compass rel to sensor suite

Distance =3D 0.0879454; dx=3D -0.0780950, dy=3D -0.0404400, dz=3D = -0.000420000.

MASS =3D 1.7056881 KILOGRAM=20

INERTIA at CENTER OF GRAVITY with respect to senor suite coordinate = frame: $(KILOGRAM * M^2)$

INERTIA TENSOR: Ixx Ixy Ixz 3.9038794e-03 7.0306293e-04 1.1226130e-03
Iyx Iyy Iyz 7.0306293e-04 5.6253084e-03 -7.0912478e-04
Izx Izy Izz 1.1226130e-03 -7.0912478e-04 6.1612419e-03

Appendix F

TCM2 sensor model equations

Given a quaternion,$\mathbf{q}(\eta, \epsilon_1, \epsilon_2 \epsilon_3)^T$ we have

$$\mathbf{P} = atan(\epsilon_1 \epsilon_3 - \eta \epsilon_2, \frac{1}{2} - \epsilon_1^2 - \epsilon_2^2), \quad \text{(F.1)}$$

$$\mathbf{R} = atan(\epsilon_2 \epsilon_3 + \eta \epsilon_1, \frac{1}{2} - \epsilon_1^2 - \epsilon_2^2), \quad \text{(F.2)}$$

and for the compass heading,

$$\psi = atan2(\epsilon_1 \epsilon_2 + \eta \epsilon_3, \frac{1}{2} - \epsilon_3^2 - \epsilon_2^2) \quad \text{(F.3)}$$

where Atan2 is a two-argument arc tangent function. [1] It must be remembered that the R and P angles measure the angles between the compass and the gravity vector, and they are not to be confused with the Euler angles θ and ϕ.

[1]Atan2(y,x) computes $tan^{-1}(\frac{y}{x})$ but uses the signs of both x and y to determine the quadrant in which the resulting angles lie. Atan2 can also be represented in terms of Atan and absolute value functions.

$$Atan2(y,x) = tan^{-1}(\tfrac{y}{x}) - \tfrac{\pi}{4}\Big(\Big(sign(x)-1\Big)\Big(sign(y)+1\Big) + \Big(sign(x)-1\Big)\Big(sign(y)-1\Big)\Big)$$

where $sign(x) = \frac{|x|}{x}$

This representation will be usefull in obtaining symbolic output from a Matlab file, as Matlab is unable to use symbolic constants in either $Atan2$ or signum functions. Note that the output from the equation shown above will output angles in the range $\pm\pi$.

Appendix G

Magnetic Field Distortion Sources in Workspace

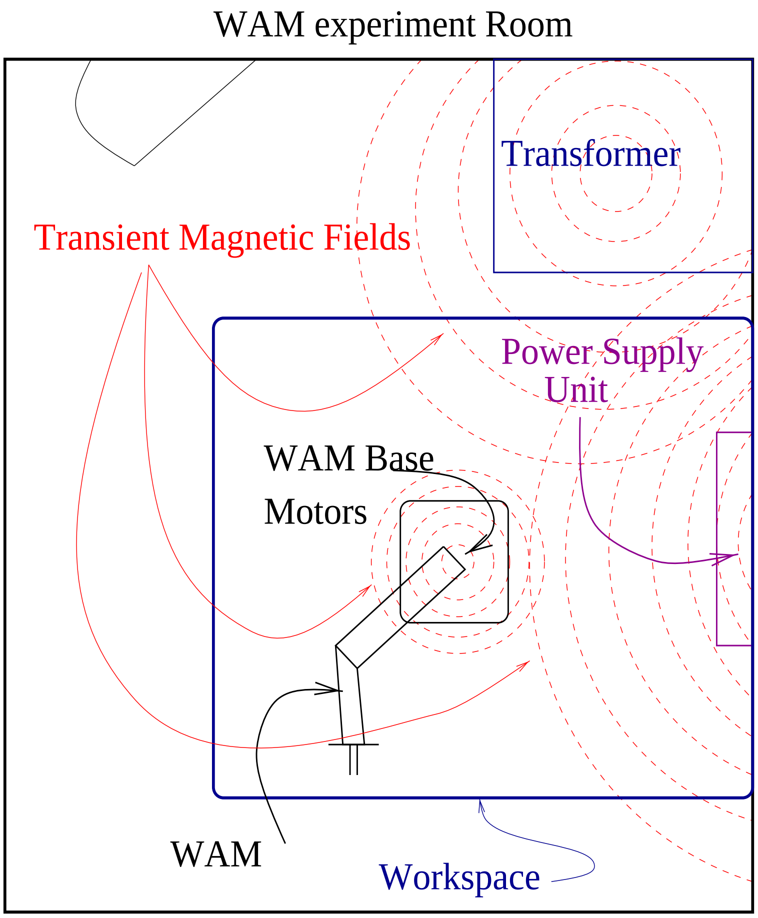


Figure G.1: This is a plan view of the workspace of the WAM. It illustrates the major sources of transient magnetic fields.

Appendix H

Experimental Tests

This is a list of the type of experiments that were carried out in order to test the characteristics of the inertial sensors.
Experiment List

- **1** movement in all directions (high a)
- **2** movement in all directions (high a)
- **3** movement in z (high a)
- **4** movement in x (high a)
- **5** movement in y (high a)
- **6** static z (upright)
- **7** static y (pointing down)
- **8** static x (pointing down)
- **9** slow motion in z direction
- **10** slow motion in y direction
- **11** slow motion in x direction
- **12** 1 meter movement (slow) and back in x
- **13** 1 meter movement (fast) and back in x
- **14** spin while upright clockwise and spin back (slowly)

Appendix I

Analysing the Accelerometer - file description

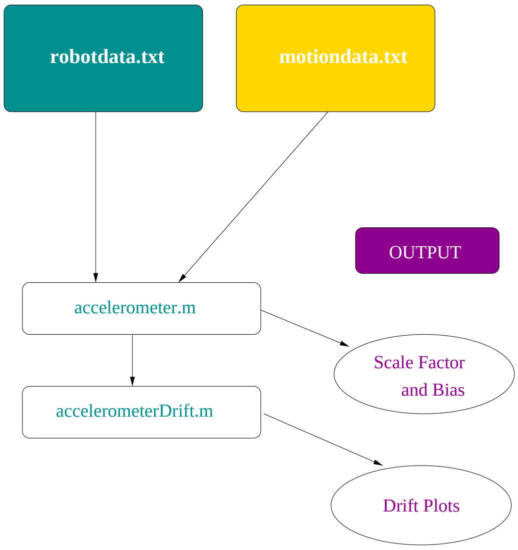


Figure I.1: Files that were neccessary in analysing the accelerometer output

Differentiator Matlab Code

This is a portion of the Matlab code used to calculate the accelerations from the WAM positions

⋮

this is a fourth order differentiating filter

[b,a] = butter(4,1);

num = 2R$_{S}amplingFreq * conv(b, [1 - 1])$

$den = conv(a, [11])$

$thissecondversionofadifferentiatorusesacutofffreq$

$acutofffreqencyof 0.05 = approximately 25Hz$

$[b1, a1] = butter(4, 0.05);$

num1 = 2*R$_{S}amplingFreq * conv(b1, [1 - 1])$

$den1 = conv(a1, [11])$

$Weightedaveragefilter$

$NWindow = 80$

$smoothImpulse = triang(NWindow);$

$bwa = smoothImpulse/sum(smoothImpulse);$

$awa = 1.0$

$thefollowingdemonstrateshowaccelerationswerecalculatedfortheIframe$

$differentiateallthedatawithoutLPFcutoff$

$XrobotVel = filter(num, den, robot(:, 2));$

$XrobotAccel = filter(num, den, XrobotVel);$

$differentiatealltthedatawithLPFcutoffsetattheappropriatevalue$

$XbutterVel = filter(num1, den1, robot(:, 2));$

$XbutterAccel = filter(num, den, XbutterVel(1 : Xlengthbuttervel));$

$XWAaccel = filter(bwa, awa, XbutterAccel);$

$\vdots$

Appendix K

Optimising the Weighted Average Filter

The weighted average filter optimal window size was determined experimentaly. This was done by adjusting the window size and observing the size of the dynamic noise covariance. The window size with the smallest noise covariance over all the axis was chosen for the filter. This was determined to be between 80 & 90 points.

Window Size	**CovX** (V^2)	**CovY**	**CovZ**
20	0.1278	0.1615	0.1223
60	0.0741	0.0757	0.0388
80	0.0607	0.0659	0.0178
90	**0.0571**	**0.0665**	**0.0166**
100	0.0550	0.0695	0.0214
120	0.0536	0.0798	0.0470
200	0.0692	0.142	0.2854

Table K.1: Data contained in WAM and Sensor Sampler output files

Appendix L

Analysing the Rate Gyroscope - file description

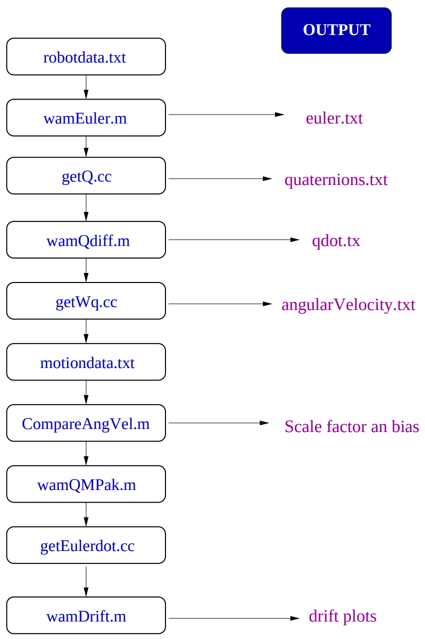


Figure L.1: Files that were neccessary in analysing the rate gyroscope output

Appendix M

Analysing the Compass - file description

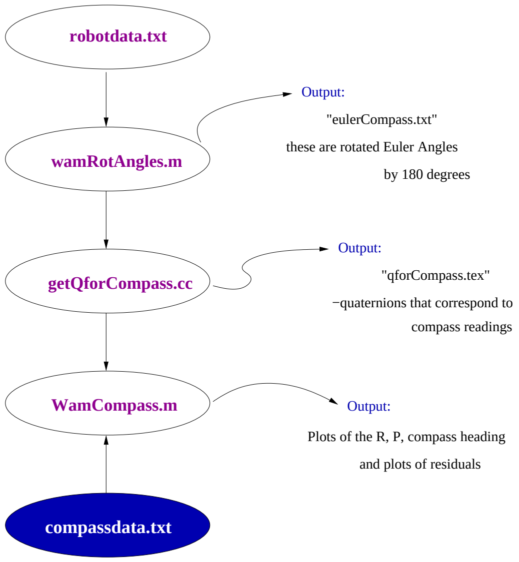


Figure M.1: Files that were neccessary in analysing the TCM2 module output

Appendix N

Description of EKF software modules

Figure N.1 describes the heirarchy of the code developed in order to implement the EKF, whereas figure N.2 outlines the order of execution of the modules. This is followed by a description of each individual component of the state estimation system.

N.1 Code Hierarchy

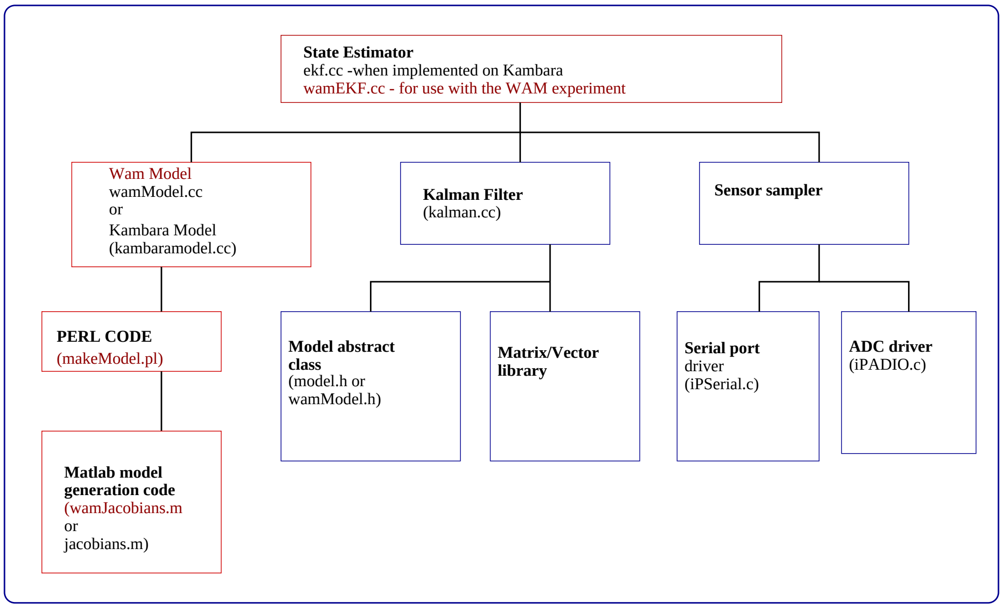


Figure N.1: Code Hierarchy for the State Estimation System - Boxes with a Red border indicate those components that were completed during the duration of my project

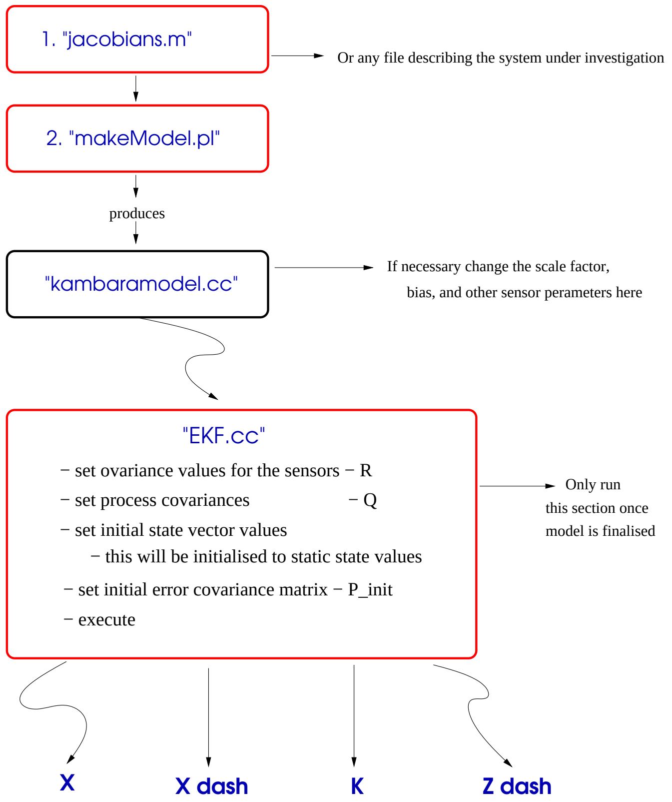


Figure N.2: This diagram illustrates the order in which the KF modules must be executed.

N.2 State Estimation and Kalman Filtering Code

N.2.0.1 State Estimator

This module runs in a loop which calls the sensor sampler to grab a set of sensor readings and update the covariance matricies to reflect which sensors have been sampled[1] and are in range. This recursive loop calls the Kalman filter module which inturn updates the state estimates through the model module[2]. The resulting state estimate and the raw sensor data are then copied into an appropriate memory area. The code for this module was written in

[1]Depending on the sampling speed, some sensors might provide "old" information. For example, if the state estimator was to cycle at a speed of 50Hz then the TCM2 would provide old information for four steps per cycle as it's maximum sampling speed is 10Hz.

[2]either kambaramodel.cc or wamModel.cc

C++. The current program is designed to run offline tests, which means that the sensors will not be sampled in real-time. Instead, sensor information is fed from the log files from previous experiments. This offline version of the SES is useful in that fine-tuning off the Kalman filter is easily achieved as manual experiments would be time consuming and cumbersome. Another disadvantage of fine-tuning the EKF in real-time would be that the experiments are not able to be repeated,(see Section2.5.4.1) therefore, the effect of varying the kalman filter parameters on the state output is more difficult to determine. Important information and the general function of the other modules shown in figure N.1 are discussed in detail in appendix **??**.

N.2.1 WAM model

This C++ class is created by the PERL program, "makeModel.pl", and holds the whole model of the WAM/sensor plate system. This model consistes of 6 member functions, 3 for the process model and 3 for the sensor model. The Matlab Model Generator generates the C code for all the functions related to the WAM. This can also be easily modified to generate the C code for the member functions related to the Kambara. As a first iteration of the state estimator, an assumption of additive Gaussian noise has been made by initialising these Jacobians to identity matrices[3]. Once this file has been created by the Perl program, the variables for the sensor scale factor, bias and temperature sensitivity parameters also need to be initialised. This file relies on the model that is being investigated, therefore, everytime the matlab model generator is modified, the Perl program must be used to re-create this file.

N.2.2 Kalman Filter

This C++ class performs the time update and measurement update steps of the extended Kalman filter algorithm. The model that is incorporated is inherited from the Model Abstract Class. A feature implemented into this class is a "fudge factor" which can be used to scale the state covariance matrix **P** if it is found that the algorithm has problems converging.

This class has been tested by implementation of a Kalman filter on a one-dimensional system This system involved a particle with a constant input force, a noisy position sensor and a noisy accelerometer. The data relevant for this test was generated by the Matlab file "kalmantest.m". The file "kalmantest.m" was also used to determine the true output of the system by simulating the Kalman filter algorithm on Matlab. After filtering the input sensor and force data by using the file "kalmantest.cc" it was found that the results from the C++ algorithm were the same as the matrix values as those from the Matlab file.

[3]for further information on this assumption see Betlehem[Betlehem, 1999]

N.2.3 Matlab Model Generator

This Matlab program generates the model equations required for the Kambara and WAM model. It converts these equations into C code and places them in an output text file.

N.2.3.1 Problems Encountered

In converting the Matlab functions into equivalent C functions, it was imposible to convert the compass model equations due to the fact that they require the use of the atan2 function. matlab is incapable in this case to convert the output of the atan2 function with symbolic constants. To get around this problem, an alternative representation for the atan2 function was developed that makes use of the atan() and abs() functions.

N.2.4 Model Abstract Class

This C++ class should be used to build any model that is intended for use with the extended Kalman filter class. It contains six member functions, 3 for the process equation and 3 for the measurement equation. Models used by the extended Kalman filter class should inherit from this abstract class.

N.2.5 Perl code generator

The Perl program "makeModel.pl" is designed to strip the C code of the output file from the Matlab Model Generator and places it into the WAM(or Kambara) model C++ class, "wamModel.cc"(or "kambaramodel.cc"). This allows for easy updating of the model if it happens to change.

N.3 Hardware related Programs

N.3.1 The Sensor Sampler

The purpose of this module is to sample all the sensors in an approximate simultaneous fashion at periodic intervals. The raw data is then passed up to the State Estimator. This program was written by Terence Betlehem and completed by Peter Brown. During the experimentation process, we were able to identify and consequently debug most of the problems related to this module. At present, there doesn't seem to be further problems with this code.

N.3.2 ADC Driver

This driver was written in C and its basic purpose is to periodically read the voltages on up to 16 ADC channels so as to sample the analog output of the sensors.

N.3.3 Serial Port Driver

This driver was written in C. It's purpose is to interface with the two serial ports on the serial board. The serial port driver will be used to interface with the TCM2 module[4].

[4]the TCM2 provides digital output

Appendix O

CD-ROM directory Structure

There are 6 directories in the CD-ROM. These are as follows:

- Thesis - contains all the files relevant to reproduce this thesis - LATEX
- Experiments - Holds all the experimental data from the experiments presented here - TEXT files
- Accelerometer - Holds the files necessary to obtain all the results for the accelerometer sensor - MATLAB files
- RateGyro - Holds the files necessary to obtain all the results for the rate gyro sensor - MATLAB & C++ files
- Compass - Holds the files necessary to obtain all the results for the TCM2 sensor - MATLAB & C++ files
- EKF - All the code associated with the state estimation system

Appendix P

WAM mount drawings

The following drawings of the sensor mount were created by Song Pho during his summer internship at RSL.