Modeling and Estimation for Maneuvering Target Tracking with Inertial Systems using Interacting Multiple Models

(M.Sc. Thesis)

Furrukh Sana

Dept. of Signal Theory and Communications,
Universitat Politecnica de Catalunya,
Barcelona, Spain.

Thesis Director: Prof. Juan A. Fernandez Rubio
Universitat Politecnica de
Catalunya (UPC), Barcelona, Spain.

Thesis Advisor: Dr. Pau Closas
Centre Tecnològic de
Telecomunicacions de Catalunya
(CTTC), Barcelona, Spain.

PREFACE

The aim of this Thesis is to study and develop Estimation Technique that enhances the Dynamic Tracking capability of Maneuvering Targets based using Inertial Systems. Inertial Measurement Systems have measurement biases and drifts and properly estimating their errors is a real time problem. Moreover, different targets perform different types of maneuvers during different stages of their trajectory and as such it is not possible to obtain accurate tracking of target maneuvers using a filters based on conventional single model approach. As such, a technique is required which is dynamic in both estimating and filtering the errors in inertial measurements and in switching to appropriate motion models according to the current maneuver of the vehicle. This thesis suggests and evaluates 'Interacting Multiple Models (IMM)' scheme for the solution to the above problem. Performance of the IMM scheme is proven over conventional single model based filters like Kalman Filter through both simulations and real target tracking.

ACKNOWLEDGEMENTS

I would like to extend my gratitude to Prof. Juan A. Fernandez Rubio for accepting to be the Director of my Thesis and helping me find this opportunity and develop my work at Centre Tecnològic de Telecomunicacions de Catalunya (CTTC), Barcelona, Spain. He has been utmost cooperative and understanding throughout the development. I would also like to thank him for providing me his valuable reference letters for applying to my PhD program.

I am immensely grateful to Dr. Pau Closas and Dr. Carles Fernandez-Prades at CTTC without the supervision of whom this thesis would not have been possible. Dr. Pau monitored the progress of this thesis very closely and provided insights at every stage and corrected the direction where ever necessary. Dr. Carles Fernandez provided very valuable ideas which resulted in increase of the quality of this thesis. Working with them at CTTC developed my research skills and enabled me to learn the techniques and approach of scientific methodologies. I am also thankful to them for encouraging me to write research publications. Also I would like to present my official gratitude to Centre Tecnològic de Telecomunicacions de Catalunya (CTTC), Barcelona, Spain for providing me the opportunity and the experimental equipment for this thesis.

I am grateful to the MERIT Masters and the Erasmus Mundus programs for providing the funding for my graduate studies. It has been a life enriching experience, both academically and culturally. I am thankful to my Professors at Politecnico di Torino, Torino, Italy and Universitat Politecnica de Catalunya, Barcelona, Spain for the knowledge they have provided me. Finally, a very heartfelt thanks to my Parents and Family who have been utmost supportive to me in my life and help me become today what I am.

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Problem Statement

Inertial Navigation Systems (INS) have been one of the most oldest and simplest forms of navigation systems. INS systems are based on Dead-Reckoning techniques. Dead-Reckoning techniques use an initial known point of origin and then record measurements of acceleration or velocity over time to obtain resultant position using integration. Although simple, INS systems are prone to accumulating errors over time resulting in very high drifts away from the true navigation solution.

Usually the technique adopted is to integrate INS systems with Global Satellite Navigation Systems (GNSS). INS and GNSS have complimentary error characteristics, but this too has its limitations. GNSS signals are very weak and noise corrupted as well as prone to jamming and spoofing. As such, dependence on GNSS for aiding purposes is not an option in many cases especially in indoor scenarios like robotic navigation or underground railway systems.

The Estimation techniques used for vehicle tracking using standalone INS systems have been based on Bayesian filtering techniques which use a particular vehicle motion model to predict the target motion. However, a vehicle's trajectory is based on different types of maneuvers during different stages and hence, a single model cannot provide accurate description of its behavior during the entire trajectory.

A technique is thus required which is dynamic in both estimating and filtering the errors in inertial measurements and in switching to appropriate motion models according to the current maneuver of the vehicle. The aim here is to reduce and eliminate the errors and increase the accuracy and reliability of navigation solution when using standalone INS systems which will enable new possibilities of using INS systems in multitude of applications.

To realize the importance of this work, consider the example in Fig. 1.1 and 1.2 of INS data collected over a period of 1 hour. The device, although stationary, experiences noise and bias offset. The data has been processed both in this raw form and using the filtering technique we have proposed in this Thesis. The error present in the raw data results in a drift of position estimate over time far away from the true value which would be origin (0,0) in this case. The estimate using the technique in this thesis is just around origin.

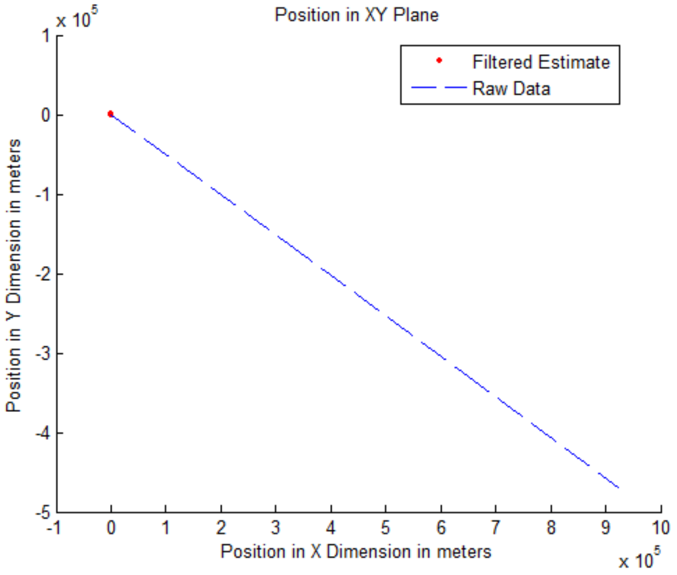


Figure 1.1: Position Estimate with Processing Raw Data of 1 hour of Stationary Mode

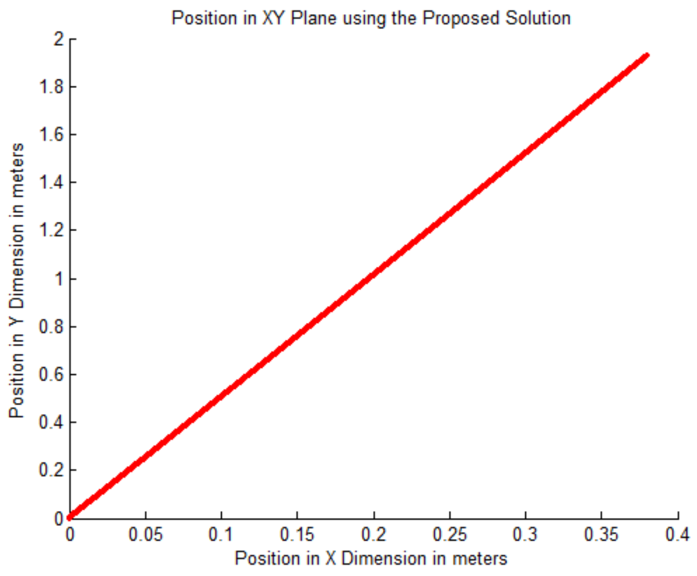


Figure 1.2: Position Estimate with the Proposed Solution of 1 hour of Stationary Mode Data

1.2 Thesis Outline

Chapter 2 provides the introduction to Inertial Navigation Systems. It provides in detail the types of INS systems and presents the explanation of their working and how to obtain navigation solutions using them. It also discusses the errors associated to INS systems and presents the analysis of the effects these errors have on the navigation solution. An Inertial Measurement Unit (IMU) device called ‘Inertia-Link’ is taken as a case and data recorded from it is used to show how the performance of a commercial INS device is deteriorated over time.

Chapter 3 is about filtering techniques and in particular to Kalman and Extended Kalman Filter. Their operational principals are explained and the reason of their selection. A Kalman Filter with a simple motion model based on Newtonian equations of motion is used to estimate the errors of the above mentioned INS device which shows that how much the performance of these filters is dependent on the motion models and the associated variances.

Chapter 4 provides a survey of Vehicle Dynamic Models. It provides some of the most basic and widely used motion models for vehicle tracking. Analysis is provided on the pros and cons of different models by testing them for different types of maneuvers. The effect of the important parameters of each model is analyzed. Based on this discussion, we develop the models which we will use in the next chapter with IMM scheme.

Chapter 5 introduces the 'Interacting Multiple Models' (IMM) algorithm and discusses its operation and advantages. We then integrate the models develop in previous chapters into the IMM and compare and analyze the results of the single model based filters and their performances to the result of the IMM approach. It shows that the IMM is capable of identifying the behavior of vehicle at a particular time and estimate it position using appropriate motion model which was not possible using just a single model approach.

In Chapter 6, we present the achievements by testing the performance of the solution we developed in chapter 5 with an experimental setup and perform target tracking of a small car on tracks. The performance is analyzed and compared to conventional filtering solutions.

Finally, Chapter 7 provides the conclusion of this study and presents the achieved improvement in performances of INS systems. It also suggests some potential improvements for future development.

CHAPTER 2

INERTIAL NAVIGATION SYSTEMS

2.1 Introduction

Inertial Navigation Systems (INS) have been present for a long time. Inertial Systems use Inertial Measurement Units (IMUs) which are typically equipped with tri-axes Accelerometers and tri-axes Gyroscopes which provide the measurement of Acceleration and Angular Velocity in 3 dimensions. These systems are based on Dead-Reckoning techniques. Using Dead-Reckoning, the position can be obtained by integrating the acceleration using Newtonian Equations of motion. Similarly, knowing the initial orientation, the Gyroscopes measure the angular velocity (or angular rate in case of digital systems) and provide the update on orientation.

Inertial Systems are used in wide range of applications from navigation of aircrafts and ships to tactical missiles and spacecrafts. However, INS systems are prone to many issues which affect their reliability and accuracy and hence limit its standalone use. The main issues are the Bias in the output values of accelerometers and gyroscopes, the noise present in the measurement sensors and the drift that occurs over time due to temperature changes, wear-n-tear and other factors. However, recent developments in MEMS technology have enabled devices which are smaller, light in weight and more reliable. These devices together with the signal processing techniques that have been developed have widened the range of possibilities for INS standalone operations.

In the following sections we will discuss the types of Inertial Measurement devices available and their characteristics.

2.2 Types of Inertial Systems

2.2.1 Gimbaled Configuration

In this configuration the accelerometer and gyro sensors are mounted on a platform using gimbals to isolate the IMU from any external rotation and keep it in alignment with the global frame. The global frame is the frame of reference in which our vehicle is moving while the body frame is frame of reference of the INS system. The gimbals allow rotation in all 3 axes. The Gyros provide the rotation information which is used to turn the motors, which in turn rotate the gimbals and align the IMU back with the global frame.

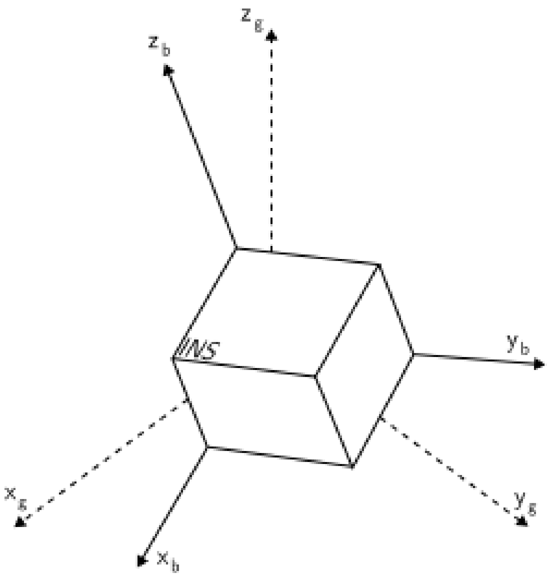


Figure 2.1: The Body and Global Frame of Reference (Adapted from [1])

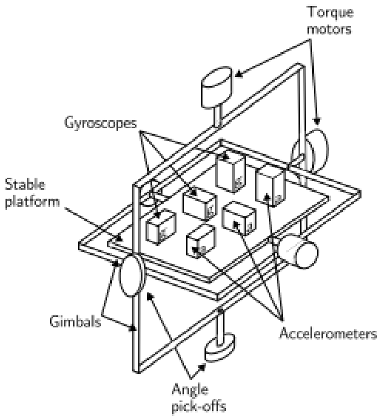


Figure 2.2: A Stable Platform IMU (Adapted from [1])

2.2.2 Strapdown Systems

These systems are mounted on a rigid platform and hence instead of the global frame provide the acceleration measurements in the local body frame of the IMU. For orientation, the outputs of rate gyroscopes are integrated and this orientation is then used to transform accelerations from body frame to global frame. The benefit here is that these systems are physically smaller in size and with less mechanical complexity but this comes at the cost of additional processing. However, since the computational devices have become faster, this downside of Strapdown systems has been overcome and such devices are therefore now preferred over gimbaled systems.

2.3 The Navigation Solution

To obtain the position information using Accelerometers and Gyros, we first, in the case of Strapdown INS devices, project the acceleration components from body frame to global frame using the rotation matrix. The elements of the rotation matrix are calculated using the values of orientation which are based on information from rate Gyros. This would be the case when using the Strapdown INS systems:

$$a_b(t) = (a_{bx}(t), a_{by}(t), a_{bz}(t))^T$$

$$a_g(t) = C(t).a_b(t)$$

Where $a_b(t)$ is the acceleration represented in the INS local body frame, $a_g(t)$ is the acceleration converted to the global frame and $C(t)$ is the Rotation Matrix at time *t.*

Then the component of acceleration due to gravity is subtracted. Now for the individual dimensions, we integrate once to obtain the change in velocity and add this to initial value of velocity and then integrate the new total velocity to obtain the change in position adding it to the initial value of position to obtain new complete position estimate using the following equations.

$$v_g(t) = v_g(0) + \int_0^t (a_g(t) - g_g(t))\ dt$$

$$s_g(t) = s_g(0) + \int_0^t v_g(t)\ dt$$

Where $v_g(0)$ and $s_g(0)$ are the initial velocity and position respectively.

2.4 Sources of Errors

The Error in the measurements of the IMU is the accumulated effect of various components of error sources. Some of these error components are compensated for during the manufacturing of the device and using on-board filtering that comes with modern day IMU devices. Details on all such sources can be found in references [1,3,4]. In the following, we concentrate on the main residual errors which are present in the final outputs of the IMU and need to be handled in real-time.

2.4.1 Bias and Drift

These are the most devastating of all error sources and amount for most of the part of residual error. Bias is a constant offset from the true measurement values. If bias is not accounted for in the accelerometer outputs, after double integration for position, the error will increase quadratically which can cause tremendous amount of drift from true position even in small period of time. For example, consider Table 2.1 which shows how even a small bias can drift the position estimate extremely far from the true value.

$$error = \frac{1}{2} bias.t^2$$

Bias (m/s^2)	**Error** *(meters)* t=100 seconds	**Error** *(meters)* t=10 minutes
.001	0.5	180
.01	5	1800
0.1	500	18000
0.2	1000	36000

Table 2.1: Effect of Acceleration Bias on Position

Similarly, drift in the angular rate information from gyros, if not accounted for; causes the IMU to falsely report rotation. Since when converting to body frame to global frame, the rotation matrix is used, any error in this matrix will cause the projection of the vertical component of acceleration, to be projected on the horizontal x-y plane and hence cause error in position estimate.

2.4.2 White Noise

The samples from the IMU sensors are corrupted with white noise sequence. This noise has a rate much greater than the sampling rate of the sensors [1]. The integration of these errors causes what can be called Angle-Random walk for Gyro errors and Velocity-Random Walk for Accelerometer errors. This noise can be of low variance but still needs to be eliminated using signal processing techniques to eliminate error in the position estimates.

2.4.3 Temperature Effects

The bias, drift and noise are sensitive to the temperature changes and as temperature change so does their values. The effect of temperature is more evident in scenarios where the device runs over long periods of time or is subject to sudden changes of environment/altitude. Fortunately, the device we used came with temperature compensation and we are using a real-time bias estimating technique so this error source does not affect our analysis.

2.4.4 Calibration Errors

Calibration Errors refer to accumulative effect of alignment errors, linearity problem, and scale-factor errors. Many of them are compensated during manufacturing and using on-board filtering of the IMU. However, if not compensated, they cause additional errors in position and orientation.

2.5 Inertia Link from MicroStrain

Inertia Link is the IMU device that we have used in our analysis. We have studied its characteristics and the errors present in its outputs and fed this information in our simulations. In this way, we have carried out different analysis by simulating the characteristics of an actual INS device.

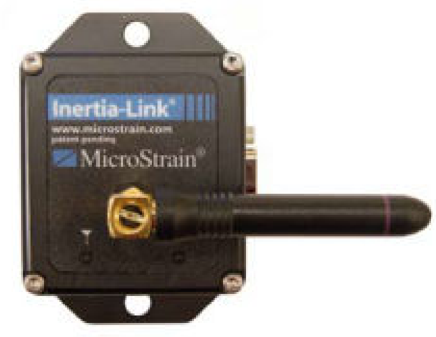


Figure 2.3: Inertia Link IMU Wireless Sensor Unit

The following information has been taking from the Inertia Link website:

> *"Inertia-Link® is a high-performance Inertial Measurement Unit and Vertical Gyro utilizing miniature MEMS sensor technology. It combines a tri-axial accelerometer, tri-axial gyro, temperature sensors, and an on-board processor running a sophisticated sensor fusion algorithm.*
>
> *Inertia-Link® offers a range of output data quantities from fully calibrated inertial measurements (acceleration & angular rate or delta-Angle &*

> *delta-Velocity vectors) to computed orientation estimates (pitch & roll or rotation matrix).*
>
> *All quantities are fully temperature compensated and corrected for sensor misalignment. The angular rate quantities are further corrected for G-sensitivity and scale factor non-linearity to third order."*

Some of the Inertia Link specifications from the device datasheet are:

Available Outputs	Acceleration and Angular rate, delta-Angle and delta-Velocity, Euler angles, Rotation matrix.
Accelerometer Range	accelerometers: ± 5 g
Orientation Range	360° about all axes
Accelerometer Bias Stability	± 0.005 g
Accelerometer non-linearity	0.2%
Gyro Range	gyros: ± 300°/sec
Gyro Bias Stability	± 0.2°/sec
Gyro non-linearity	0.2%
Orientation Accuracy	± 0.5° typical for static test conditions ± 2.0° typical for dynamic (cyclic)test conditions & for arbitrary orientation angles
Output Data Rates	1 to 100 Hz

Table 2.2: Available Outputs from Inertia Link and their Specifications

2.6 Inertia Link Output Analysis

The Inertia Link IMU comes with a software package which allows collecting basic output data from the inertial sensors. The software has different modes to record different parameters. We used this software to record the IMU data in static condition i.e. when the IMU device was held stationary on a stable platform. This procedure was helpful in determining the behavior of the outputs and the errors present in them as during the static condition the accelerometers should read zero acceleration and any variations would be due to the noise. Similarly, the values of orientation parameters namely Pitch, Roll and Yaw (Heading) should also be stable.

However, as expected, the performance of IMU was prone to errors mainly bias and measurement noise from sensors. The analysis of all those is presented below.

2.6.1 The Orientation Parameters:

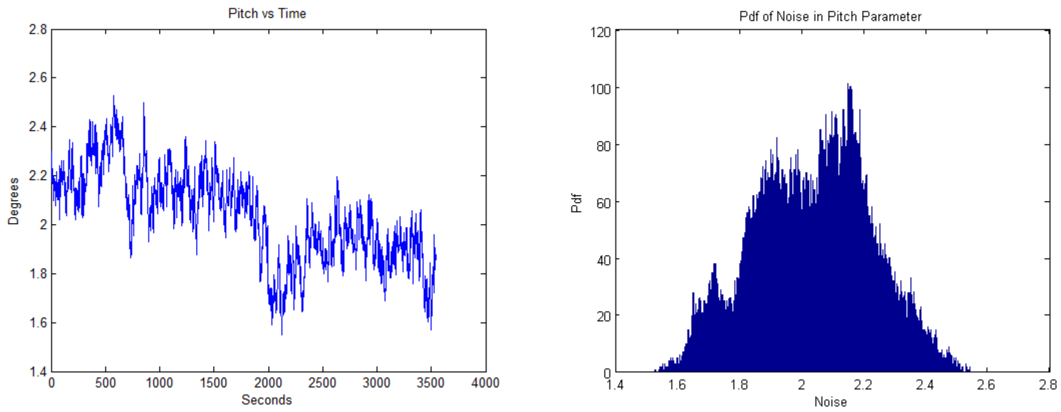
The graphs below show the recorded variations in the orientation parameters when the device was in static condition. Ideally these parameters should have a constant value but in this case they are subjected to variance over time. The variations in the Pitch and Roll parameters are also due to the instability and inaccuracy of the device as mentioned in the specifications. The main problem is with the Heading parameter, as can be seen; that the values of the Heading are continuously cyclic from -180 to 180 degrees. This total instability of the Heading parameter is explained in the Inertia Link manual to be the result of lack of magnetometers in the Inertia Link which are required for accurate Heading information. In this condition the Heading output available is not usable as the PDF of the noise is uniform which is the worst PDF for estimating errors. Figure 2.4 shows 1 hour of recorded data and the respective noise PDFs.

2.6.2 Acceleration and Angular Velocity

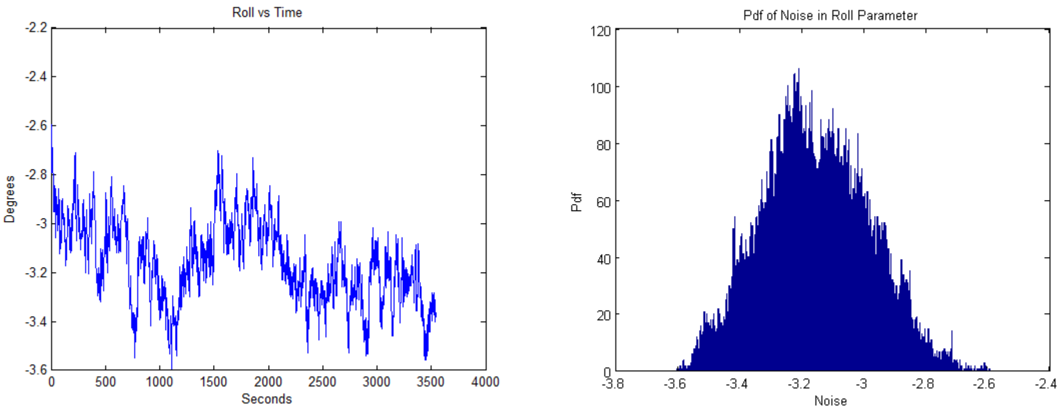
The Acceleration is measured by the Inertia Link in three dimensional components. The two components from the horizontal plane X and Y and the third representing the vertical axis Z which is subjected to the Earth’s gravity in addition to any forced acceleration.

Figure 2.5 shows the realization collected with the IMU device in the static condition. This data is useful as it provides information about the noise present in the three components of acceleration. The noise, although it may seem small, has detrimental effect on the accuracy of position estimate as shown in Table 2.1. In Fig 2.6 the noise was compared to a Gaussian noise with same mean and variance to see if it can be modeled as Gaussian when using signal processing techniques to eliminate it. Similar analysis is done in Figures 2.7-2.8 for Angular Rate.

Pitch



Roll



Heading

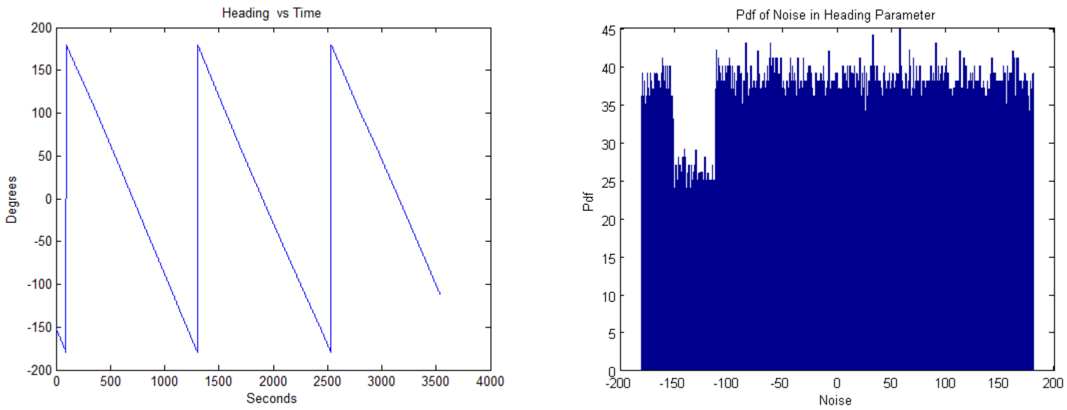


Figure 2.4: The Parameters of Attitude in Stationary State and their Noise Analysis

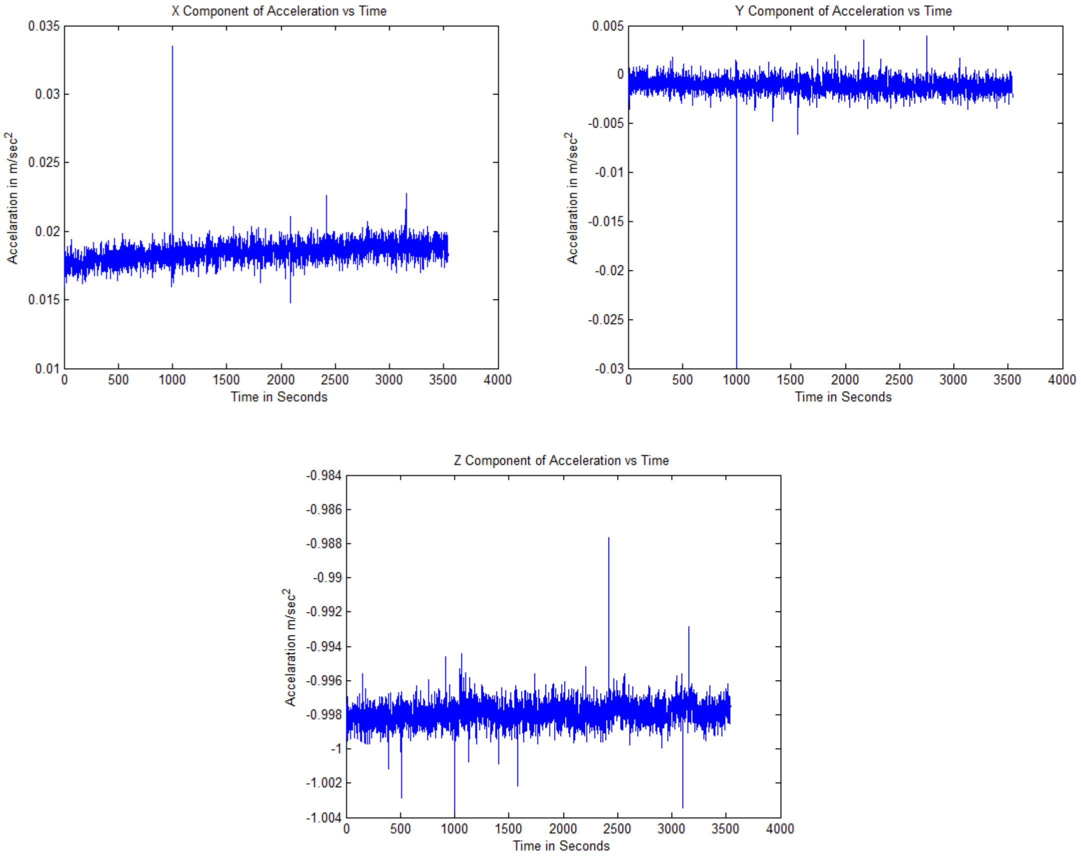


Figure 2.5: The variations in 3 dimensional Acceleration measurements in IMU stationary state

Comparison with Gaussian Noise

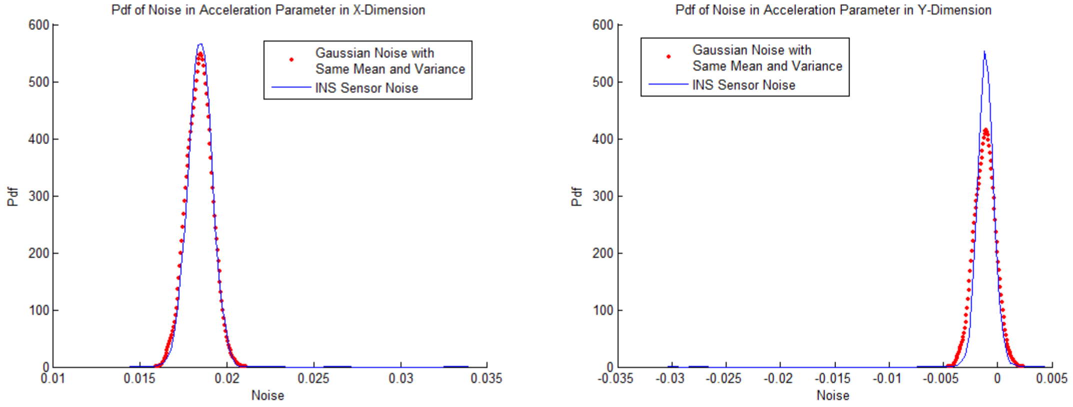


Figure 2.6a: Comparison of Noise in Acceleration measurements with Gaussian Noise

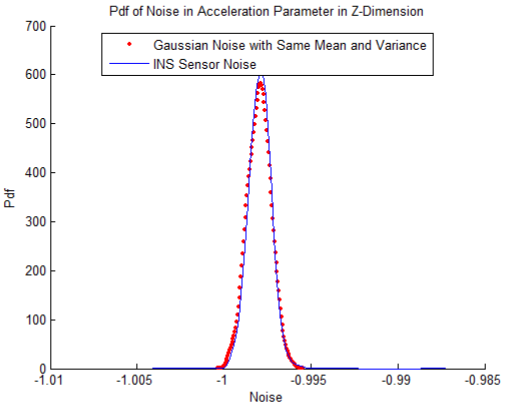


Figure 2.6b: The comparison of Noise in Acceleration measurements with Gaussian Noise (Continued from last page)

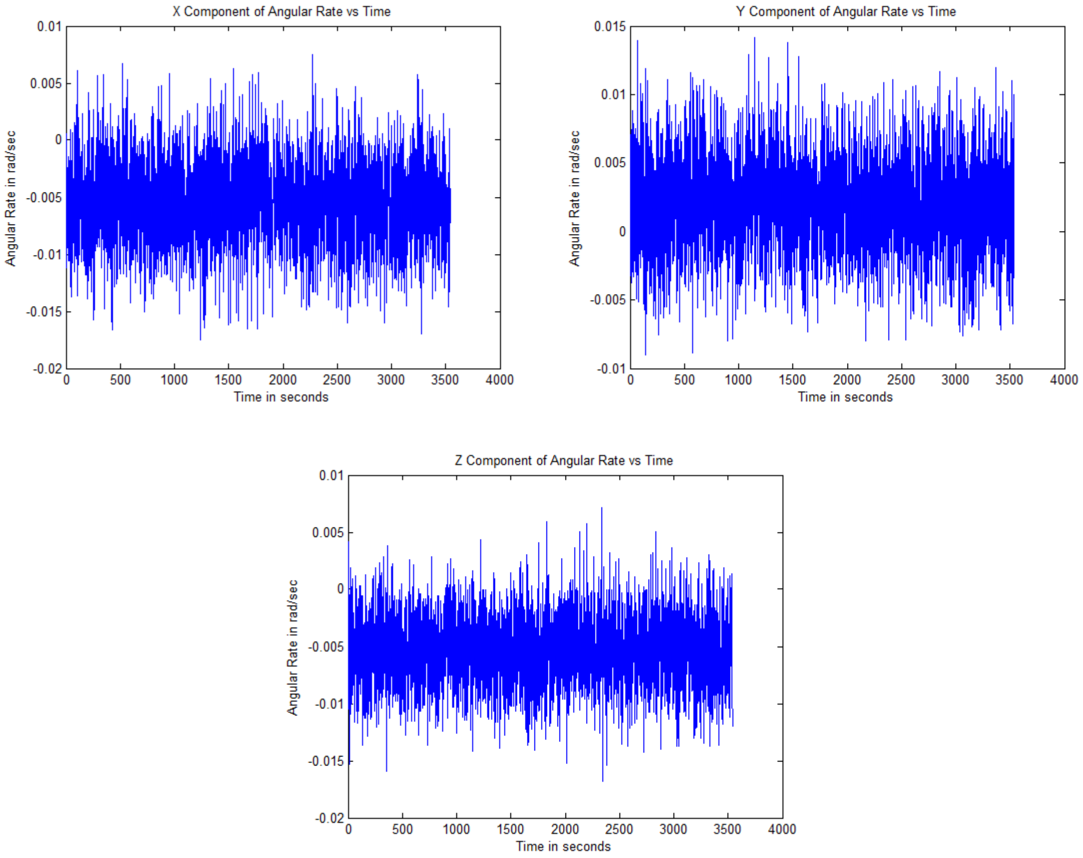


Figure 2.7: The variations in 3 dimensional Angular Rate measurements in IMU stationary state

Comparison with Gaussian Noise

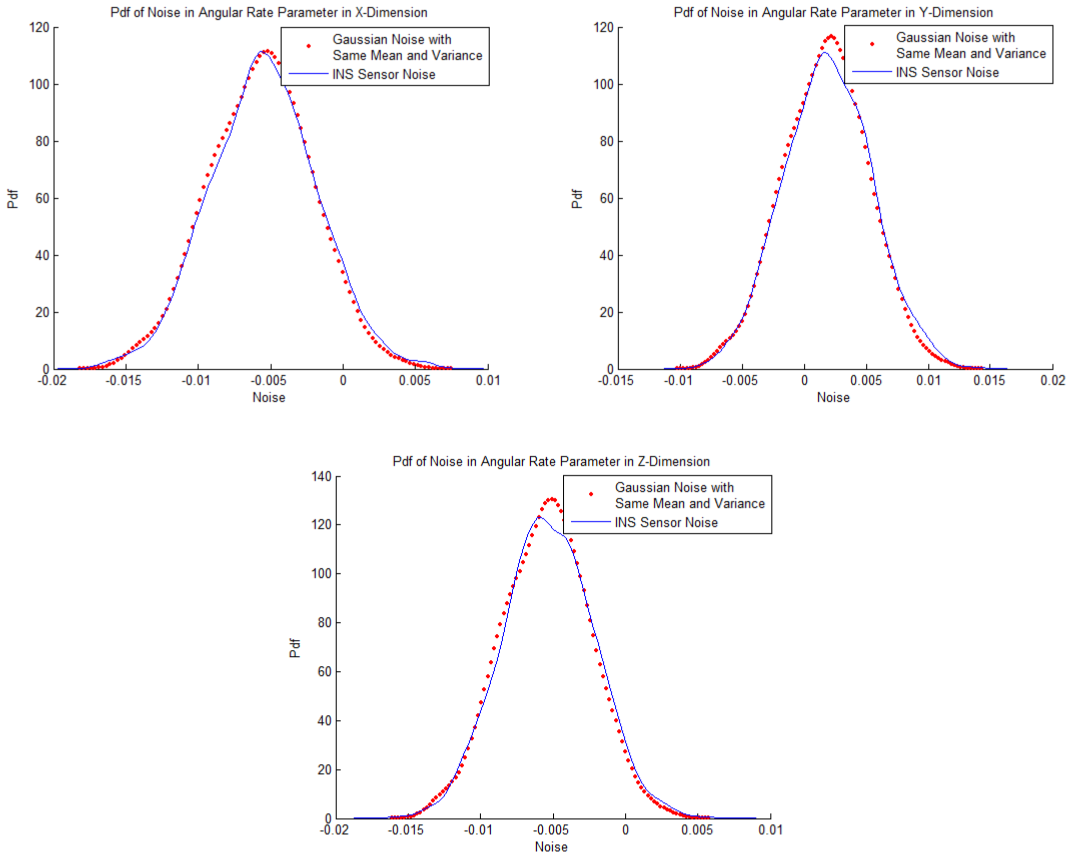


Figure 2.8: Comparison of Noise in Angular Rate measurements with Gaussian Noise

From the above graphs, it is evident that the noise has a non-zero mean which indicates Bias Error. The noises in different parameters have been compared to Gaussian noises with respectively same mean and variances. We performed different test such Anderson-Darling test and Kolmogorov–Smirnov to test its Gaussianity. The noise in Angular Rate parameters was found to be Gaussian but for Acceleration outputs the tests failed. But, even though the noise is not strictly Gaussian, as can be seen, the difference between actual distributions and Gaussian distributions is not large and as such we can approximate these noises as Gaussian Noises. This is a big advantage as when using Bayesian Filtering techniques for estimating the errors, the Gaussian assumption simplifies the problem to a great deal and simple filters such as Kalman and Extended Kalman filters can be employed.

The above analysis shows the Error characteristics in the different outputs of a typical IMU device. Our main focus is in obtaining the position estimate of the vehicle rather than its orientation. Therefore, further onwards; we will be treating our vehicle as a Point Target concerned with its position in 2D XY Plane. However, the principals of filtering discussed in Chapter 3 also hold valid in detecting the Bias and Noise in Orientation and Angular Velocity values.

CHAPTER 3

KALMAN FILTERS

The choice of filtering technique to eliminate noise depends on the nature of noise present in the system. Kalman Filters are based on the assumption of Gaussianity and if noise is not Gaussian then other filters such as Particle Filters are employed. As the analysis done in Section 2.6 of Chapter 2 indicates that the noises present in the INS sensors outputs can be approximated as Gaussian, Kalman Filter was chosen to process the Raw INS data and estimate the errors in measurements and provide position estimates.

Kalman Filter is an estimation technique based on Bayesian estimation principles. Its purpose is to predict the true values of measurements and State values from those values of measurements which are corrupted by noise and other inaccuracies. State values are the parameters that are calculated by Kalman Filter and are related to the measured quantities. Kalman filter works by predicting the future values of measurements and state values based on the measurements available up to the current point in time, the system dynamical model, the control inputs and specified noise variances.

The system dynamical model defines the expected evolution of the state values in relation to each other and measurements. The system model also defines the expected amount of variation in state values from one estimate to another using variance of what is called ‘Process Noise’. Also, the expected variance of the noise induced into the true measurements values in the sensors is defined using variance of ‘Measurement Noise’. Both these noises are assumed to be Zero-mean Gaussian Noise processes and the performance of Kalman Filter is based heavily on the principle of Gaussianity.

As it predicts, Kalman Filter also computes the uncertainty of that estimate. It then updates and corrects any error in that estimation as the future measurement values are made available. The Kalman filter averages a prediction of a system's state with the new measurement using a weighted average. The purpose of the weights is that values with better estimated uncertainty are more reliable. The weights are calculated from the covariance of the estimates which provide a measure of the estimated uncertainty of the prediction of the system's state. The result of the weighted average is a new state estimate that lies in between the predicted and measured state, and has a better estimated uncertainty than either alone.

The proper specification of the dynamical model and the variances are important factors in Kalman filter performance. Also, if the system in non-linear, then this simple form of Kalman Filter has to be altered to accommodate system non-linearity resulting in what is called the ‘Extended Kalman Filter’ (EKF).

In the following sub-section we provide equations for Vector State and Vector Observation Kalman Filter which is most widely used form of Kalman Filter. A detailed treatment of all cases and their derivations can be found in [12]. Also, an overview of EKF is provided.

3.1 Vector State Vector Observation Kalman Filter

This implies that measurements from more than one sensor are available and also the number of states to estimate based on those measurements; are also more than one.

The equation that defines the evolution of states is called the 'State Equation' which in this case is given by:

$$\mathbf{s}[n] = \mathbf{A}\mathbf{s}[n-1] + \mathbf{B}\mathbf{u}[n] \qquad n \geq 0$$

Where **A**, **B** are known $p \times p$ and $p \times r$ 'State Transition' and 'Input Control' matrices respectively, **u**[n] is a vector WGN with $\sim N(0,Q)$ and **s**[-1] $\sim N(\mu_s, C_s)$ and is independent of **u**[n]. **Q**[n]represents the covariance matrix of process noises. Here **A** represents the System State Model or System dynamical Model which describes the evolution of states from one instance to another. **B** is a vector of control inputs which relate the **u**[n] to the state values and is optional.

The equation that relates the measurements to the states is called the 'Measurement Equation' and is given by:

$$\mathbf{x}[n] = \mathbf{H}[n]\mathbf{s}[n] + \mathbf{w}[n]$$

Where **x**[n] is the $M \times 1$ vector of measurement variables, **H**[n] is a known $M \times p$ matrix that relates states to the measurements from M sensors and **w**[n] is zero-mean WGN noise which represents the error in the measurements. The elements in **w**[n] are independent of each other and **u**[n] and **s**[-1] and **w**[n]$\sim N(0,\mathbf{C}[n])$. **C**[*n*] here is the covariance matrix of measurement noises.

The Equations of a single iteration of Kalman Filter's Estimation process are then given by:

Prediction:

This involves predicting the next value of State vector at time n using the State Transition Matrix and the previous value of State vector at time n-1.

$$\hat{\mathbf{s}}[n \mid n\text{-}1] = \mathbf{A}\hat{\mathbf{s}}[n\text{-}1 \mid n\text{-}1]$$

Minimum Prediction MSE Matrix $(p \times p)$:

Calculates the uncertainty of the prediction made in the previous step

$$\mathbf{M}[n \mid n\text{-}1] = \mathbf{AM}[n-1 \mid n-1]\mathbf{A}^T + \mathbf{BQB}^T$$

Kalman Gain Matrix $(p{\times}M)$**:**

Calculate the weight according to which correction is made to the previous estimate

$$\mathbf{K}[n] = \mathbf{M}[n \mid n\text{-}1]\mathbf{H}^T[n](\mathbf{C}[n] + \mathbf{H}[n]\mathbf{M}[n \mid n\text{-}1]\mathbf{H}^T[n])^{\text{-}1}$$

Correction:

Correction of predicted **s**[n] based on new measurements and the Kalman Gain

$$\hat{\mathbf{s}}[n \mid n] = \hat{\mathbf{s}}[n \mid n\text{-}1] + \mathbf{K}[n](\mathbf{x}[n]\text{-}\mathbf{H}[n]\hat{\mathbf{s}}[n \mid n\text{-}1])$$

Minimum MSE Matrix $(p \times p)$ **:**

Calculates the uncertainty of the new estimate.

$$\mathbf{M}[n \mid n] = (\mathbf{I}\text{-}\mathbf{K}[n]\mathbf{H}[n])\mathbf{M}(n \mid n\text{-}1]$$

Where the mean square matrices are defined as

$$\mathbf{M}\ [n \mid n] = E[(\mathbf{s}[n] - \hat{\mathbf{s}}[n \mid n])(\mathbf{s}[n] - \hat{\mathbf{s}}[n \mid n]^T\ ]$$
$$\mathbf{M}\ [n \mid n-1] = E[(\mathbf{s}[n] - \hat{\mathbf{s}}[n \mid n-1])(\mathbf{s}[n] - \hat{\mathbf{s}}[n \mid n-1])^T\ ]$$

3.2 Extended Kalman Filter

The Extended Kalman filter is a variation of Kalman filter employed when the system has non-linear equations resulting that instead of

$$\hat{\mathbf{s}}[n] = \mathbf{A}\hat{\mathbf{s}}[n\text{-}1] + \mathbf{Bu}[n]$$
$$\mathbf{x}[n] = \mathbf{H}[n]\mathbf{s}[n] + \mathbf{w}[n]$$

We would have

$$\hat{\mathbf{s}}[n] = \mathbf{a}(\hat{\mathbf{s}}[n-1]) + \mathbf{B}\mathbf{u}[n]$$
$$\mathbf{x}[n] = \mathbf{h}(\mathbf{s}[n]) + \mathbf{w}[n]$$

As a result we have to linearize **a** and **h**. We linearize **a**(**s**[n-1]) about the estimate of **s**[n-1] and **h**(**s**[n]) about **s**[n]. The resulting State and Measurement equations will be:

$$\mathbf{s}[n] = \mathbf{A}[n-1]\mathbf{s}[n-1] + \mathbf{B}\mathbf{u}[n] + \mathbf{a}(\hat{\mathbf{s}}[n-1\,|\,n-1]) - \mathbf{A}[n-1]\hat{\mathbf{s}}[n-1\,|\,n-1])$$

$$\mathbf{x}[n] = \mathbf{H}[n]\mathbf{s}[n] + \mathbf{w}[n] + (\mathbf{h}(\hat{\mathbf{s}}[n\,|\,n-1]) - \mathbf{H}[n]\hat{\mathbf{s}}[n\,|\,n-1]$$

Where **A**[n-1] and **H**[n] are defined using Jacobeans:

$$\mathbf{A}[n-1] = \left.\frac{\partial \mathbf{a}}{\partial \mathbf{s}[n-1]}\right|_{\mathbf{s}[n-1]=\hat{\mathbf{s}}[n-1|n-1]}$$
$$\mathbf{H}[n] = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{s}[n]}\right|_{\mathbf{s}[n]=\hat{\mathbf{s}}[n|n-1]}$$

The Recursive estimation Equations for the Extended Kalman Filter will now be:

Prediction:

$$\hat{\mathbf{s}}[n\,|\,n-1] = \mathbf{a}(\hat{\mathbf{s}}[n-1\,|\,n-1])$$

Minimum Prediction MSE Matrix $(p \times p)$**:**

$$\mathbf{M}[n\,|\,n-1] = \mathbf{A}[n-1]\mathbf{M}[n-1\,|\,n-1]\mathbf{A}^{T} + \mathbf{B}\mathbf{Q}\mathbf{B}^{T}$$

Kalman Gain Matrix $(p \times M)$**:**

$$\mathbf{K}[n] = \mathbf{M}[n\,|\,n-1]\mathbf{H}^{T}[n](\mathbf{C}[n] + \mathbf{H}[n]\mathbf{M}[n\,|\,n-1]\mathbf{H}^{T}[n])^{-1}$$

Correction:

$$\hat{\mathbf{s}}[n\,|\,n] = \hat{\mathbf{s}}[n\,|\,n-1] + \mathbf{K}[n](\mathbf{x}[n] - \mathbf{h}(\hat{\mathbf{s}}[n\,|\,n-1]))$$

Minimum MSE Matrix $(p \times p)$ **:**

$$\mathbf{M}[n \mid n] = (\mathbf{I} - \mathbf{K}[n]\mathbf{H}[n])\mathbf{M}(n \mid n-1]$$

Important thing to remember is that the Extended Kalman Filter has no optimality property and its result will depend on the accuracy of linearization which limits its usage as calculating Jacobeans in real-time is not easy and results are not accurate enough.

3.3 Kalman Filter based Processing of Raw INS Data

The choice between using Kalman or Extended Kalman Filter is based on the relationship between the State Vector and the Measurement Vector. Our State Vector comprises of True values of Position, Velocity, Acceleration and Bias in Acceleration in 2D XY Plane, whereas, the Measurement Vector comprises of Biased and Noise corrupted version of the X and Y component of true acceleration. As such the Measurement Acceleration is the sum of the true Acceleration and Bias.

Using the Newton's Equation of Motion, we developed a simple Dynamic Model for use as the State Transition Matrix to show the development and performance of Kalman Filter and its sensitivity to selection of proper noise Covariances.

The **State Model** developed is:

$$\mathbf{s}[n] = \mathbf{A}\mathbf{s}[n-1] + \mathbf{u}[n]$$

$$\begin{bmatrix} Pos_x[n] \\ Vel_x[n] \\ Acc_x[n] \\ Bias_x[n] \\ Pos_y[n] \\ Vel_y[n] \\ Acc_y[n] \\ Bias_y[n] \end{bmatrix} = \begin{bmatrix} 1 & T & T^2/2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T & T^2/2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & T & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Pos_x[n-1] \\ Vel_x[n-1] \\ Acc_x[n-1] \\ Bias_x[n-1] \\ Pos_y[n-1] \\ Vel_y[n-1] \\ Acc_y[n-1] \\ Bias_y[n-1] \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ u_{Acc_x} \\ u_{Bias_x} \\ 0 \\ 0 \\ u_{Acc_y} \\ u_{Bias_y} \end{bmatrix}$$

Where *Acc, Vel* and *Pos* are Acceleration, Velocity and Position respectively and subscripts denote their dimension. **u**[n] is the vector of 'Process Noise'. It is very important as this specifies that how much variations in a parameter of state vector can be expected and allowed. A small value of Process Noise will result in large variations to be suppressed and vice-versa. Here, the Process Noise is specified for the parameters of Acceleration and Bias only as the other two parameters i.e. Position and Velocity are dependent on Acceleration itself, so variance of Acceleration defines the variance of them as well. *T* is the sampling time period i.e. time between two consecutive measurements from the IMU. As *T* is constant, the model is Linear in its behavior.

The **Measurement Model** corresponding to the above State Model is:

$$\mathbf{x}[n] = \mathbf{H}[n]\mathbf{s}[n] + \mathbf{w}[n]$$

$$\begin{bmatrix} A_x[n] \\ A_y[n] \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} Pos_x[n] \\ Vel_x[n] \\ Acc_x[n] \\ Bias_x[n] \\ Pos_y[n] \\ Vel_y[n] \\ Acc_y[n] \\ Bias_y[n] \end{bmatrix} + \begin{bmatrix} \omega_{Ax} \\ \omega_{Ay} \end{bmatrix}$$

Here ***Ax*** and ***Ay*** represent the Biased and Noisy version of Acceleration in X and Y dimensions respectively. This means that measured acceleration is sum of true acceleration and bias plus some measurement noise. The vector **w**[n] represents measurement noise. It is the noise induced when reading the measurements from the IMU sensors. A small value of measurement noise indicates that the measurements are more reliable and are hence followed more closely by the Kalman Filter. A large measurement noise, on the other hand means that measurements are not so useful and Kalman Filter uses the State Model to predict and estimate the values of state vector.

As can be seen, both State and Measurement Models are Linear. Therefore, we do not need to employ Extended Kalman Filter and simple Kalman Filter will suffice for our analysis.

Now we process data recorded from Inertia Link IMU device over a period of approximately 1 hour in static condition and try to determine the value of Bias present in Acceleration measurements and predict the position. Ideally the position should (0,0) in Cartesian coordinates. It is interesting to see the difference between the resultant positions when using Raw and Processed data. The sampling rate here is T=0.1 seconds which gives us 36000 samples.

3.3.1 The Measurement Noise

The values for measurement noises in X and Y components of acceleration were determined using the recorded data in stationary condition. As in stationary condition, the mean of the samples represent the bias and the variance of the samples represents the variance of the remaining Zero-mean Gaussian Noise present in the system. Since we are estimating the bias in real-time by including it in the state vector, we only needed to specify measurement variances for accelerations which were determined to be:

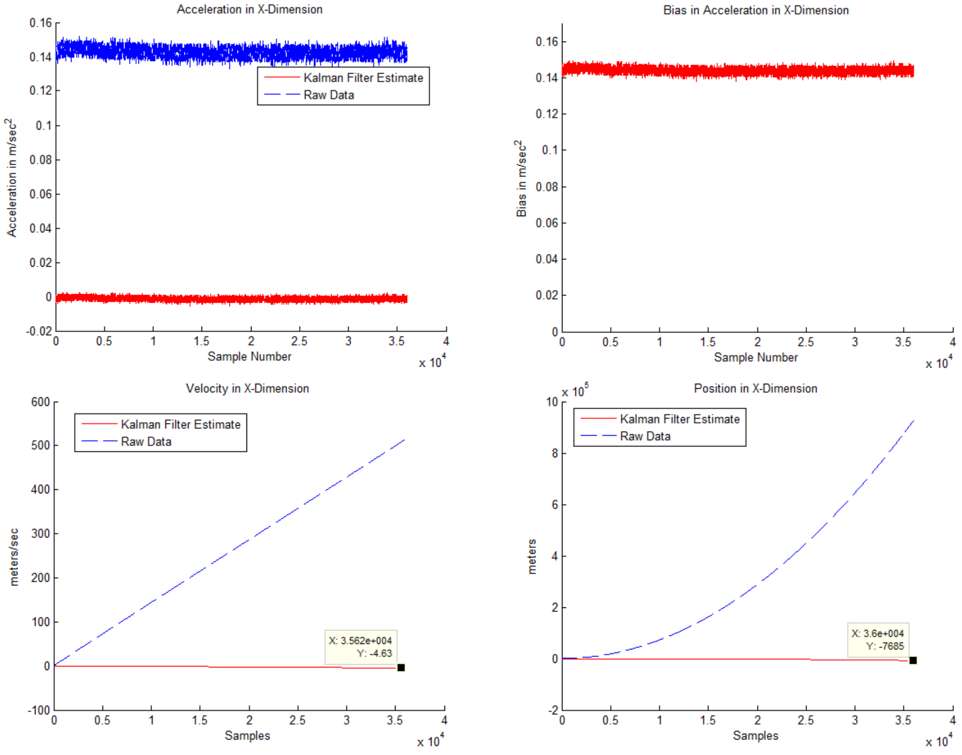
- Measurement Noise Variance in X Component of Acceleration = 5.2464e-006
- Measurement Noise Variance in Y Component of Acceleration = 5.4411e-006
- Sampling Period T = 0.1 seconds or 10Hz

3.3.2 Processing with High Process Noise Variance

Allowed Variance in Acceleration = 1e-3
Allowed Variance in Bias = 0.0015

When processing with a relatively high value of process noise, the small variations in the acceleration which are due to noise are not filtered out properly and integrated. This integration results in a linear drift for velocity estimate and exponential drift in position estimate as can be seen in Figure 3.1. The final estimate using Kalman Filter, although has smaller error then the error due to raw data processing, is still very large and unacceptable which can be seen Figure 3.5

Processing in X- Dimension:



Processing in Y- Dimension:

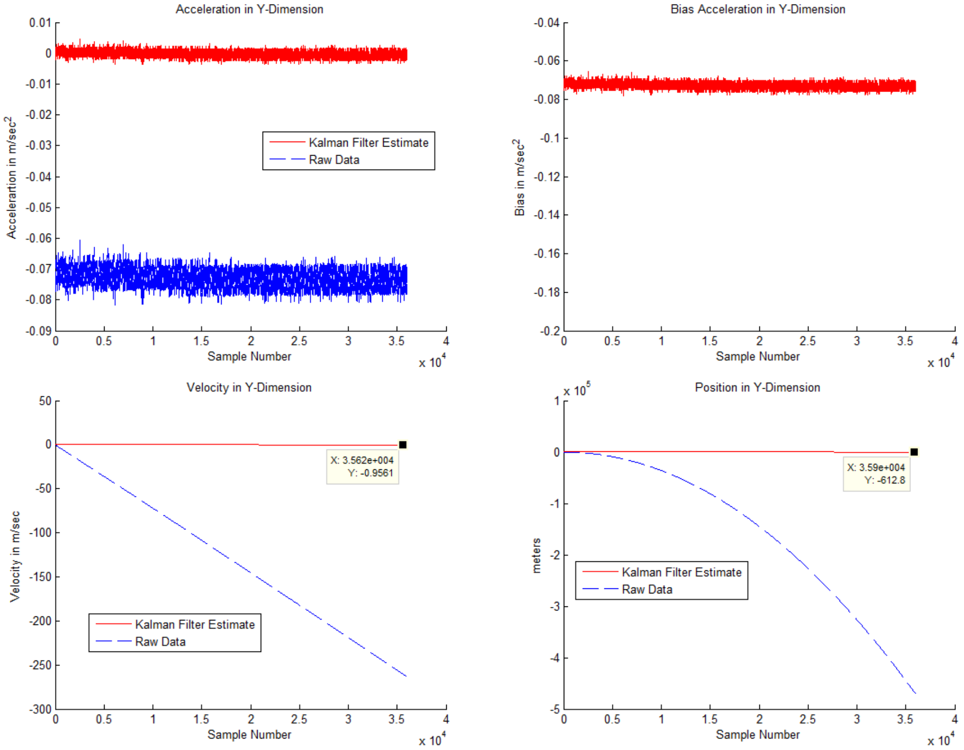


Figure 3.1: Processing INS data with variance equal to 1e-3

Position Estimate in XY Plane

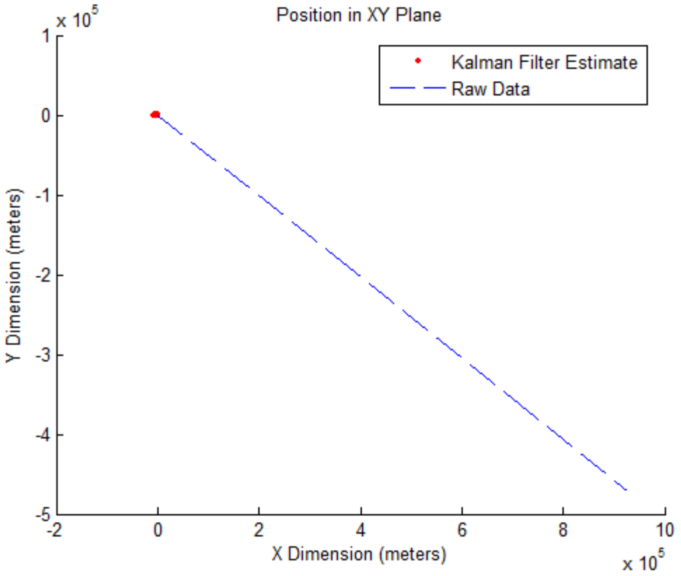


Figure 3.2: Position Estimates using Raw and Filtered Data (Variance=1e-3)

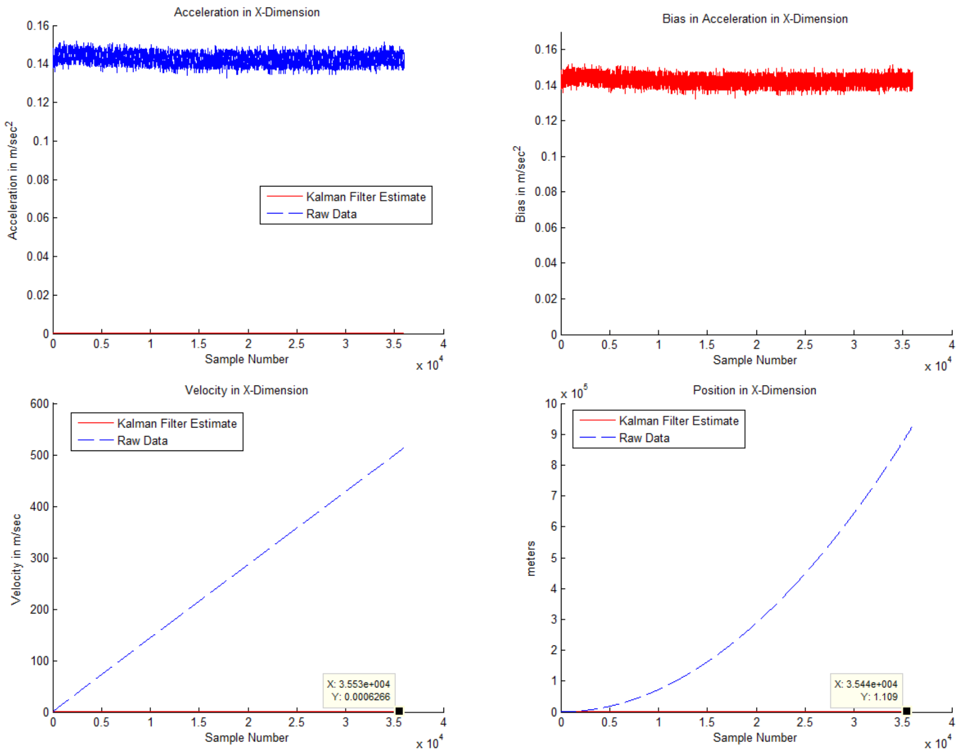
3.3.3 Processing with Low Process Noise Variance

Allowed Variance in Acceleration = 1e-9

Allowed Variance in Bias = 0.0015

Due to a very low process noise variance, the Kalman filter eliminates all the variations in the acceleration measurements. After the removal of bias and noise, the estimated acceleration is almost zero and so are the velocity and position estimates. This can be seen in Figure 3.3 and 3.4.

Processing in X- Dimension:



Processing in Y- Dimension:

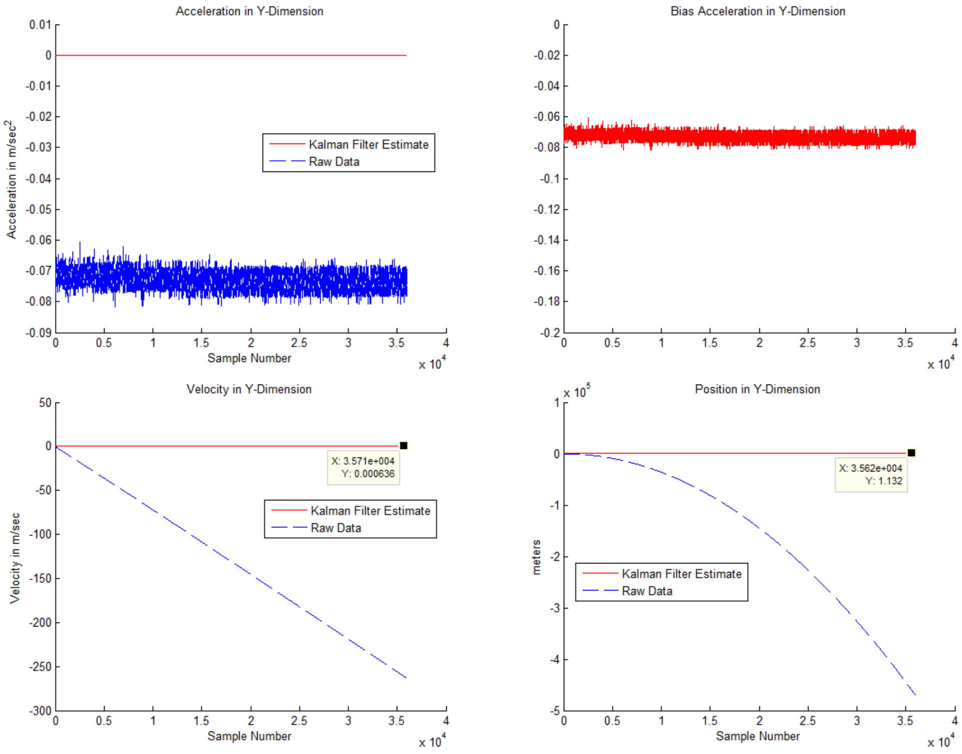


Figure 3.3: Processing INS data with variance equal to 1e-9

Position Estimate in XY Plane

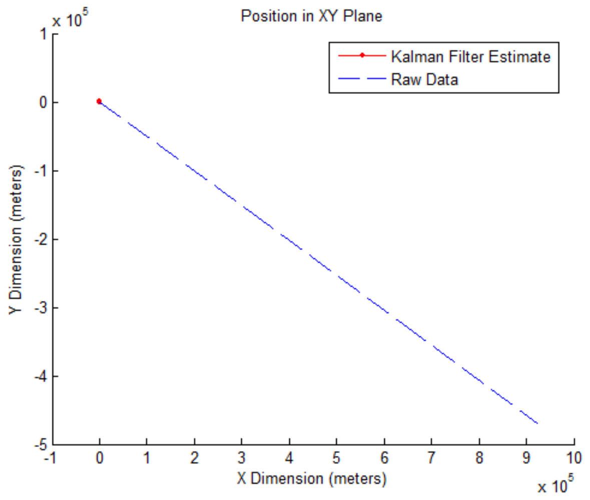


Figure 3.4: Position Estimates using Raw and Filtered Data (Variance=1e-9)

3.3.4 Comparison

From the above graphs, the benefit of using the Kalman Filter is clear as the difference between values of position using Raw IMU data and filtered data is extraordinarily tremendous. Still, to choose between Low and High Process covariance, the below graphs of Kalman Filter estimated positions are helpful.

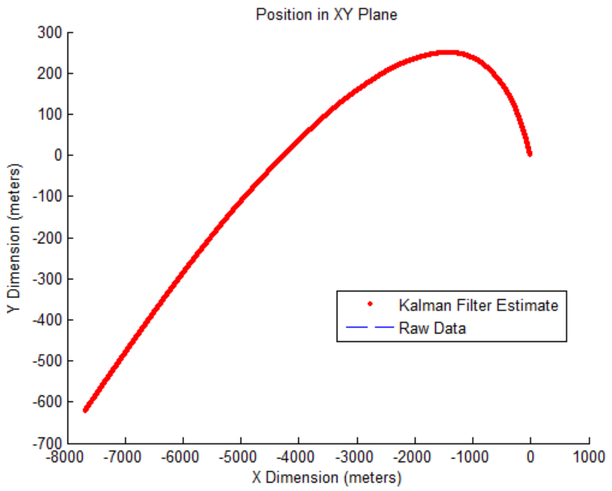


Figure 3.5 Position Estimate with 1e-3 variance

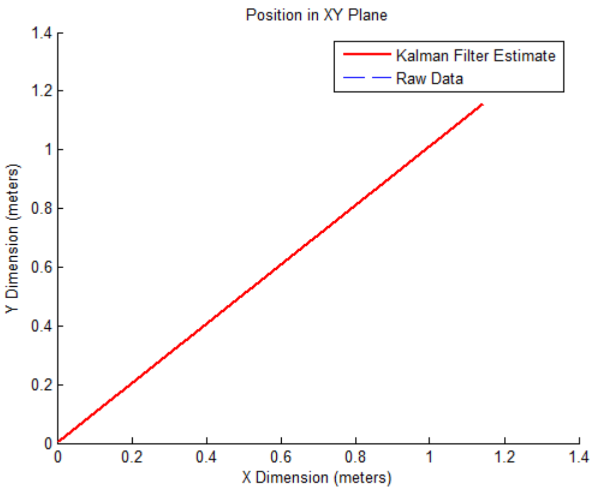


Figure 3.6: Position Estimate with 1e-9 variance

It can be seen that in case of High Process Variance, the resultant position is at ~(-7658,-603)meters which is far much away from it ideal (0,0) position than compared to the position calculated using Low Process Variance which is only (0.38,1.95)meters. This is effect of choosing the Process Noise Variance carefully and according to the specific scenario. Since, the data was associated to stationary state, a Low Process Variance is found to generate the corresponding correct results in this case. We will be using the model developed here with this low process variance in chapter 5 as one of the models in the IMM structure and refer to it as the 'Static Model'.

CHAPTER 4

VEHICLE DYNAMICS MODELS

The vehicle dynamic models describe different possible evolutions of a trajectory for a vehicle. Different types of vehicles like cars, ships, planes perform different types of maneuvers from simple to complex. As such a single model does not represent all such maneuvers and particular models are constructed to accurately describe and aid in predicting the trajectory according to the application. These Vehicle Dynamic Models, also known as Motion Models, are used in filtering techniques such Kalman Filter as State Transition Matrix. With the help of these models, Kalman filter is able to predict the next value in target tracking. The better are these models, the better is the performance of estimation filter. Therefore, a lot time was spent in studying and selecting models. One of the issues while designing motion models is that one should be able to relate the state vector to the measurement model. That is the parameters to be predicted must be related to the observations available. Many models in the literature were related to target tracking using range and bearing measurements form radars and as such we had to do slight modifications to relate them to inertial systems.

A target dynamic model or motion model describes the evolution of the target state with respect to time. A non-maneuvering model is the *straight* and *level* motion at a *constant velocity* sometimes referred to as *uniform motion.* All other types of motion belong to maneuvering class.

In the following sub-sections, we discuss some of the typical types of models. We discuss here the Maneuvering Dynamic Models coupled and un-coupled across different coordinates. A detail treatment can be found in [13] which has been the main reference is developing motion models in this thesis.

In all cases the 1 dimensional state vector, unless otherwise stated, is defined as:

$$x = [position, velocity, acceleration, bias]'$$

And the Measurement Model associated is same as defined in Section 3.3 of Chapter 3.

4.1 Coordinate Un-Coupled Models

Most target motion models are coupled across different dimensions i.e. the motion in one dimension effects on the motion in other dimension. But for most of the cases this coupling is assumed weak and the models are uncoupled which makes it easier to consider a particular direction at a time.

We discuss here the models that were selected to be used during the development of this thesis.

4.2 Wiener Process Acceleration Model

This model assumes acceleration to be a Wiener process or more specifically that it is a process with independent increments which is not necessarily a Wiener process. It is also referred to as the Constant Acceleration or nearly-constant Acceleration Model (CA).

This model has two variations. The first one is the **White Noise Jerk Model** which assumes that the Acceleration Derivative i.e. Jerk is an independent white noise process.

The Discrete-time Equivalent of this model in 1 Dimension with $x = [position, velocity, acceleration]'$ is:

$$x_{k+1} = F_1 x_k + w_{k,} \qquad F_1 = \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix}$$

$$Q = \mathrm{cov}(w_k) = S_w Q_3, \qquad Q_3 = \begin{bmatrix} T^5/20 & T^4/8 & T^3/6 \\ T^4/8 & T^3/3 & T^2/2 \\ T^3/6 & T^2/2 & T \end{bmatrix}$$

Where, $E[w(t+\tau)w(t)] = S_w \delta(\tau),$ and S_w is the power spectral density, not the variance, of the continuous-time white noise.

The second version called the **Wiener-Sequence Acceleration Model** assumes the Acceleration Increment as an independent white noise process. Acceleration Increment over a time is the integral of the jerk over that time. This model results in:

$$x_{k+1} = F_1 x_k + G_1 w_{k,} \qquad G_1 = \begin{bmatrix} T^2/2 \\ T \\ 1 \end{bmatrix}$$

The vector w_k is the discrete-time zero-mean Gaussian noise in acceleration and G_1 relates it to the state vector elements.

Note that its noise term has a covariance different from that of the White-Noise Jerk model:

$$Q_1 = \mathrm{cov}(G_1 w_k) = \mathrm{var}(w_k) \begin{bmatrix} T^3/4 & T^3/2 & T^2/2 \\ T^3/2 & T^2/2 & T \\ T^2/2 & T & 1 \end{bmatrix}$$

These models are crude as actual maneuvers are actually quite complex but they do provide modeling simple motions of a moving target.

In our Thesis we used the **Wiener-Sequence Acceleration Model** and modified it slightly to accommodate for the inclusion of Bias parameter in State Vector and Covariance Matrix.

The resultant one-dimensional State model and Covariance Matrix are

$$x_{k+1} = F_2 x_k + G_2 w_{k,} \qquad G_2 = \begin{bmatrix} T^2/2 \\ T \\ 1 \\ 0 \end{bmatrix}$$

Where

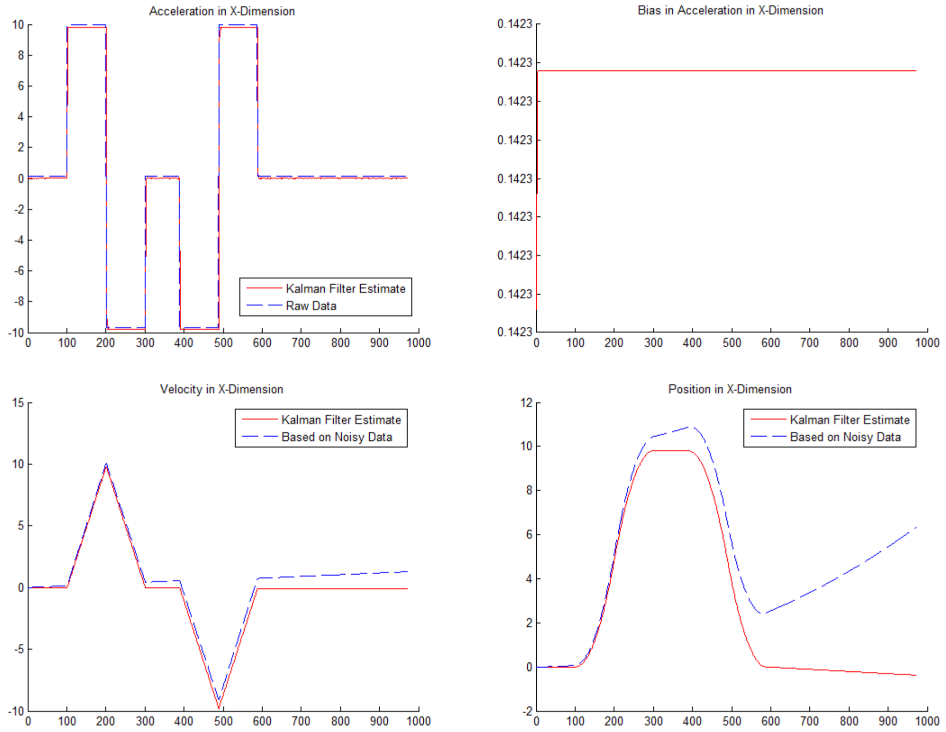
$$x = [Position, Velocity, Acceleration, Bias]$$

and

$$F_2 = diag(F_1, 1)$$
$$Q_2 = diag(Q_{1,} 1)$$

The value of variance for w_k was chosen to be 0.001 which was sufficient to provide correct results in the simulation. The Bias is a constant offset, which is assumed to be invariant and hence it was set to zero process noise. Some of the results of simulations with synthetic data induced with the same noise and offset bias corresponding to the Inertia Link IMU device are shown on next page in Figure 4.1. In all figures, Acceleration has units of meter/sec^2, velocity is expressed in meters/second and position is in meters.

X Dimension



Y Dimension

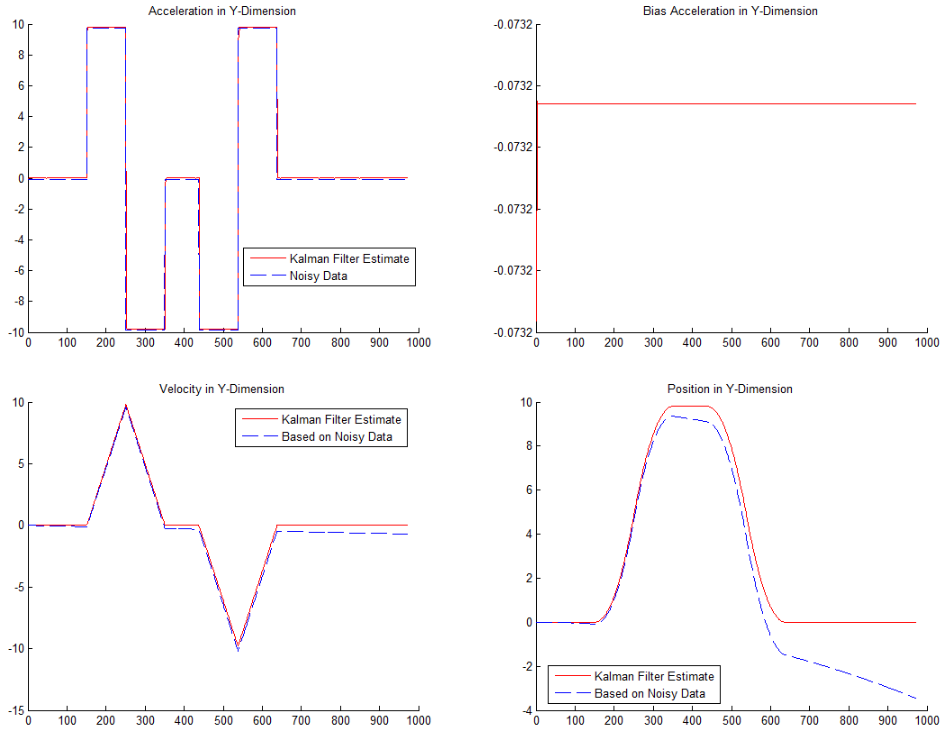


Figure 4.1: Synthetic Data of Acceleration and its Processing using Wiener Model

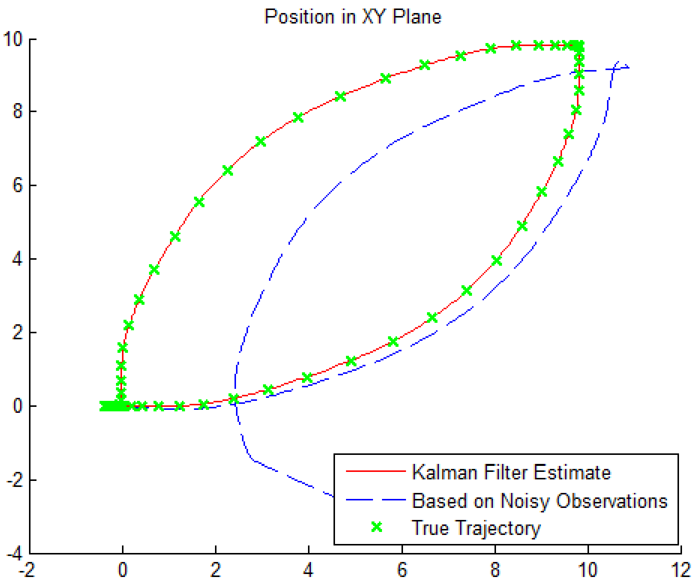


Figure 4.2: Comparison of trajectory estimates using Wiener Model and Noisy Data with the True trajectory

4.3 Singer Model

In stochastic modeling, a random variable is used to represent an unknown time-invariant quantity whereas an unknown time-varying quantity is modeled by a random process. White-noise constitutes the simplest class of random processes but where white-noise models are not suitable either the independent increment i.e. wiener models are used or the Markov process models which include Wiener and white-noise as special cases [13].

A white noise process is uncorrelated in time while in a Markov process a value at one time depends on its immediate neighbors. The Singer Model which has been described in detail in [14] and [15] assumes target acceleration as a Zero-mean Stationary First-order Markov Process where autocorrelation is given by:

$$r(\tau) = Ea(t)a(t+\tau) = \sigma_m^2 \, e^{-\alpha|\tau|}, \qquad \alpha \geq 0$$

Such a process $a(t)$ is the process of a linear time-invariant system

$$\dot{a}(t) = -\alpha a(t) + w(t)$$

Where $a(t)$ is a zero-mean white-noise process with constant power spectral density $S_w = 2\alpha\sigma^2$. The discrete time equivalent would be

$$a_{k+1} = \beta a_k + w_k^a$$

Where w_k^a is zero-mean white-noise sequence with variance $\sigma^2(1-\beta^2)$.
The state space representation is then given by

$$x_{k+1} = F_3 x_k + w_k = \begin{bmatrix} 1 & T & (\alpha T - 1 + e^{-\alpha T})/\alpha^2 \\ 0 & 1 & (1-e^{-\alpha T})/\alpha \\ 0 & 0 & e^{-\alpha T} \end{bmatrix} x_k + w_k$$

α and σ^2 are design parameters and the performance of Singer model depends on the accuracy in determining them. The α is defined as $1/\tau_m$ i.e. the reciprocal of the maneuver time constant and depends on how long the maneuver lasts. For example, for an aircraft’s lazy turn it can be 60 seconds and for an evasive maneuver 10-20 seconds. σ^2 is the instantaneous variance of acceleration treated as a random variable.

A distribution was suggested by Singer in [14] using *ternary-uniform mixture* shown in Figure 4.3. A target may move without acceleration with probability P_0, accelerate or decelerate with a maximum probability of $P_{\max}$ or accelerate or decelerate at a rate uniformly distribute over $(-A_{\max,} A_{\max})$. Given this, variance then results in

$$\sigma_m^2 = \frac{A^2{}_{\max}}{3}[1 + 4P_{\max} - P_0]$$

Where $P_0, P_{\max}$ and $A_{\max}$ are design parameters.

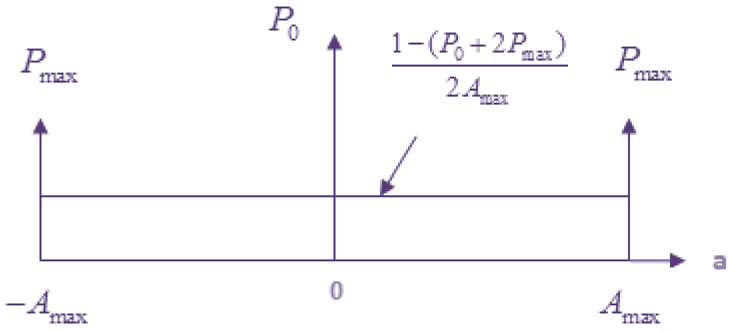


Figure 4.3: Ternary-Uniform Mixture Distribution as Suggested by Singer [14]

The covariance matrix for Singer Model for a fixed sensor and sampling period T is given by:

$$\lim_{\alpha T \to 0} Q_3(k) = 2\alpha\sigma_m^2 \begin{bmatrix} T^5/20 & T^4/8 & T^3/6 \\ T^4/8 & T^3/3 & T^2/2 \\ T^3/6 & T^2/2 & T \end{bmatrix}$$

Again after accommodating for bias, the 1-dimensional State Vector and corresponding Covariance Matrix becomes:

$$x_{k+1} = F_4 x_k + w_{k,}$$

Where

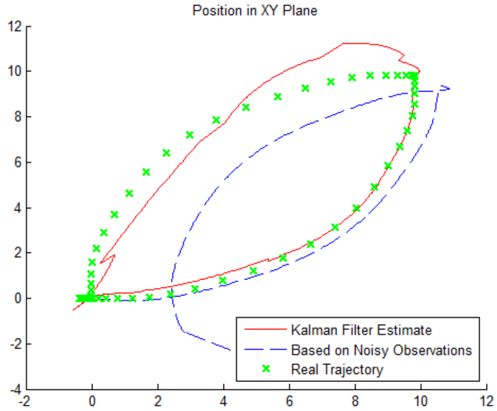
$$x = [Position, Velocity, Acceleration, Bias]$$

$$F_4 = diag(F_3, 1)$$
$$Q_4 = diag(Q_{3,} 1)$$

The choice of α and *T* determine the behavior of Singer Model. As τ_m increases, and α decreases, Singer model corresponds more to a Constant Acceleration (CA) model. On the other hand as τ_m decreases the Singer Model behaves more like Constant Velocity (CV) model where acceleration is considered noise. Consequently, the value of α determines its behavior between CA and CV models.

Figure 4.4 shows some results which were obtained during simulations for finding the correct value of τ_m for our analysis. They show the effect of high and low values of τ_m with respect to the maneuvers. The simulations were done with a Sampling rate T=0.01 seconds and maneuver lasting for 1 second. In all figures, Position is expressed in meters.

For τ_m=0.2



For τ_m=0.5

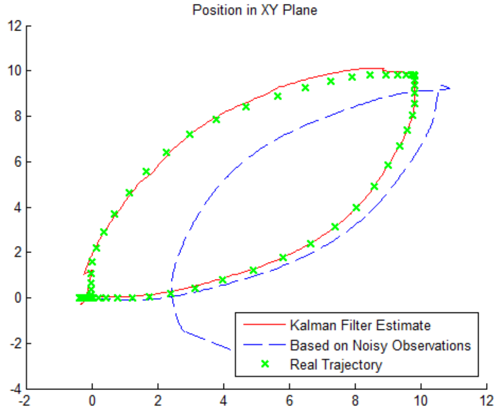
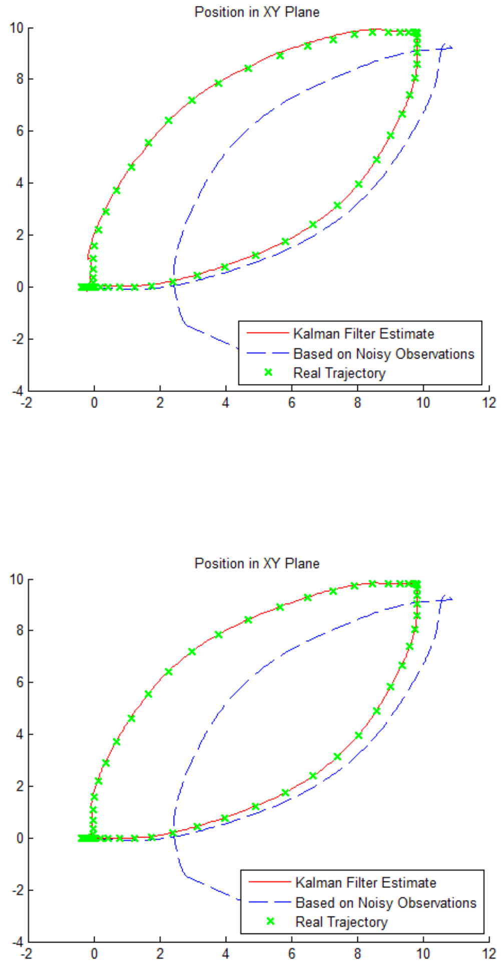


Figure 4.4a: Effect of maneuvering time-constant on tracking performance (a)



For τ_m =1

For τ_m = 3

Figure 4.4b: Effect of maneuvering time-constant on tracking performance

The Singer acceleration model is a standard model for target maneuvers. It was the first model that characterizes the unknown target acceleration as a time-correlated (i.e., colored) stochastic process, and has served as a basis for the further development of effective target maneuver models. This is the advantage of Singer model over other models as acceleration in many cases is time-correlated and Singer model is able to use this correlation to better predict the states. Another advantage is that in Singer model the choice of process variance is not a trial-and-test procedure and by knowing the correct maneuvering time constant, the right value is calculated using the ternary-uniform mixture distribution as shown in Figure 4.3.

A complete analysis and derivation can be found in the original paper written by Singer himself [14].

4.4 Coordinated Turn Model

Two dimensional models are naturally turn motion models and are in particularly called Coordinated Turn models. Coordinated Turn implies constant forward speed and constant turn rate. They are established based on target kinematics rather than random processes. The underlying equations for a 2D horizontal plane CT motion are derived from the standard curvilinear-motion model from kinematics:

$$\begin{aligned}
\dot{x}(t) &= V(t)\cos\varphi(t) \\
\dot{y}(t) &= V(t)\sin\varphi(t) \\
\dot{V}(t) &= a_t(t) \\
\dot{\varphi}(t) &= \frac{a_n(t)}{V(t)}
\end{aligned}$$

Where (x,y) are the Cartesian coordinates, V is the forward speed, and φ is the heading angle.

a_t and a_n denote target acceleration in tangential (along-track) and normal(cross-track) directions in the horizontal plane respectively. [13]

There are 3 special cases regarding a_t and a_n :

1. $a_n = 0$, $a_t = 0$ — rectilinear, constant velocity motion;
2. $a_n = 0$, $a_t \neq 0$ — rectilinear, accelerated motion (CA motion if a_t = constant);
3. $a_n \neq 0$, $a_t = 0$ — circular, constant speed motion (CT motion if a_n = constant).

The third case is known as a (standard) coordinated turn (CT), which has a constant forward speed and constant turn rate.

CT model has two variations, the case where the Turn rate ω is known and the case where it is unknown. Since the IMU device we used provides us with the Turn rate information, we used the CT Model with known Turn rate in our analysis.

4.4.1 CT Model with Known Turn Rate

The discrete time CT model with the state vector defined as $x = [x, \dot{x}, y, \dot{y}]^{'}$ is given by:

$$x_{k+1} = F_5(\omega)x_k + w_k = \begin{bmatrix} 1 & \sin \omega T & 0 & -(1-\cos \omega T)/\omega \\ 0 & \cos \omega T & 0 & -\sin \omega T \\ 0 & (1-\cos \omega T)/\omega & 1 & (\sin \omega T)/\omega \\ 0 & \sin \omega T & 0 & \cos \omega T \end{bmatrix} x_k + w_k$$

Since $\omega$is known, the model is linear. An approximation of the above model is given by:

$$F_6(\omega)x_k \approx \begin{bmatrix} 1 & T & 0 & -\omega T^2/2 \\ 0 & 1-(\omega T)^2/2 & 0 & -\omega T \\ 0 & \omega T^2/2 & 1 & T \\ 0 & \omega T & 0 & 1-(\omega T)^2/2 \end{bmatrix} x_k$$

Which is a 2nd order polynomial in ω. This approximation is simple but provides less accurate results and is valid only when $\omega T \approx 0$. In the cases where the Turn rate is known, like in our case, The CT model provides good tracking performance.

To include acceleration and bias in the state vector, the modified state model becomes,

$$x_{k+1} = F_7(\omega)x_k + w_k = \begin{bmatrix} 1 & \sin\omega T & 0 & 0 & 0 & -(1-\cos\omega T)/\omega & 0 & 0 \\ 0 & \cos\omega T & 0 & 0 & 0 & -\sin\omega T & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & (1-\cos\omega T)/\omega & 0 & 0 & 1 & (\sin\omega T)/\omega & 0 & 0 \\ 0 & \sin\omega T & 0 & 0 & 0 & \cos\omega T & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} x_k + w_k$$

Where

$$x = [Position_x, Velocity_x, Acceleration_x, Bias_x, Position_y, Velocity_y, Acceleration_y, Bias_y]'$$

$$w_k = [w_{pos_x}, w_{vel_x}, w_{Acc_x}, w_{Bias_x}, w_{pos_y}, w_{vel_y}, w_{Acc_y}, w_{Bias_y}]$$

The Coordinated Turn model is critical is detecting coordinate turn motions as simple un-coupled models such as Wiener and Singer model discussed above are unable to detect the coupling between the X and Y dimensions i.e. the dependence of position and velocity in one dimension on the position and velocity in the other direction. Consider the example in Figure 4.5 which shows the data of coordinated turn trajectory as processed by a simple uncoupled model and using the above coordinated turn model

Sampling Period T = 0.01 second, Angular Rate ω = 0.3rad/sec

In all figures, Acceleration has units of meter/sec^2, velocity is expressed in meters/second and position is in meters.

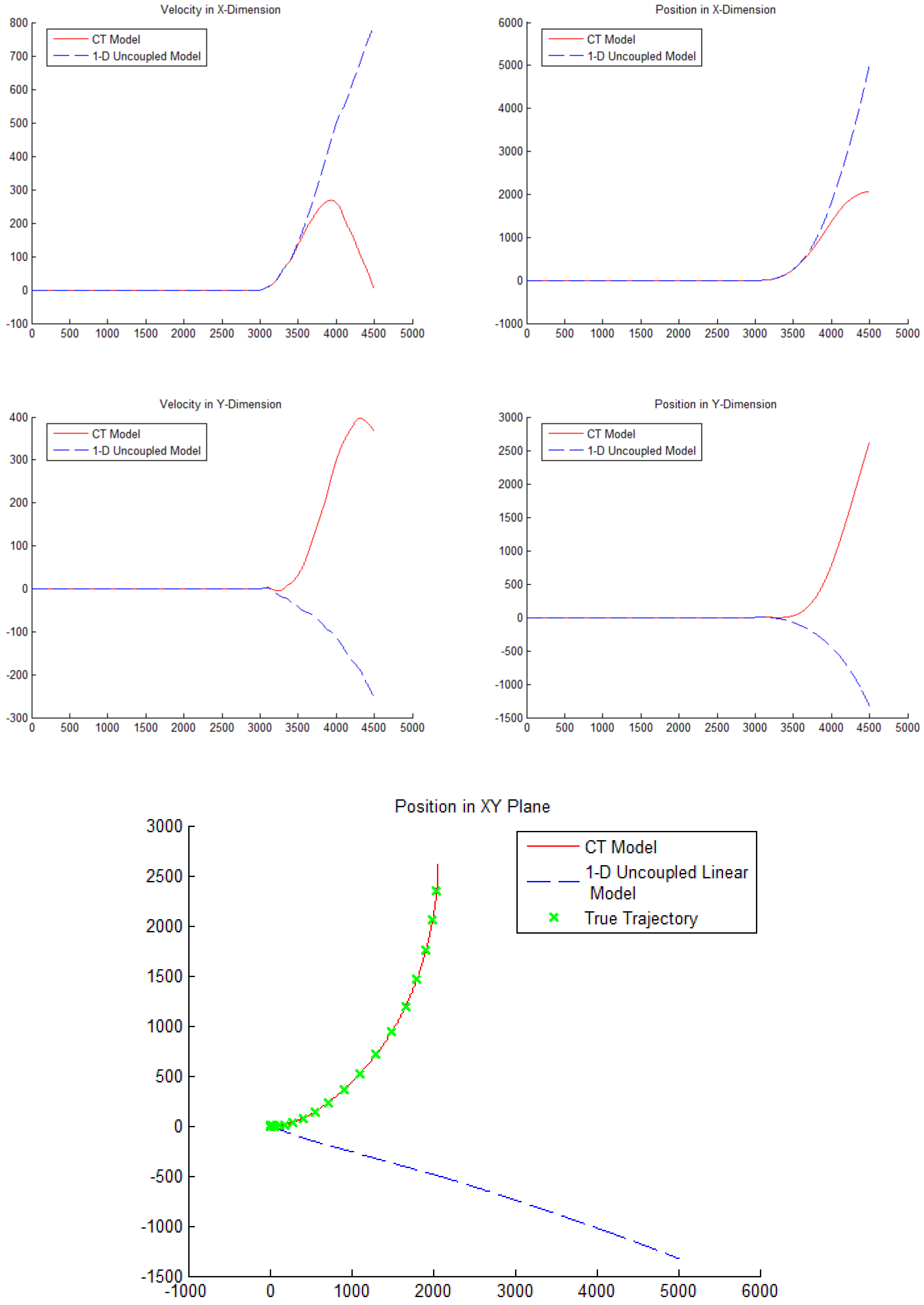


Figure 4.5: Difference in processing measurements by CT Model and Wiener Model

In the figure above, the uncoupled model fails to follow the correct trajectory and generates a linear motion. Whereas, the Coordinated Turn model uses the coupling between the two dimensions and the extra information provided by the turn rate to produce the correct angular trajectory.

Remarks:

We have discussed few of the most simple and commonly used models in maneuvering target tracking and analyzed the effect of their important parameters on their performance. We also simulated the performances of these models with synthetic data that was induced by an offset bias and Gaussian noise with variance equal to the noise in the IMU device we took as a reference case. This enabled us to see the expected theoretical performance of an INS system with these models. But, as said earlier, no vehicle's trajectory can be modeled by just any one of the above models and a maneuvering target has to be tracked with simultaneous models working together and detecting which type of maneuver is presently active and estimate its position according to the corresponding model. In the next chapter we will study and develop the technique that combines these models in such a way that these models work in parallel and estimates of the most relevant model are used to provide final navigation solution.

CHAPTER 5

INTERACTING MULTIPLE MODELS

In many target tracking applications the estimation of position, velocity and acceleration is difficult to achieve accurately using just a single model and/or sensor. Different types of vehicles perform different types of maneuvers at different stages and as such multiple models are required to provide estimation for different stages. Different type of Multi Model techniques have been evolved each with its own performance and complexity. The Interacting Multiple Model scheme is a sub-optimal filter which has been shown to achieve excellent compromise between performance and complexity. Compared to Decision based models, where only one model is being used for estimation, Interacting Multiple Model technique uses parallel filters estimating based on different models and then the final estimate is generated on a weighted sum basis of all individual estimates. The advantage here is that in decision-made techniques, if the hypothesis is incorrectly validated as true, all estimates will be based on a model that does not describe the current mode of maneuver, whereas, in IMM technique since all models are working in parallel and the final result is a weighted combination, the results are much more reliable.

The complexity of IMM is nearly linear [17] and another advantage is its modularity which means IMM can be set up using different building blocks, isolated from each other. So there can be different types of filters like Kalman Filter or Extended Kalman

Filter to account for the non-linearities in the system and models based on different sets of measurement sensors.

5.1 Baseline IMM Algorithm:

The simplest form of IMM Algorithm, which is a 3 step process, is presented below with the description of the nomenclature:

The quantities pertinent to filter *j* are denoted with subscript *j*

M_f is the Model Space representing the set of all models i.e. filters used in the system.

Z^k Denotes the measurement sequence through time *k*

$\hat{x}(i\,|\,l)$ Denotes the State estimate at time *i* conditioned on Z^l and $P(i\,|\,l)$ is the associated Covariance matrix;

$N(y;\bar{y},P)$ Denotes the (multivariate) Gaussian density function of *y* with mean $\bar{y}$ and covariance *P*

$\sum_i$ Denotes $\sum_{m_i \in M_f}$

$\hat{x}_j(k\,|\,k), P_j(k\,|\,k)$ Estimate and its Covariance in Mode-Matched filter *j* at time *k*

$\hat{x}_{0j}(k\,|\,k), P_{0j}(k\,|\,k)$ The mixed Initial Condition for Mode-Matched filter *j* at time *k*

$\hat{x}(k\,|\,k), P(k\,|\,k)$ The final combined State Estimate and its Covariance;

$\mu_j(k)$ The Mode Probability at time *k*

$\mu_{i|j}(k\,|\,k)$ Mixing Probability at time *k* (the weights with which the estimates from the previous cycle are given to each filter at the beginning of the current cycle)

$\Lambda_j(k)$ The likelihood function of Mode-Matched filter *j*

STEP 1: Interaction

Interaction is the phase where the parameters for initializing the filters; for each iteration of the IMM algorithm; are generated. It is based on weighted sum of the estimates from all the filters during the previous iteration. The weights are based on the 'Mixing Probability'. The 'Mixing Probability' is calculated at the beginning of each cycle using:

$$\forall i, j \in M_f, \mu_{i|j}(k-1|k-1) = 1/\overline{c}_j p_{ij} \mu_i(k-1)$$

Where

$$\overline{c}_j = \sum_i p_{ij} \mu_i(k-1)$$

Then the state vector and covariance matrix is calculated to provide initial values to every filter.

$$\hat{x}_{0j}(k-1|k-1) = \sum_i \hat{x}_i(k-1|k-1)\mu_{i|j}(k-1|k-1)$$

$$\begin{aligned} P_{0j}(k-1|k-1) = \sum_i \{ & P_i(k-1|k-1) \\ & + [\hat{x}_i(k-1|k-1) - \hat{x}_{0j}(k-1|k-1)] \\ & \times [\hat{x}_i(k-1|k-1) - \hat{x}_{0j}(k-1|k-1)]^T \} \\ & \times \mu_{i|j}(k-1|k-1) \end{aligned}$$

The estimates of state vector and uncertainty matrix are then forwarded to the individual filters.

STEP 2: Filtering

When the filters are initialized with the new estimates from previous cycles, they use them to provide new estimates based on their own state equations and motion models. This is a big advantage of IMM as all filters are independent of their structure and models. As such different models can work in parallel to provide the best possible estimates according to their state evolution models and the model which predicts the measurements closest to incoming true measurements is given more weight. After computing their new estimates and uncertainty, these values are returned to main IMM algorithm for final estimate using combination.

$$\forall j \in M_f$$

The Initial Prediction:

$$\hat{x}_j(k|k-1) = F_j(k-1)\hat{x}_{0j}(k-1|k-1) + \Gamma_j(k-1)\overline{v}_j(k-1)$$

$$P_j = (k|k-1) = F_j(k-1)P_{0j}(k-1|k-1)F_j(k-1)^T$$
$$+ \Gamma_j(k-1)Q_j(k-1)\Gamma(k-1)^T$$

Residual Covariance:

$$S_j(k) = H_j(k)P_j(k|k-1)H_j(k)^T + R_j(k)$$

Filter Gain:

$$W_j(k) = P_j(k|k-1)H_j(k)^T S_j(k)^{-1}$$

Measurement Prediction:

$$\hat{z}_j(k|k-1) = H_j\hat{x}_j(k|k-1)$$

Residual Error:

$$r_j(k) = z(k) - z_j(k|k-1)$$

Correction and Final Estimate with Covariance Update:

$$\hat{x}_j(k|k) = \hat{x}_j((k|k-1) + W_j(k)r_j(k)$$

$$P_j(k|k) = P_j(k|k-1) - W_j(k)S_j(k)W_j(k)^T$$

After estimating the new state vector values and the associated uncertainty, the individual filters calculate the ‘Mode Probability’ which is based on the on the residual error i.e. the difference between predicted measurements and true measurements defines the probability of a particular model to be active at time *k*. The less is the residual error of a filter, the more the probability that that filter and model best describe the current maneuver. Mathematically, this is done using the likelihood function:

$$\Lambda_j(k) = N(r_j(k); 0; S_j(k))$$

And **Mode Probability** is given by:

$$\mu_j(k)=\frac{1}{c}\Lambda_j(k)\sum_i p_{ij}\mu_i(k-1)=\frac{1}{c}\Lambda_j(k)\overline{c}_j$$

Where ***c*** is a normalizing factor:

$$c=\sum_j \overline{c}_j\Lambda_j(k)$$

STEP 3: Combination

This is the process where the new individual estimates are combined again using weights based on 'Mode Probability' at that time instant. This is a simple step but increases the accuracy of the final estimates.

$\forall j \in M_f$

$$\hat{x}(k\mid k)=\sum_j \hat{x}_j(k\mid k)\mu_j(k)$$

$$P(k\mid k)=\sum_j\{P_j(k\mid k)+[\hat{x}_j(k\mid k)-\hat{x}_j(k\mid k)\times[\hat{x}_j(k\mid k)-\hat{x}_j(k\mid k)]^T\}\mu_j(k)$$

We applied the IMM Algorithm to our problem of estimating position using Inertial Measurement System. We did simulations by using synthetic data for a trajectory with stationary, linear uncoupled motion and turn motion and we integrated the corresponding models in the IMM to see if IMM is able to identify the different stages of motion and estimate the position correctly. In all the simulation, the characteristic of the IMU device we have taken as a case were fed into the simulation. The Results for IMM system utilizing 2 and 3 parallel filters are shown in the next sections.

5.2 Interacting Multiple Model Scheme with 2 Parallel Filters:

In order to provide good tracking performance using the IMU device we require at least two models to work in parallel. As shown in Chapter 2 Section 2.6, even when the device is stationary, the constant bias and noise causes false acceleration measurements which cause drift in position estimate over time. So one of the models we are going to use in IMM scheme with 2 filters will be the one we developed in Chapter 3 Section 3.3.3 to eliminate this noise and bias in stationary mode, which we referred to as 'Static Model'. The other model is required to detect motion and provide the information about position as the trajectory evolves. We will be using the Singer model developed in Chapter 3 with maneuvering time-constant τ_m = 3 seconds.

In the below experimental results, we are using synthetic data, again induced with bias offset and Gaussian noise, which describes a trajectory with first 3000 samples of stationary phase and then acceleration along the X and Y dimensions. It is interesting to see how IMM, as it detects change in acceleration, shifts weights from one model to another based on which model best describes the current state. The sampling time here is T=0.01 seconds. In all figures, Acceleration has units of meter/sec^2, velocity is expressed in meters/second and position is in meters.

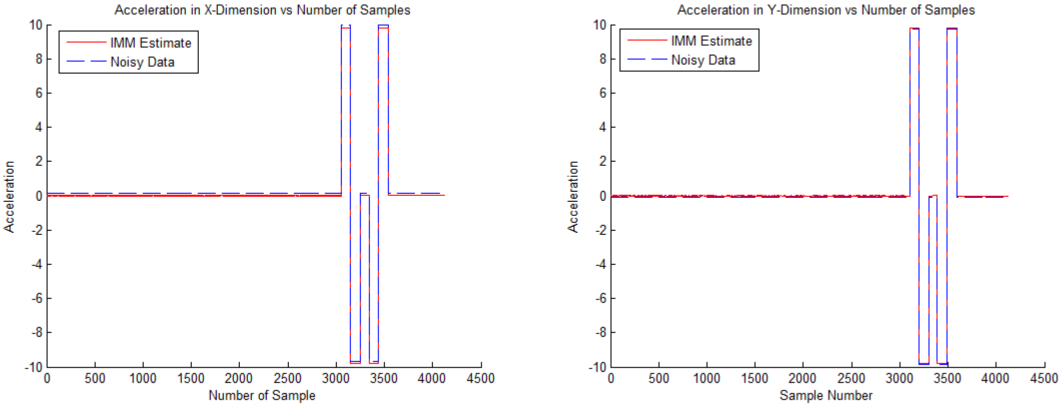


Figure 5.1: Simulated Measurements of Acceleration in meter/sec^2 in X and Y Dimensions

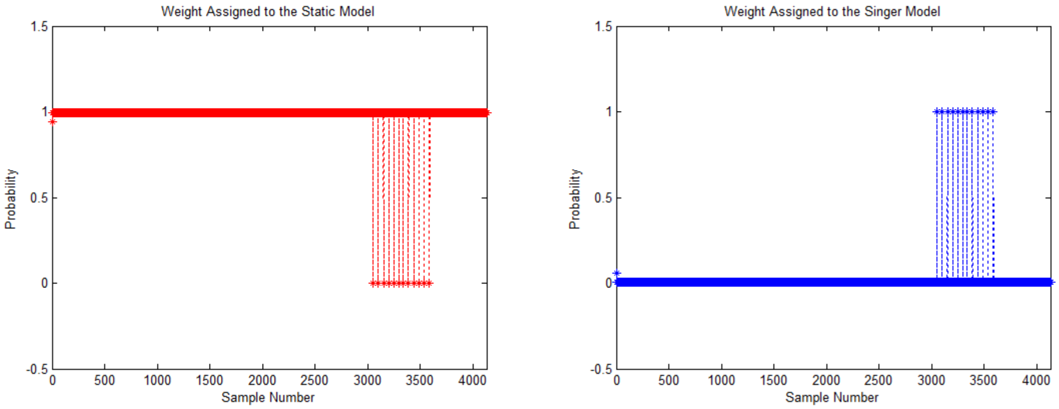


Figure 5.2: Shifting of weights between Model 1 and Model 2

As the state changes from stationary to accelerated state, the values predicted by Static Model are no longer close to the new measurements and hence, the IMM shifts its weights to the Singer Model which provides better and accurate measurement predictions.

Consider the figures 5.3 and 5.4, although there was small acceleration due to noise up to sample number 3000, the Static Model eliminated it and the difference between the position obtained by processing raw measurements and filtered measurements is clear. See how the positions in both X and Y dimensions equal to 0 up to the 3000 sample in the IMM case.

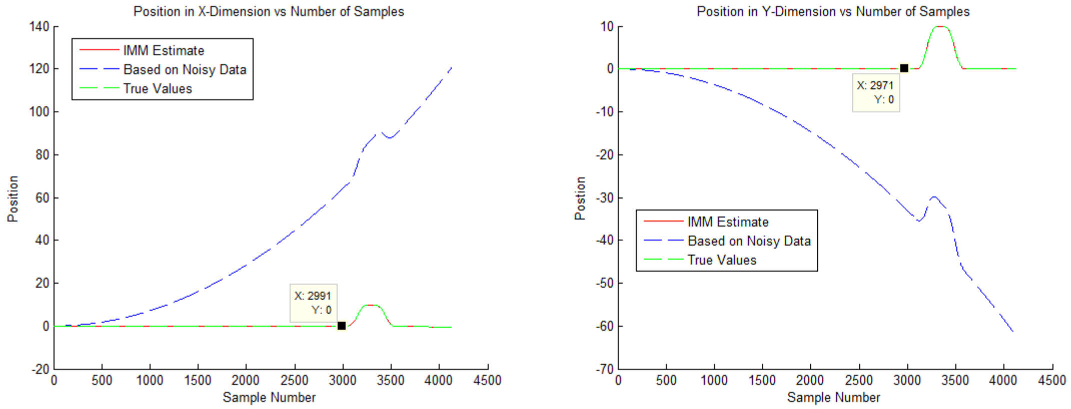


Figure 5.3: Comparison of the Position Estimates in X and Y Dimensions (unit: meters)

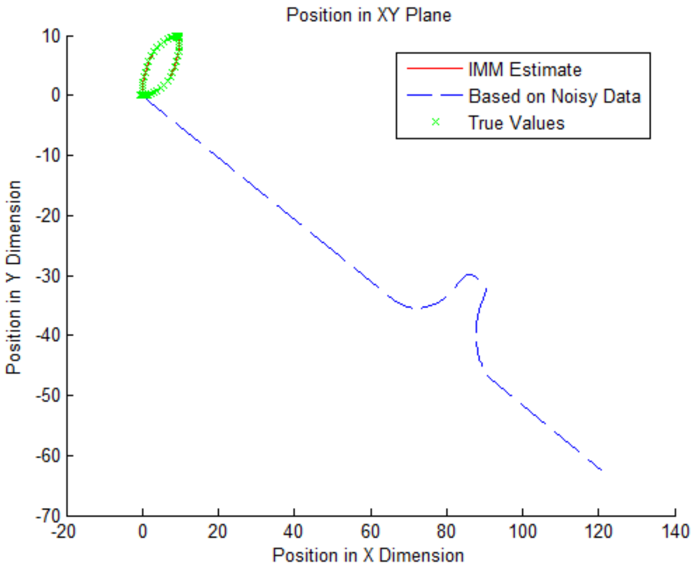


Figure 5.4: Comparison of the Trajectory based on Position estimates in XY Plane. The IMM Estimates are overlapped by the True values (unit: meters)

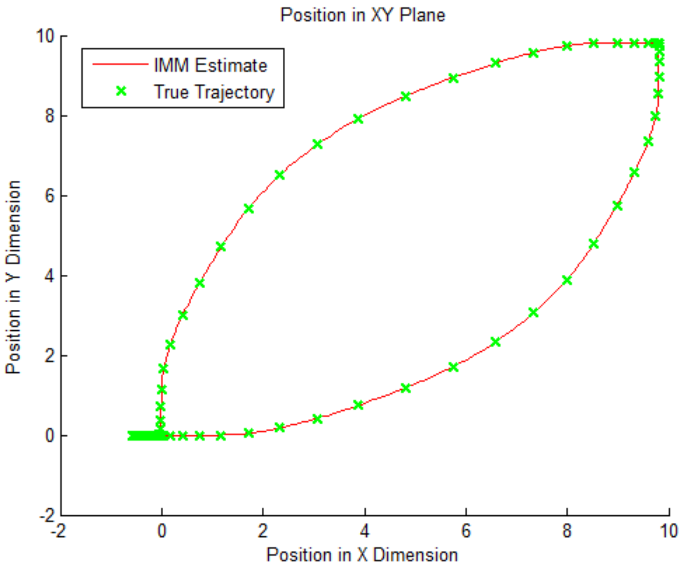


Figure 5.5: A Zoom-In of the Fig. 5.4 showing IMM Estimate compared to True values of Position (unit: meters)

This was a rather simple trajectory to test IMM performance. In section 5.3 we will test the IMM scheme with a more complex trajectory. But for now, it is interesting to see the benefits of IMM over Single Model based filters.

5.2.1 Performance comparison with Single Model based Filters

Above graphs show the performance of IMM filter in choosing the right model to estimate the positions. Below is a comparison of IMM filter with Single Model based filters which shows the deterioration of position estimates because of using only one model to describe the entire trajectory.

We simulated filters with Static and Wiener models using the same trajectory we used above with the IMM and compared the results with the true values. The results are shown below:

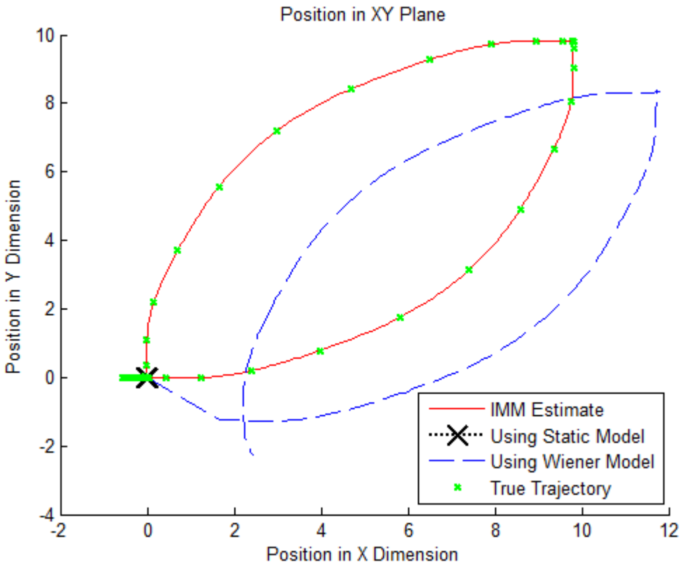


Figure 5.6: Comparison of the Position Estimates using IMM Filter and Single Model Filters (unit: meters)

In the above graphs the resultant position after the entire set of measurements with Static Model in still near the origin and the error is almost equal to the true values meaning all measurement have been ignored. This is because Static model has low process variance so it suppressed the measurements of acceleration even though they were not noise. In contrast, Wiener is able to predict the motion in the XY plane but it could not eliminate the noise in the measurements as effectively and hence has offset from true position values. In comparison, IMM filter provides the best of both worlds and its results overlap the true trajectory.

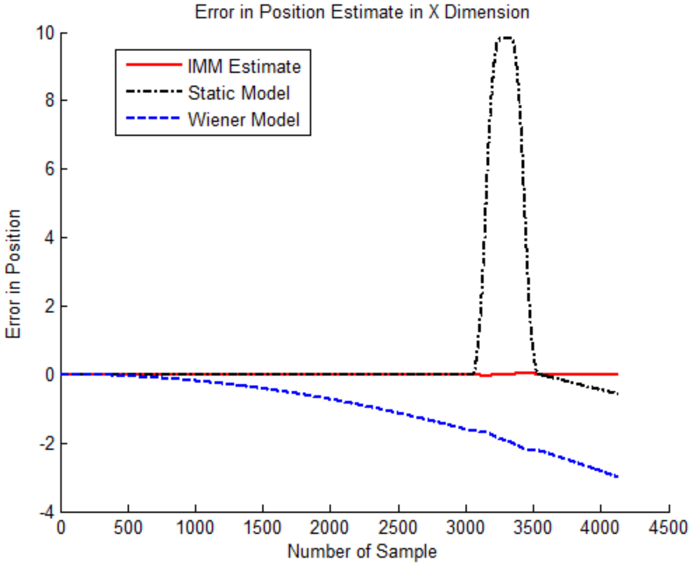


Figure 5.7: Error in Position Estimates in meters in X Dimension

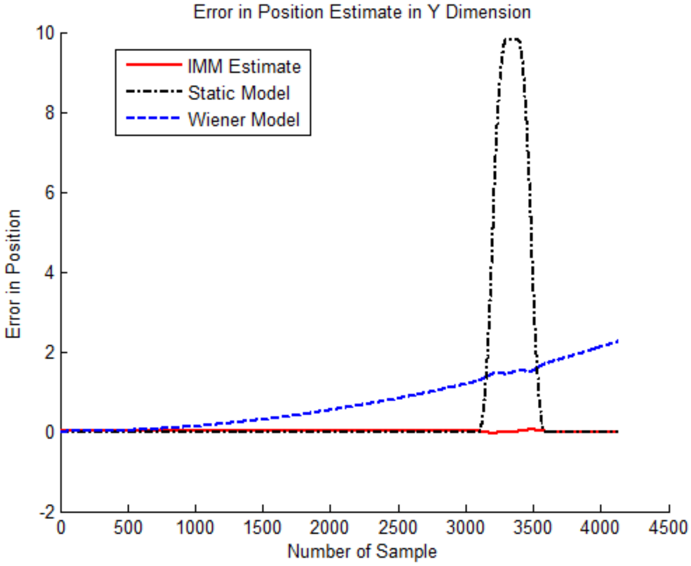


Figure 5.8: Error in Position Estimates in meters in Y Dimension

5.3 Interacting Multiple Model Scheme with 3 Parallel Filters:

In this section we test the performance of the IMM techniques utilizing 3 Parallel filters. Two of the filters are as same as in the previous section i.e. filters using Static and Singer Models. The third filter is based on the Coordinated Turn Model. The performance is tested on synthetic data describing a trajectory as in the previous section but with the addition of a coordinate turn motion in the end.

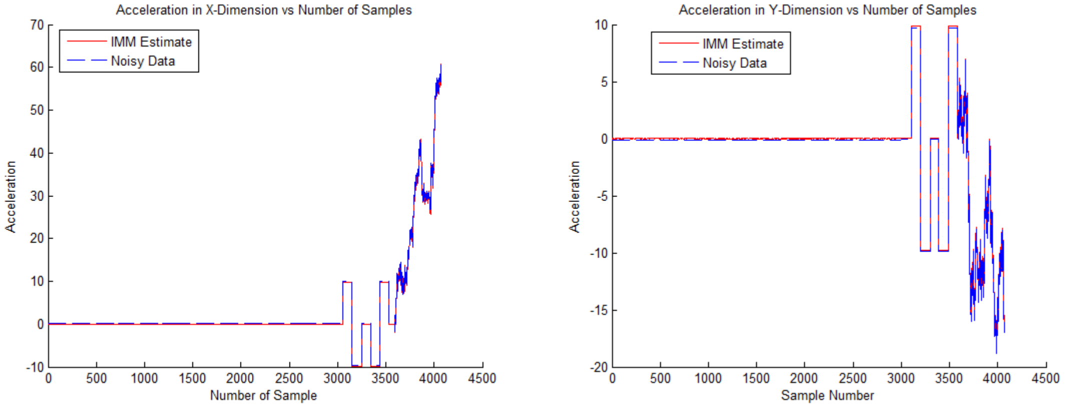


Figure 5.9: Measurements of Acceleration in meter/sec^2 in X and Y Dimensions

Up to sample number 3000, the state is stationary. From sample number 3000 to around 3600 there is motion in XY plane but this motion is uncoupled in dimensions. From 3600 onwards the acceleration defines a coordinated turn motion in which the X and Y coordinates are coupled to each other and the Angular Rate used in the simulation is 0.3 rads/sec.

By selecting proper process variances, we were able to make IMM identify the difference between linear motion and coordinated turn motion. Although, the probability is assigned to both Singer and CT Model during the coordinated turn phase of the trajectory, more weight is given to the CT model which gives acceptable performance results. See the difference in Figures 5.12-16.

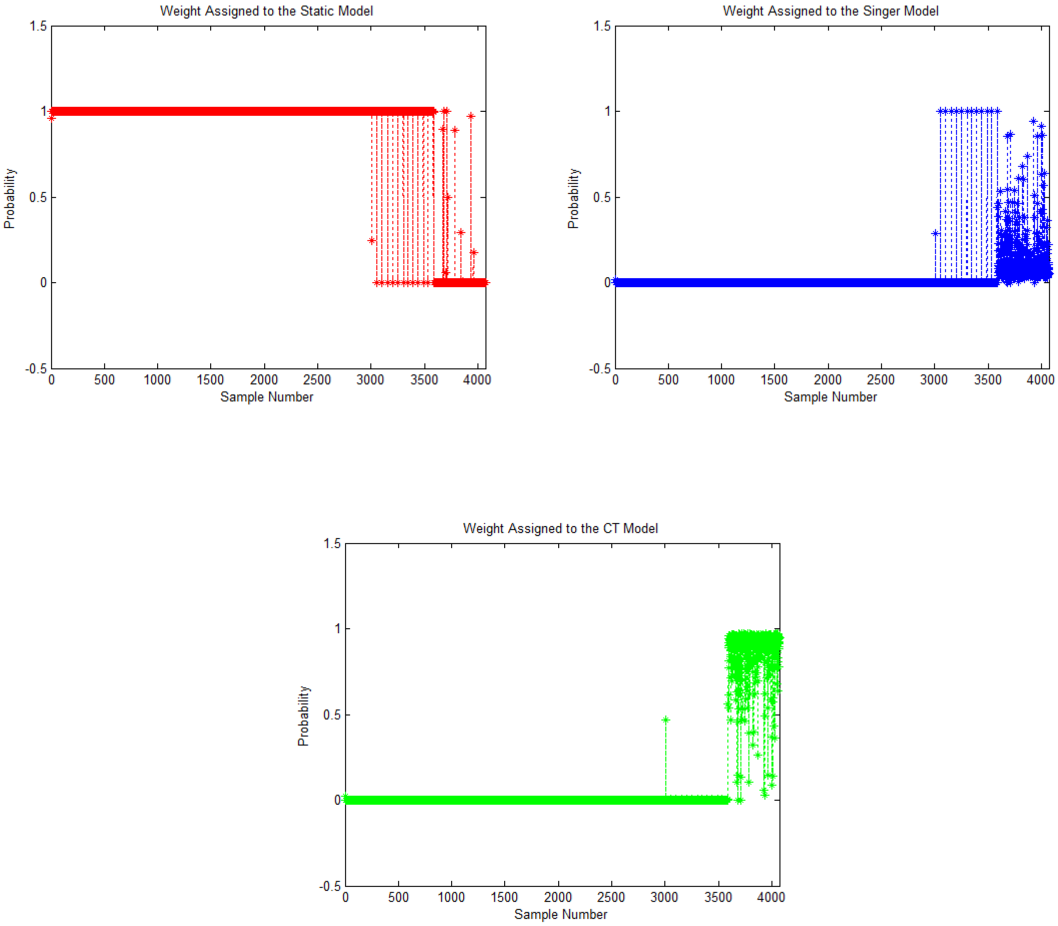


Figure 5.10: Shifting of weights between Model 1, 2 and 3

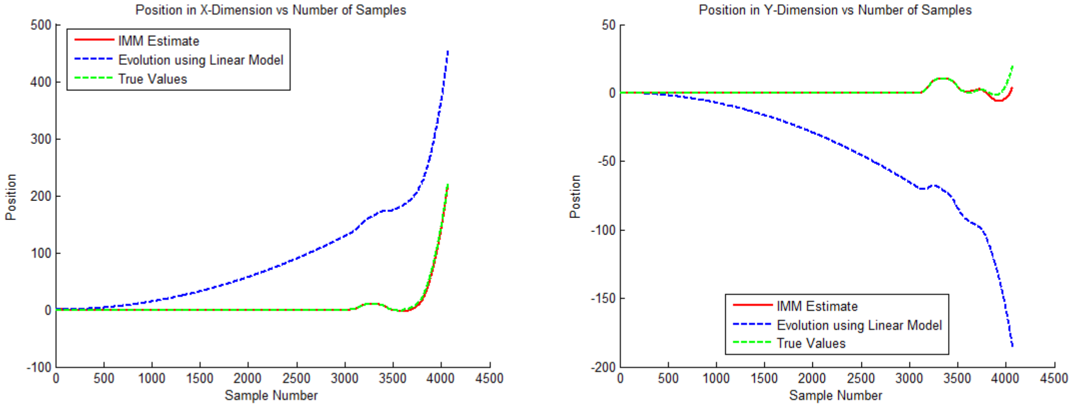


Figure 5.11: Comparison of the Position Estimates in meters in X and Y Dimension

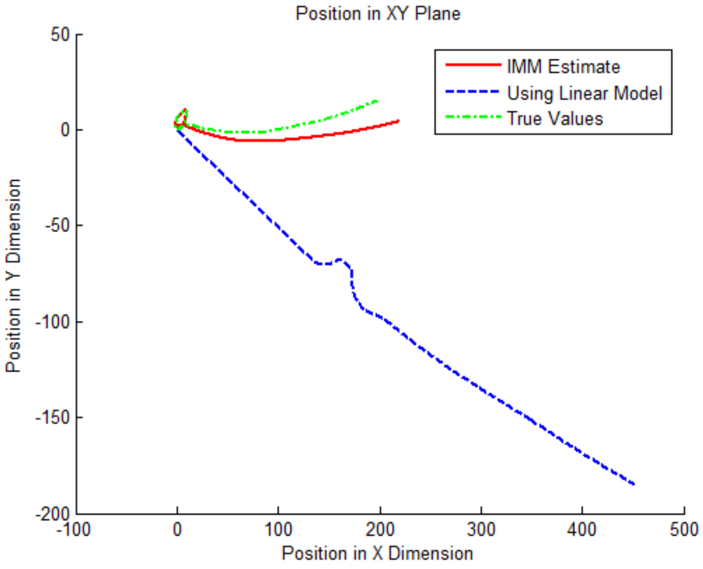


Figure 5.12: Comparison of the Position Estimates in XY Plane (unit: meters)

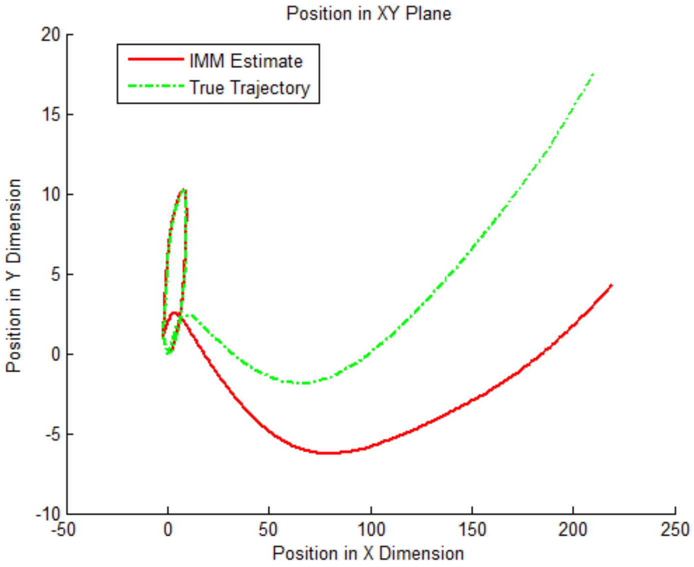


Figure 5.13: A Zoom-In of the Figure 5.12 showing IMM Estimate compared to True values of Position (unit: meters)

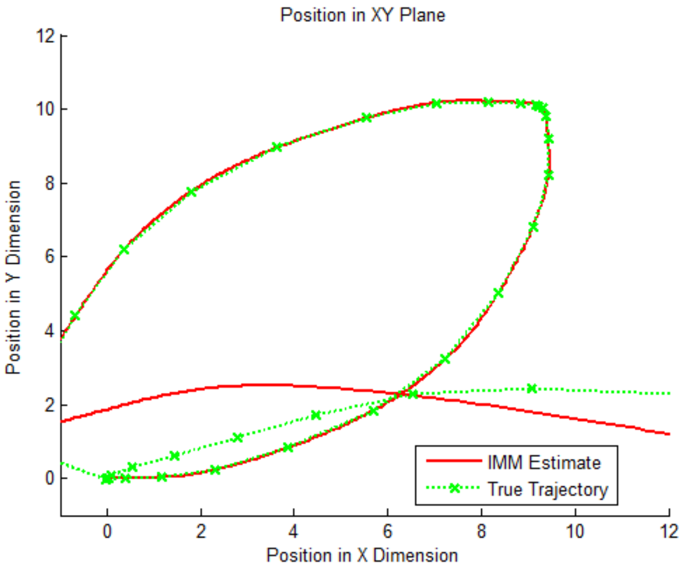


Figure 5.14: A Further Zoom-In of the Figure 5.12 showing Quality of IMM Estimation (unit: meters)

As compared to IMM scheme with 2 models, here the results are slightly less accurate. This is because the trajectory in this case is more complex and the maneuvers are harder to distinguish. But still compared to single model filter, the estimates provided by the IMM are far much better which can be seen below:

5.3.1 Performance comparison with Single Model based Filters

Again, we processed the set of measurements describing the above trajectory using single model based filters, as before, the results are extremely poor. Again the IMM filter proves to provide the best estimates closest to the true values with minimum error in both dimensions compared to the other filters. See the figures on following pages.

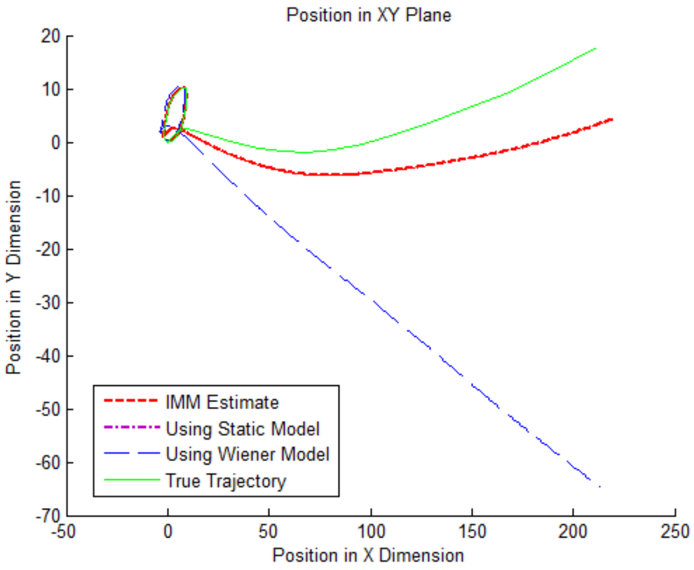


Figure 5.15: Comparison of the Position Estimates using IMM Filter and Single Model Filters (unit: meters)

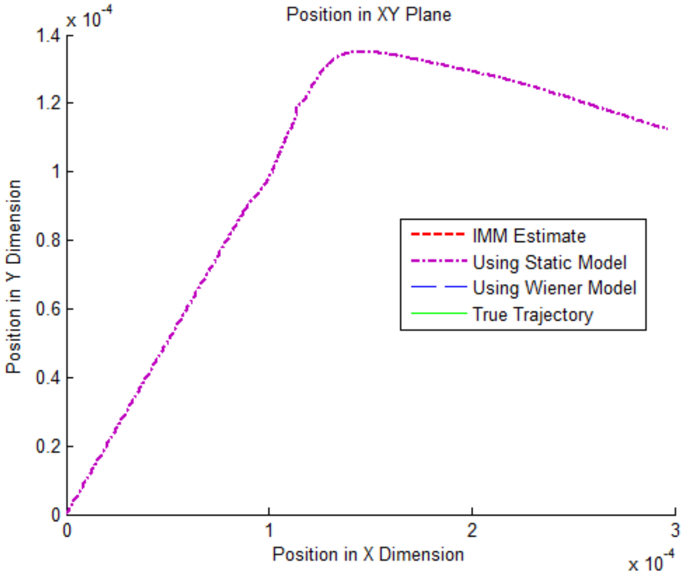


Figure 5.16: A Zoom-In of fig 5.15 showing Position Estimate obtained by using only Static Model (unit: meters)

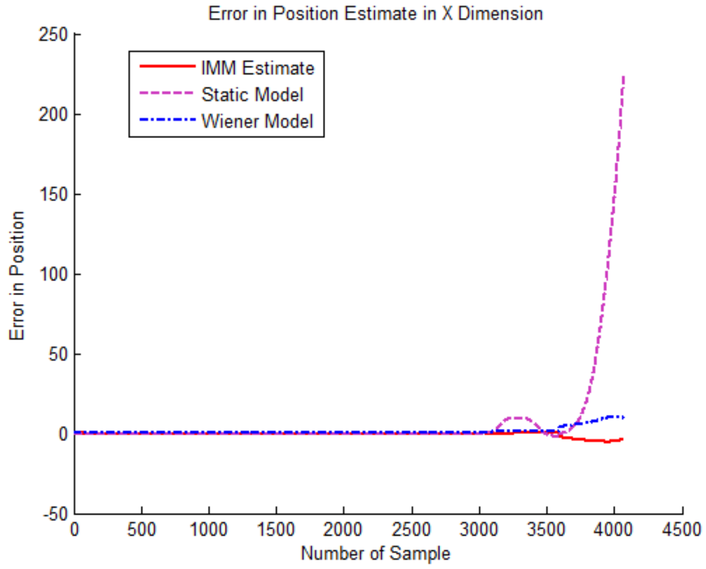


Figure 5.17: Error in Position Estimates in meters in X Dimension

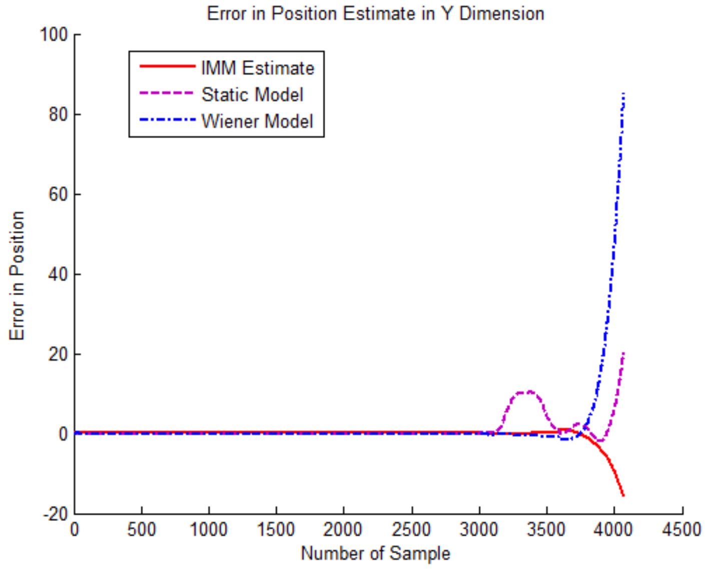


Figure 5.18: Error in Position Estimates in meters in Y Dimension

CHAPTER 6

EXPERIMENTAL RESULTS

After Analyzing the INS Characteristics in Chapter 2, studying Signal Processing techniques in Chapter 3, developing models in Chapter 4 and performing their integration into the IMM in Chapter 5, we are now ready to process real INS motion data to track a moving target. In this chapter, details are provided about the experimental setup and the results of comparison in processing INS data with single model based filters and IMM filter.

6.1 Experimental Setup

For doing experiments with INS motion data, we mounted the INS device on a small electric car which moves on tracks with controllable acceleration. These tracks can be connected in different ways and provide power to the car to run the motor which provides acceleration through contact brushes.

The models we can use with this experimental setup are the first 3 models in i.e. Static Model to suppress drift in stationary mode and Wiener and Singer model to track the vehicle while in motion. The fourth model i.e. Coordinate Turn model is basically for aircrafts where the turn rate of the aircraft vehicle can be controlled by the pilot. Since this is not possible with this experimental setup, we did our experiments with the linear trajectories on tracks with length of approximately 1.2 meters.

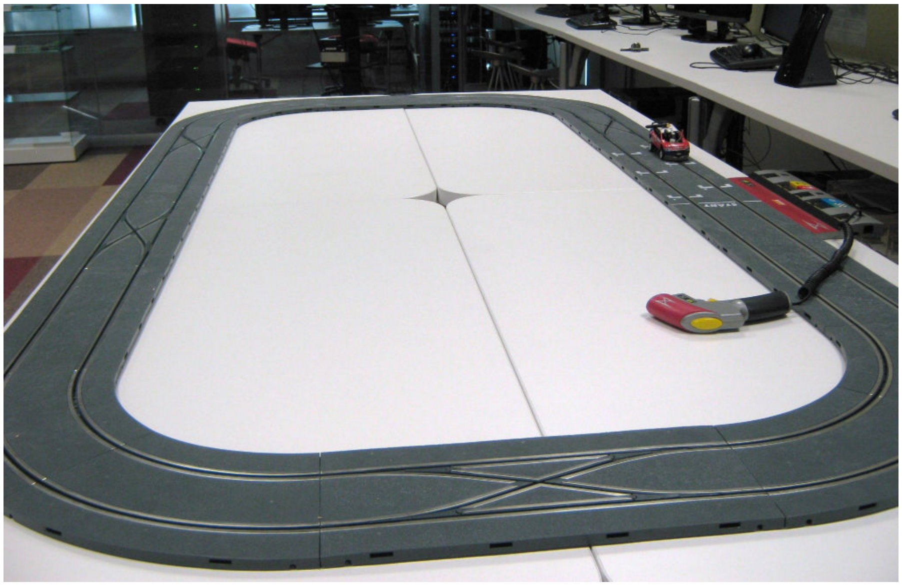


Figure 6.1: The Experimental Setup

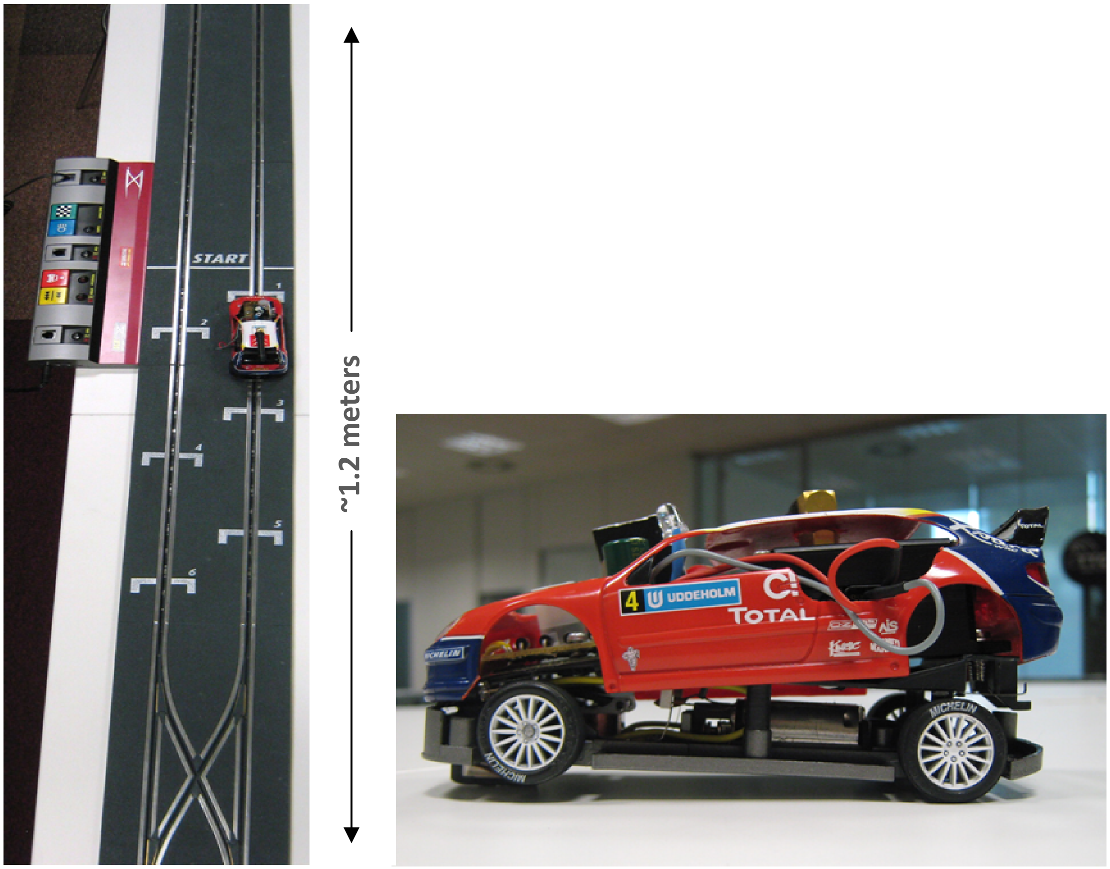


Figure 6.2: Length of the Track and the Electric Car used in the Experiments

In order to mount the INS device on the car, we had to perform its integration with the power supply of the car which it obtains from the electric lines in the tracks. This was a critical issue, as the car is equipped with motors inside it and it could induce noise in the power supply and vibration in the INS device platform creating additional measurement noise. By carefully designing the electrical power circuit, we were able to provide INS device a smooth power supply.

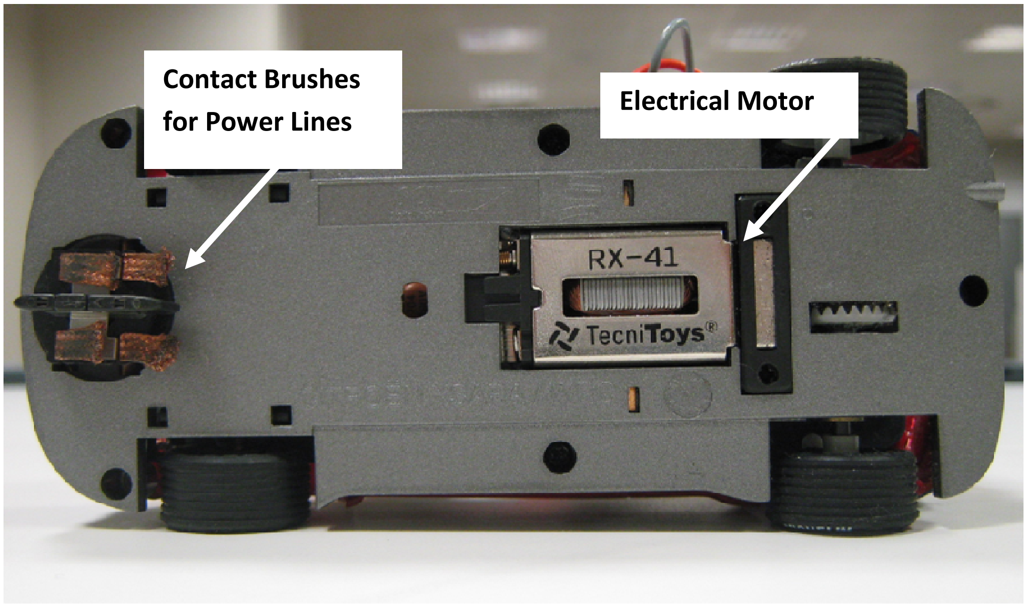


Figure 6.3: The Electric Car's Contact Brushes for Power Supply and Electric Motor for Acceleration

6.2 Mechanical integration of inertia-link Wireless Sensor

The parameters and requirements of this integration are:

Power supply system using the lane power lines

- Lane voltage: 18 V DC
- Inertia-Link IMU voltage requirement: 5.2 – 16 V DC

Issues:

- On-Off Keying (OOK) scalextric telemetry every 100 ms on power lines
- PWM-driven motor spurious interferences
- Lane power line metallic brushes contacts: Sparks and sporadic micro power cuts

Solution:

- EMC Filtered DC to DC linear voltage converter based on MC7806 linear voltage regulator
- Output capacitor: 1500 uF

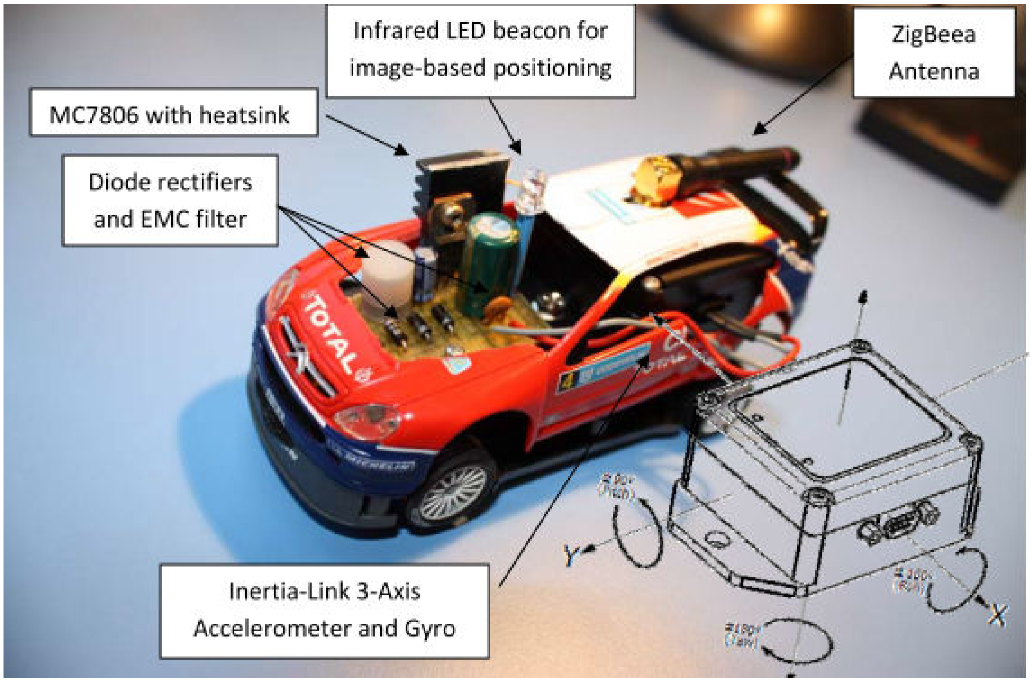
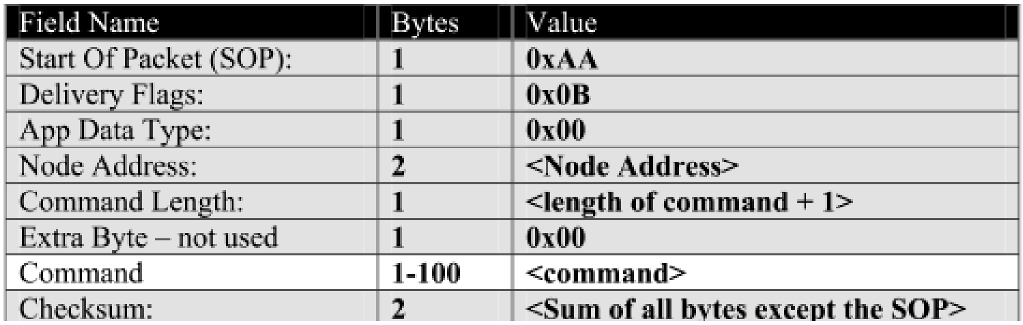

Infrared LED beacon for image-based positioning
ZigBeea Antenna
MC7806 with heatsink
Diode rectifiers and EMC filter
Inertia-Link 3-Axis Accelerometer and Gyro
TOTAL
±90° (Pitch)
Y
(Roll)
X
±180° (Yaw)

Figure 6.4: Integration of IMU with the Car and Electronic Power Circuit for Smooth Power to IMU device

6.3 Communications Protocol

Inertia Link IMU is a wireless IMU device and data is received from the IMU sensors using a wireless Zig-Bee protocol. However, the IMU device comes with a software driver package which simplifies access to the IMU by providing a simpler interface. The simpler protocol is required to route the correct commands to obtain the wireless sensor data to the correct sensor and re-route the reply of the sensor to the host collecting the data. We had to develop a software code with this protocol to access the data regarding acceleration, angular rate and orientation matrix measurements. The protocol is defined by the Inertia Link manual as:

6.3.1 Command Packet (from the host PC to the sensor)



Field Name	Bytes	Value
Start Of Packet (SOP):	1	**0xAA**
Delivery Flags:	1	**0x0B**
App Data Type:	1	**0x00**
Node Address:	2	**<Node Address>**
Command Length:	1	**<length of command + 1>**
Extra Byte – not used	1	**0x00**
Command	**1-100**	**<command>**
Checksum:	2	**<Sum of all bytes except the SOP>**

Table 6.1: Protocol for Sending Command

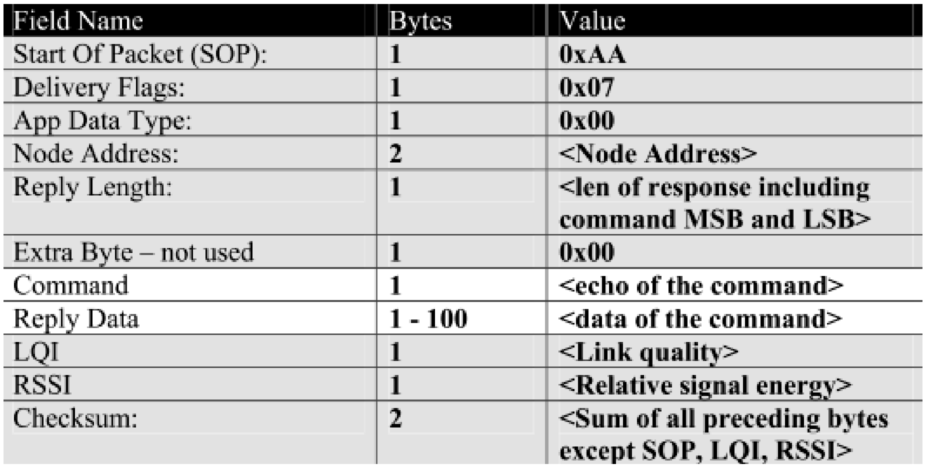
All fields preceding 'Command' bytes are required by the protocol to for routing purposes. The field 'Node-Address' provides the address of the sensor which in our case was 89. The 'Command' byte determines which data is sent as a reply.

The ACK/NACK byte

The command packet is always followed by an ACK or NACK byte sent back to the PC from the wireless sensor. The ACK byte value is 0xAA and the NACK byte value is 0x21. The ACK signifies that the packet was accepted and broadcast by the wireless base station to the wireless sensor. A NACK indicates that the packet was not accepted for any reason such as error.

6.3.2 The Data Packet (from the sensor to the PC)

The data packet is essentially the same as command packet except for the additional bytes which carry the reply data from the sensor. Each reply byte has its own representation format described in the Inertia Link manual [6] according to which it needs to be interpreted.



Field Name	Bytes	Value
Start Of Packet (SOP):	1	**0xAA**
Delivery Flags:	1	**0x07**
App Data Type:	1	**0x00**
Node Address:	2	**<Node Address>**
Reply Length:	1	**<len of response including command MSB and LSB>**
Extra Byte – not used	1	**0x00**
Command	1	**<echo of the command>**
Reply Data	1 - 100	**<data of the command>**
LQI	1	**<Link quality>**
RSSI	1	**<Relative signal energy>**
Checksum:	2	**<Sum of all preceding bytes except SOP, LQI, RSSI>**

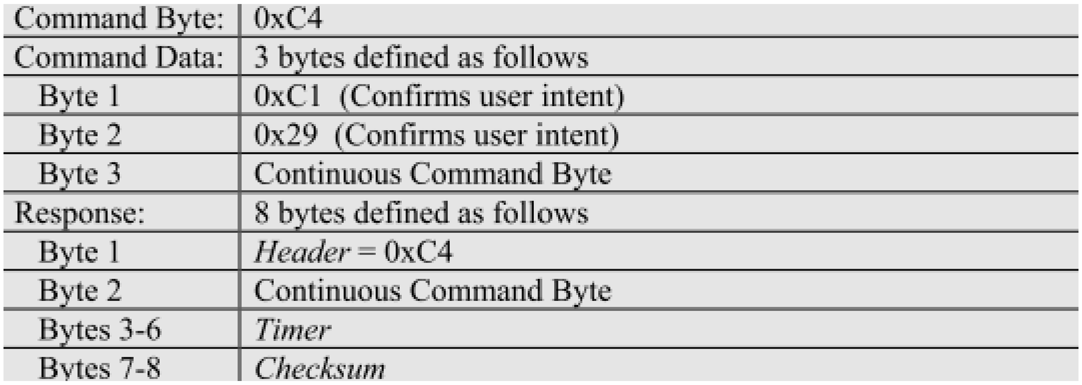
Table 6.2: Format of the Reply Packet

6.3.3 Setting up Output Mode

The two commands that we require to obtain the inertial data from the IMU at the continuous rate of 100Hz using the above protocol are:

1. Set Continuous Mode (0xC4)

This command sets the Inertia Link in a continuous mode in which it outputs the data continuously at a rate of 100Hz i.e. 0.01 seconds of sampling time. This is very necessary because accuracy of tracking depends of data rate and sampling time period and polling technique cannot work as there are chances of missing important packets while data is being processed by the PC. In this way data keeps on collecting in the PC buffer and recorded in a .csv (comma separated values) file.

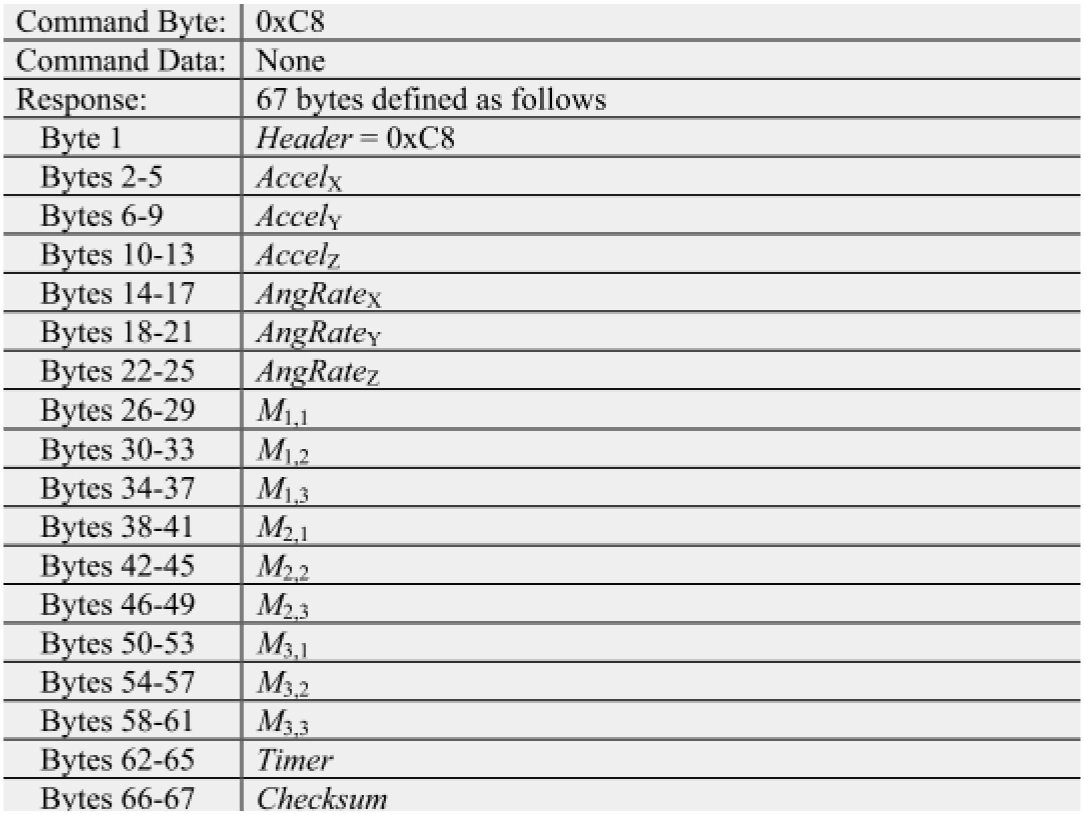


Command Byte:	0xC4
Command Data:	3 bytes defined as follows
Byte 1	0xC1 (Confirms user intent)
Byte 2	0x29 (Confirms user intent)
Byte 3	Continuous Command Byte
Response:	8 bytes defined as follows
Byte 1	*Header* = 0xC4
Byte 2	Continuous Command Byte
Bytes 3-6	*Timer*
Bytes 7-8	*Checksum*

Table 6.3: Format of Continuous Mode Command

2. Acceleration, Angular Rate & Orientation Matrix (0xC8)

This is the command to obtain the measurements of Acceleration, Angular Rate and Orientation Matrix. This command is sent in the ‘command byte’ field of the ‘Set Continuous Mode’ command so that measurements can be obtained with 100Hz sampling.



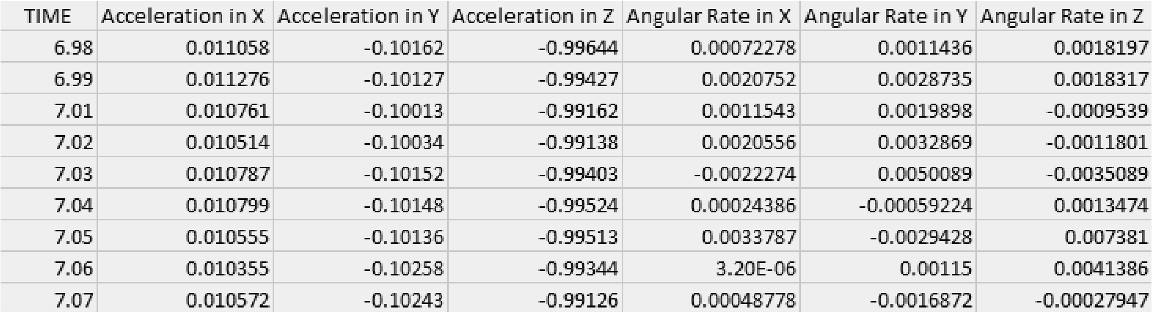
Command Byte:	0xC8
Command Data:	None
Response:	67 bytes defined as follows
Byte 1	*Header* = 0xC8
Bytes 2-5	$Accel_X$
Bytes 6-9	$Accel_Y$
Bytes 10-13	$Accel_Z$
Bytes 14-17	$AngRate_X$
Bytes 18-21	$AngRate_Y$
Bytes 22-25	$AngRate_Z$
Bytes 26-29	$M_{1,1}$
Bytes 30-33	$M_{1,2}$
Bytes 34-37	$M_{1,3}$
Bytes 38-41	$M_{2,1}$
Bytes 42-45	$M_{2,2}$
Bytes 46-49	$M_{2,3}$
Bytes 50-53	$M_{3,1}$
Bytes 54-57	$M_{3,2}$
Bytes 58-61	$M_{3,3}$
Bytes 62-65	*Timer*
Bytes 66-67	*Checksum*

Table 6.4: Command for Inertial Data Measurements

6.3.4 Collecting Data

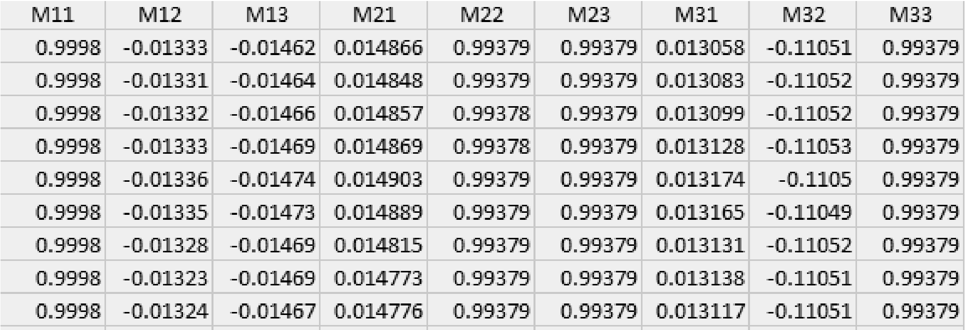
Data from the serial buffer is read using Matlab code and the bytes are interpreted in the format they are represented in the Inertia Link communications protocol manual [6]. A .csv file is used to then save this data for offline processing. The format of the .csv file is given below:

Column 1:7



TIME	Acceleration in X	Acceleration in Y	Acceleration in Z	Angular Rate in X	Angular Rate in Y	Angular Rate in Z
6.98	0.011058	-0.10162	-0.99644	0.00072278	0.0011436	0.0018197
6.99	0.011276	-0.10127	-0.99427	0.0020752	0.0028735	0.0018317
7.01	0.010761	-0.10013	-0.99162	0.0011543	0.0019898	-0.0009539
7.02	0.010514	-0.10034	-0.99138	0.0020556	0.0032869	-0.0011801
7.03	0.010787	-0.10152	-0.99403	-0.0022274	0.0050089	-0.0035089
7.04	0.010799	-0.10148	-0.99524	0.00024386	-0.00059224	0.0013474
7.05	0.010555	-0.10136	-0.99513	0.0033787	-0.0029428	0.007381
7.06	0.010355	-0.10258	-0.99344	3.20E-06	0.00115	0.0041386
7.07	0.010572	-0.10243	-0.99126	0.00048778	-0.0016872	-0.00027947

Column 8:16



M11	M12	M13	M21	M22	M23	M31	M32	M33
0.9998	-0.01333	-0.01462	0.014866	0.99379	0.99379	0.013058	-0.11051	0.99379
0.9998	-0.01331	-0.01464	0.014848	0.99379	0.99379	0.013083	-0.11052	0.99379
0.9998	-0.01332	-0.01466	0.014857	0.99378	0.99379	0.013099	-0.11052	0.99379
0.9998	-0.01333	-0.01469	0.014869	0.99378	0.99379	0.013128	-0.11053	0.99379
0.9998	-0.01336	-0.01474	0.014903	0.99379	0.99379	0.013174	-0.1105	0.99379
0.9998	-0.01335	-0.01473	0.014889	0.99379	0.99379	0.013165	-0.11049	0.99379
0.9998	-0.01328	-0.01469	0.014815	0.99379	0.99379	0.013131	-0.11052	0.99379
0.9998	-0.01323	-0.01469	0.014773	0.99379	0.99379	0.013138	-0.11051	0.99379
0.9998	-0.01324	-0.01467	0.014776	0.99379	0.99379	0.013117	-0.11051	0.99379

Figure 6.5: Format of CSV File use for Recording Measurements

6.3.5 Orientation/Rotation Matrix

The orientation matrix is based upon 9 elements as shown below. This matrix is required to transform the measurements of acceleration from local body frame to the global body frame. For simple linear motion, we will be using the local body frame as it is more convenient to visualize a straight line motion in local body frame. For curvilinear motion, the measurements in global body frame are better.

$$M = \begin{bmatrix} M11 & M12 & M13 \\ M21 & M22 & M23 \\ M31 & M32 & M33 \end{bmatrix}$$

The orientation matrix describes the attitude of the IMU device and vehicle it is mounted on. But, as shown in Chapter 2, Inertia Link is not equipped with a magnetometer and hence it cannot provide heading information correctly. We expect that this will cause error when converting from local body frame to the global body frame as well.

6.4 Tracking the Vehicle

With the above experimental setup we recorded many realizations of a linear trajectory about 1.2 meters in length. Some of the realizations also included stationary phase before or after the motion which contained noise that without eliminating would result in drift of the position estimates. Consider the results in Figures 6.6 - 6.10 to see the performance of IMM in detecting in which mode, stationary or motion, the vehicle is and providing the position estimates:

Realization 1:

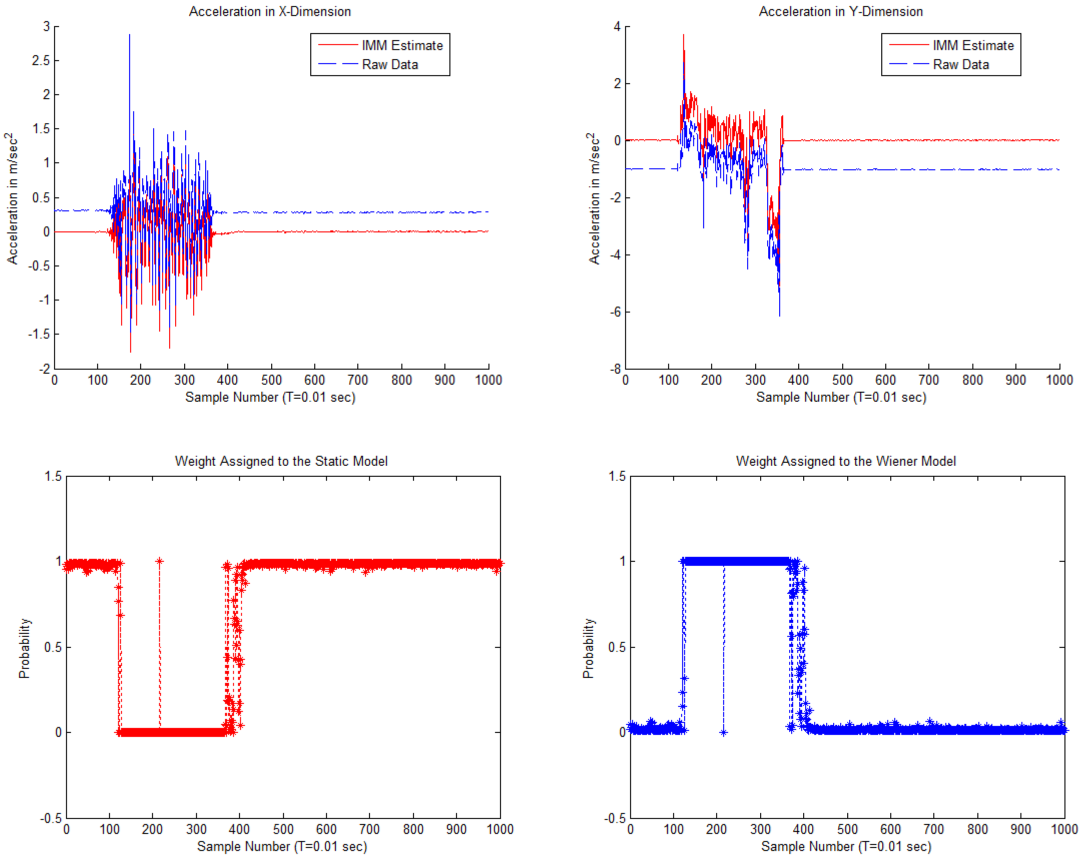


Figure 6.6: Measurements of Acceleration and IMM Model Switching

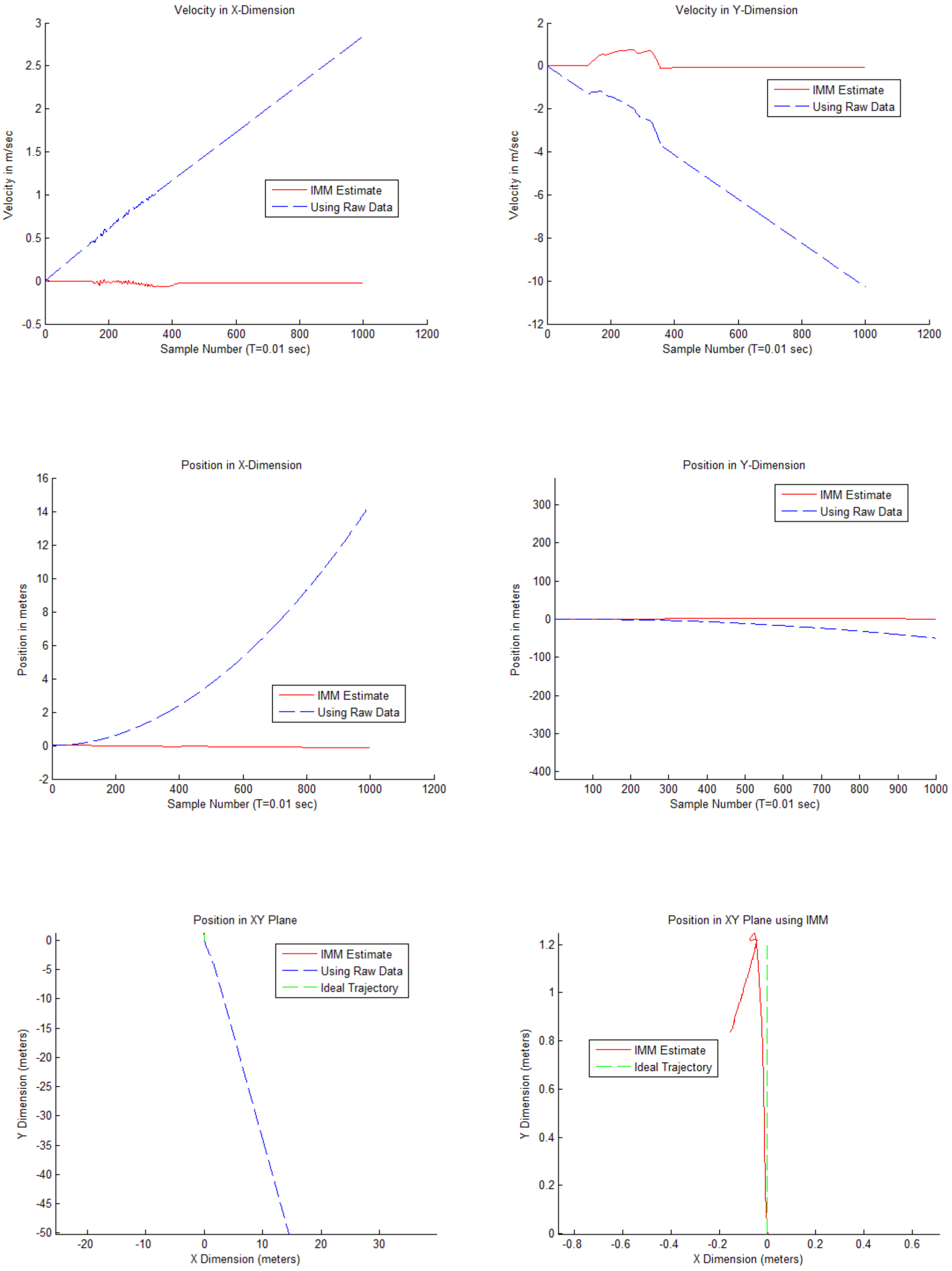


Figure 6.7: Result of Processing data with IMM Filter for Realization 1

Realization 2:

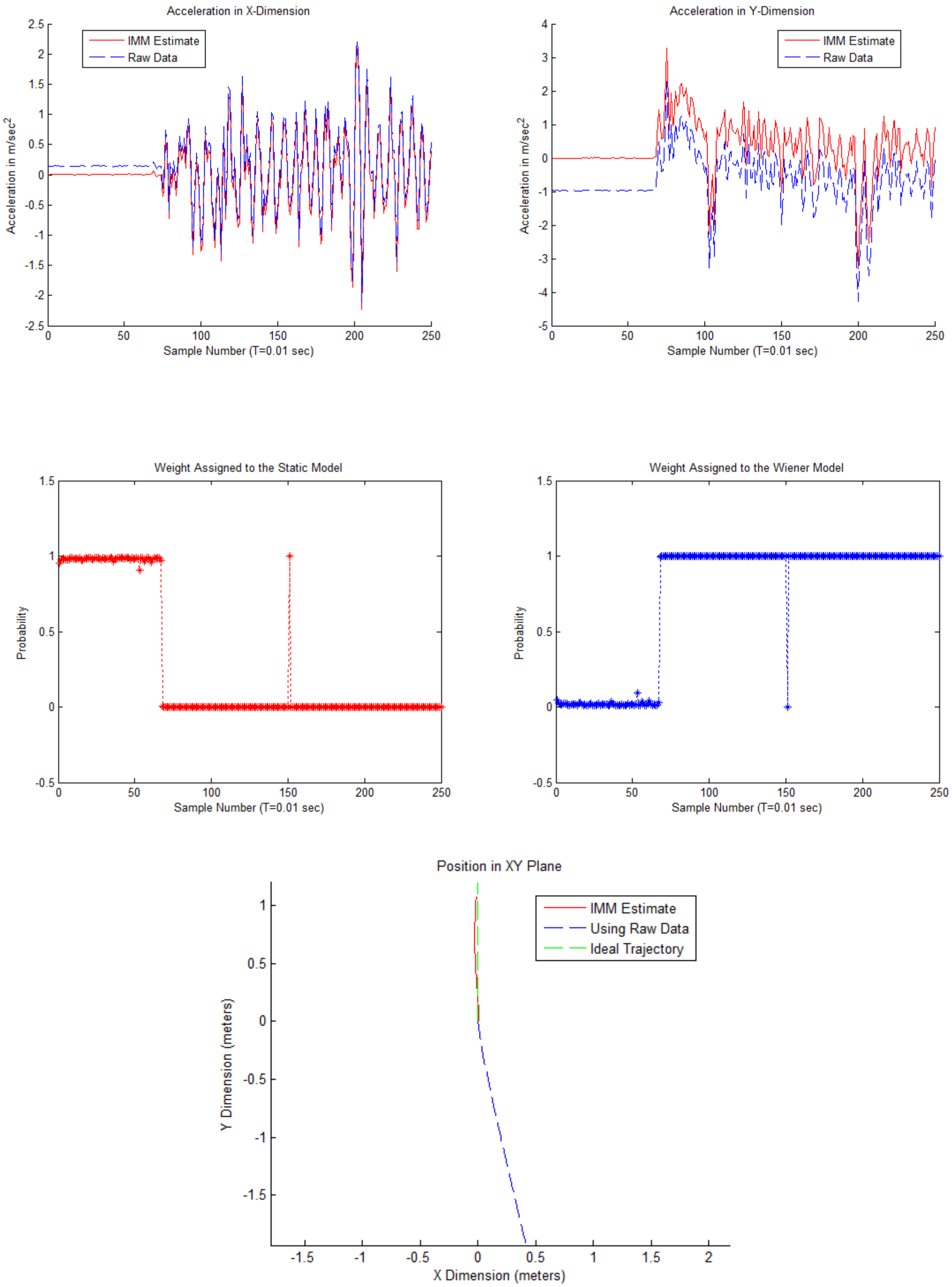


Figure 6.8a: Comparison of Position Estimates for Realization 2

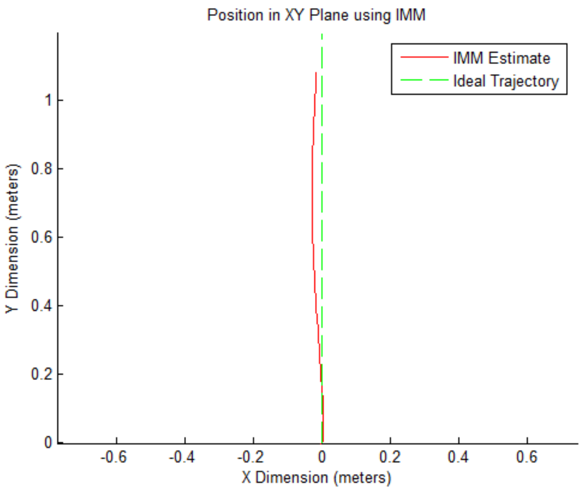


Figure 6.8b: Result of Processing data with IMM Filter for Realization 2

Realization 3:

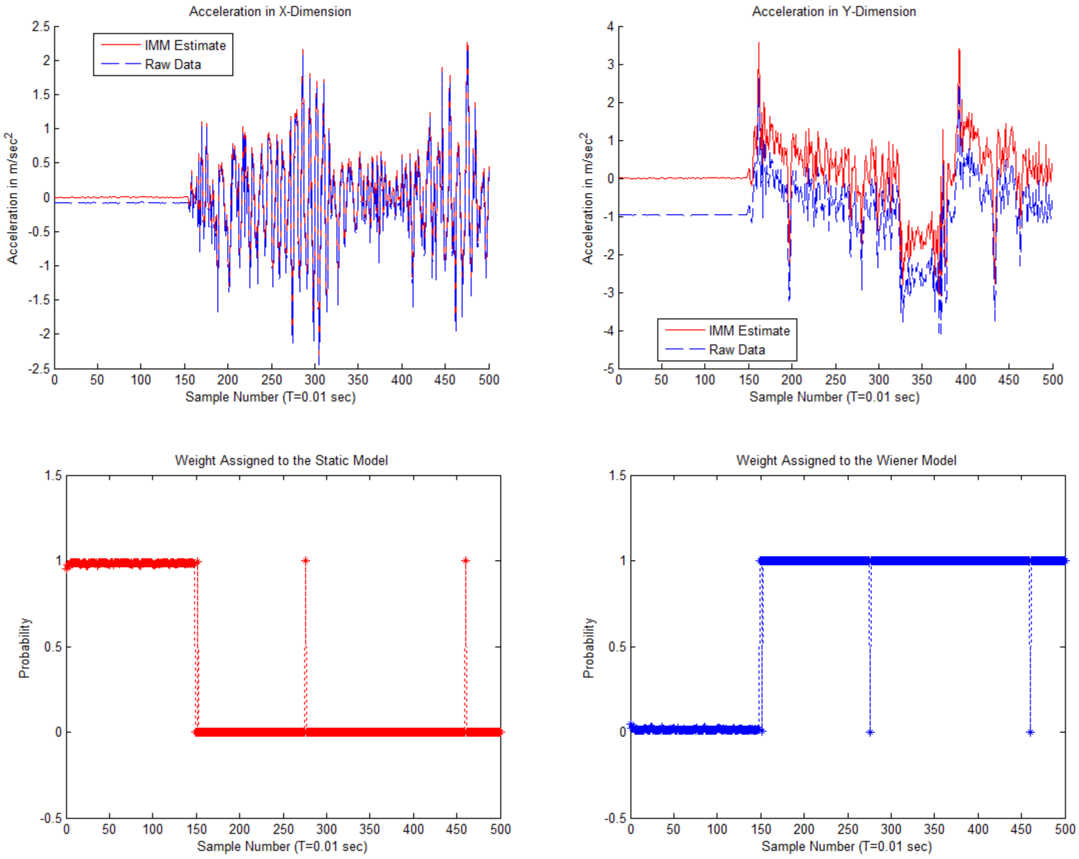


Figure 6.9: Measurements of Acceleration and IMM Model Switching for Realization 3

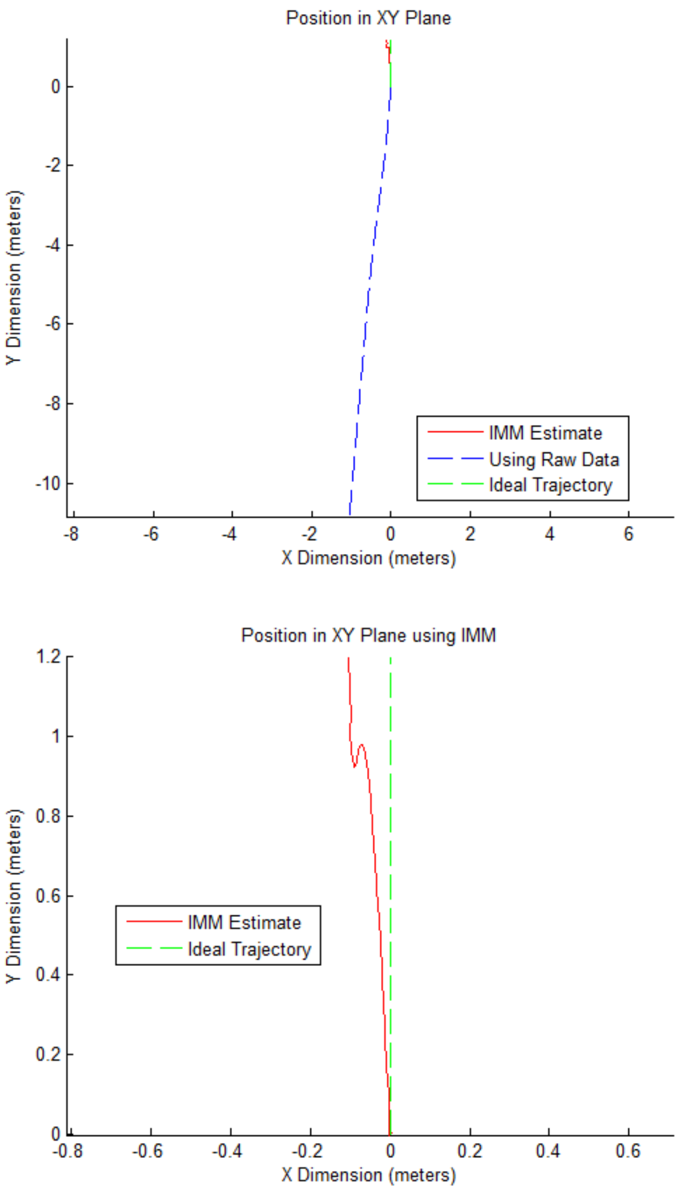


Figure 6.10: Result of Processing data with IMM Filter for Realization 3

These examples show IMM performance in selecting the right model. The performance of the IMM algorithm is very dynamic and this is possible by selecting the right models and tuning their parameters and covariances carefully. But the essential question is what are the needs and advantages of IMM techniques over conventional single model based filters. In the next section, this question is answered by providing such analysis which empirically proves the superior performance of IMM algorithm in tracking a real target.

6.5 IMM vs. Single Model based Filters in Real Target Tracking

In order to show the performance difference between the IMM scheme and conventional filters, three cases are presented which make the performance and advantages of IMM scheme quite obvious.

Case 1: We have the realization 2 as same as above. We use the Wiener and Static model based filters on it. The Resultant position estimates are:

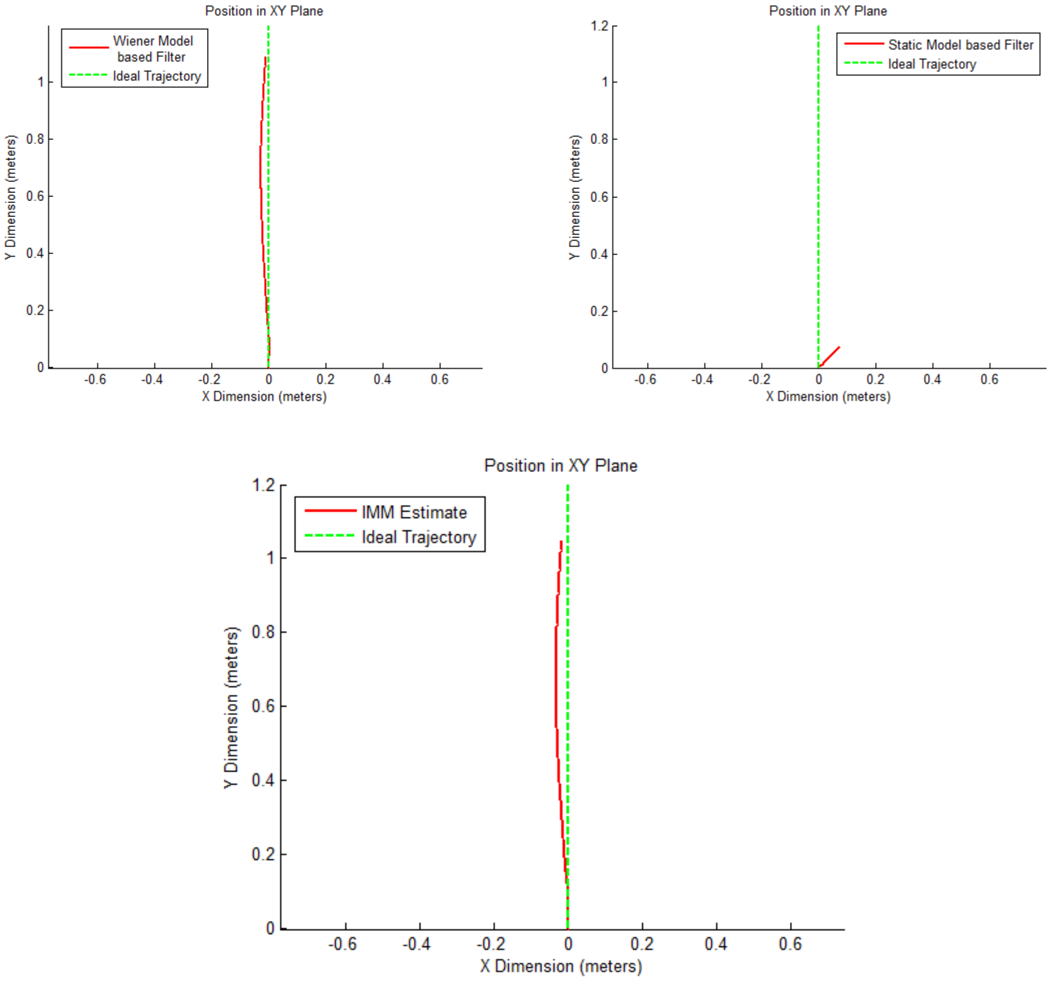


Figure 6.11: Comparison of Estimates for Realization 2 Data with IMM, Wiener and Static model based Filters

Clearly, while Wiener model was able to perform as good as IMM in estimating close to true trajectory, the Static model failed, suppressing all the motion by considering the acceleration measurements as noise.

Case 2: We have a realization of the vehicle in stationary mode for a period of 50 seconds. The performance of different filters in suppressing the noise can be seen here:

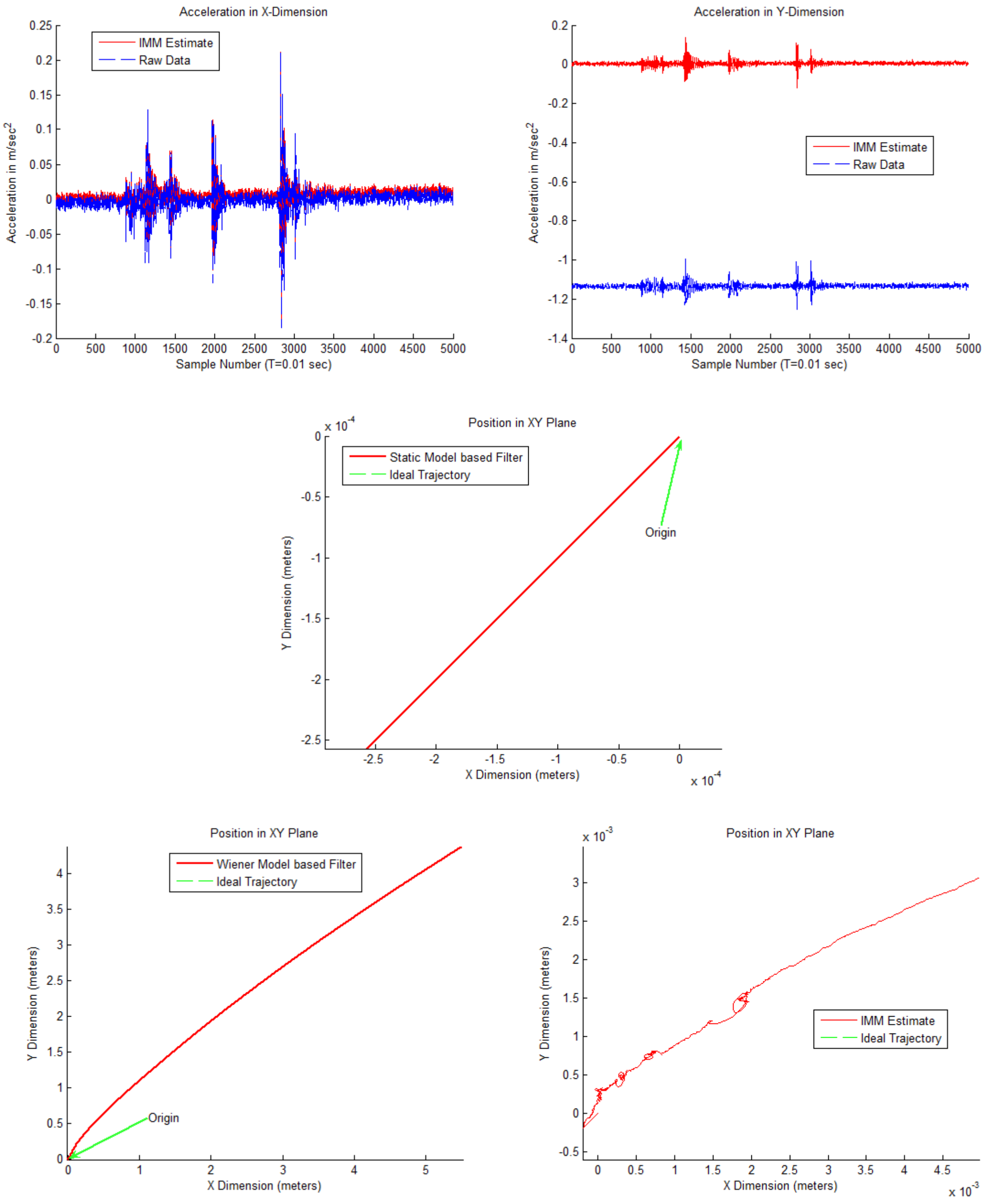


Figure 6.12: Comparison between IMM, Wiener and Static model based Filters

Here, as expected, the static model does suppresses the noise but Wiener model, which allows a high process variance, takes the noise samples and process them to estimate false position which cause large drift from the true position value which should be the origin in this case. IMM, in this case also, by selecting correct weights for the model, performs almost as good as the Static Model based filter.

Case 3: We had a vehicle perform some straight line motion and then come to rest and stay in that mode for a while. During the stationary state, noise is present in the acceleration measurements. It can be seen that Wiener model is unable to suppress this noise in stationary phase whereas static model is unable to detect motion in maneuvering state properly. In contrast, IMM provides the best of both the worlds. It detects motion and estimates position accurately and then when it detects low measurements due to noise, it switches to stationary mode and prevents the drift. Even though at some samples some high noise values in stationary mode cause the IMM to switch falsely to the Wiener model, still the results obtained using IMM are very good and close to the ideal trajectory.

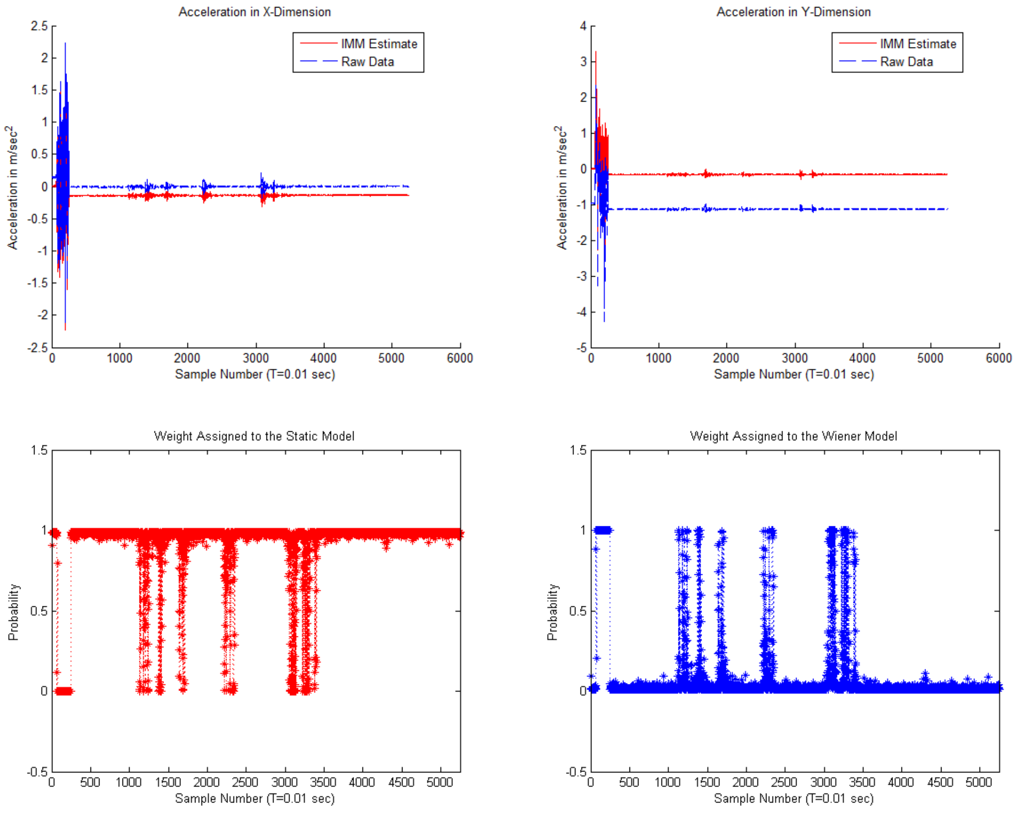


Figure 6.13: A Realization with both Stationary and Motion Phases

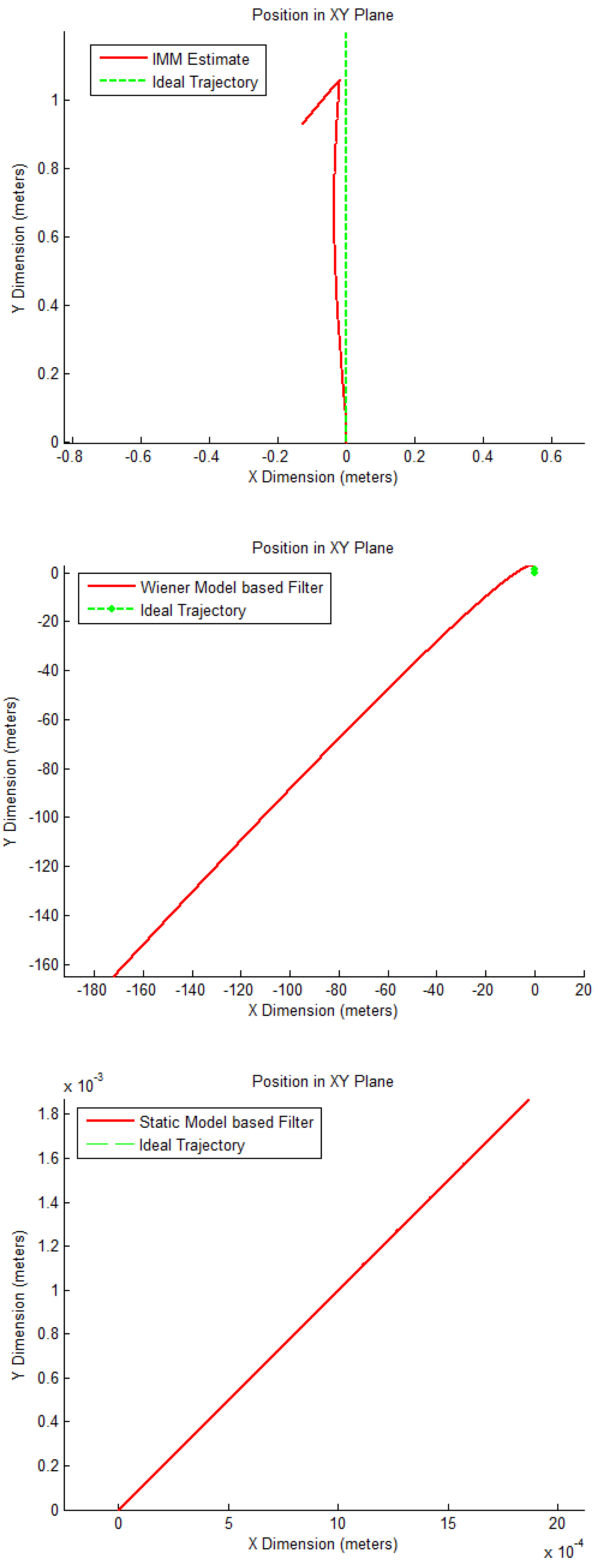


Figure 6.14: Position Estimates using IMM, Wiener and Static model based Filters

All this analysis has proven the superior performance of Interacting Multiple Model (IMM) technique over single model based filters. We have shown this difference in performance using both simulations and real data. With the proper analysis of the Inertia Measurement Unit device characteristic, carefully selecting models and using the IMM technique, the performance in maneuvering target tracking using inertial systems can be highly improved and stand-alone INS devices can be used in a number of applications.

CHAPTER 7

CONCLUSION

Achieving the task that was defined in the Problem Definition required study of different areas from Inertial Navigation Systems to Mathematical Modeling, Signal Processing and Bayesian Filtering and advance topics such as Interacting Multiple Model. The proposed solution was developed after considering multitude of different options. The potential of the IMM solution to the problem in maneuvering target tracking using Inertial Systems was tested both theoretically using simulation and practically using experiments. It was a time-constraint project is which very good performance results were obtained. Here is a summary of what were the steps taken to reach this accomplishment and what are the improvements achieved in performance through this solution. Also some suggestions for future work are included.

7.1 Steps to the Solution

- Study of the fundamentals of Inertial Navigation. The types of devices available and their characteristics.
- Analysis of Errors in Inertial System and their effect on the degradation of the Navigation Solution. Characterizing the Noise in the Measurement sensors and obtaining its parameters.
- Study of a large number of mathematical models related to Maneuvering Target Tracking and the selection of appropriate models from this pool.

- Analyzing the effect of important parameters of individual models and tuning them.
- Study of Signal Processing and Filtering techniques for Error estimation. Choosing the right covariances.
- Integration of selected models into the Interacting Multiple Model Scheme.
- Running Simulations to test the performance of IMM filter compared to conventional filters.
- Setting up experimental equipment for testing the performance of the suggested solution in real world scenario. Integrating the IMU device with the experimental setup.
- Developing software for communications with the IMU and recording the measurements.
- Estimating the trajectory of a moving target using the proposed solution. Analysis and comparison of final results.

7.2 Achievements

The improvements achieved have been shown and proven through both simulations and real target tracking experiments. Here is a summary,

- Eliminated the Error in Navigation Solution due to Bias and Noise. Added immunity to the drift in stationary mode.
- Developed Models capable of describing different types of maneuvering and non-maneuvering states.
- Developed a robust technique of identifying the current maneuver in progress in real-time and estimating the true trajectory.
- Immense improvement in the accuracy of Position Estimation using Inertial Systems.
- Increased the reliability of low-cost Inertial devices.
- Increased the possibilities of the use of Stand-alone INS systems.

7.3 Possible Future Improvements

The solution has provided a ground work over which there is a lot of space for creativity. Here are some of the suggestions.

- Integration with Wireless Sensor Networks for indoor Navigation.
- Integration with cellular networks localization data for outdoor Navigation.
- Use of advance filtering techniques such as Particle Filters for IMU devices having non-Gaussian noise in sensors.
- Using advance versions of IMM algorithm based on sojourn time spent in a particular maneuver to anticipate the next maneuver.

7.4 Final Remarks

I conclude this Thesis with the statement that based on the performance evaluations; this thesis has been tested successful in achieving the targets set in the Problem definition.

References

1. Oliver J. Woodman, An **introduction to Inertial Navigation**, University of Cambridge, Computer Laboratory Technical Report, August 2007.
2. Kevin J. Walchko, Paul A. C. Mason, **Inertial Navigation**, Florida Conference on Recent Advances in Robotics, 2002.
3. Zhiqiang Xing, Gebre-Egziabher, D., **Modeling and Bounding Low Cost Inertial Sensor Errors,** Position, Location and Navigation Symposium, 2008 IEEE/ION Page(s): 1122 – 1132, 2008.
4. Blankinship, K.G.; **A General Theory for Inertial Navigator Error Modeling,** Position, Location and Navigation Symposium, 2008 IEEE/ION Page(s):1152 – 1166, 2008.
5. Mohinder S. Grewal, **Global Positioning Systems Inertial Navigation and Integration** 2nd Ed, Wiley Press, ISBN-13 978-0-470-04190-1, 2007.
6. Microstrain, **3DM-GX2® Data Communications Protocol for Inertia-Link®**, Version 1.17, 2009. www.microstrain.com.
7. Wei Wang, Dan Wang, **Land Vehicle Navigation Using Odometry/INS/Vision Integrated System** Cybernetics and Intelligent Systems, 2008 IEEE Conference on Page(s): 754 – 759, September 2008.
8. B.A.Ragel, M.Farooq, **Comparison of Forward Vs. Feedback Kalman Filter For Aided Inertial Navigation System,** 7th International Conference on Information Fusion (FUSION), 2005.
9. Yuanxin Wu,Xiaoping Hu, Meiping Wu, Dewen Hu, **Strapdown Inertial Navigation using Dual Quaternion Algebra: Error Analysis,** Aerospace and Electronic Systems, IEEE Transactions on, Page(s): 259 – 266, 2006.
10. Carles Fernandez–Prades, Jordi Vila–Valls, **Bayesian Nonlinear Filtering Using Quadrature and Cubature Rules Applied To Sensor Data Fusion For Positioning**, IEEE International Conference on Communications, Cape Town, South Africa, 2010.
11. Billur Barshan, Hugh F. Durrant, **Inertial Navigation Systems for Mobile Robots**, IEEE Transactions On Robotics And Automation, Vol. 11, No. 3, June 1995.
12. Steven M. Kay, **Fundamentals of Statistical Signal Processing: Estimation Theory**, Prentice Hall Press, 1993.
13. Rong Li, X., Jilkov V.P., **Survey of Maneuvering Target Tracking. Part I: Dynamic Models,** Aerospace and Electronic Systems, IEEE Transactions on, Volume: 39, Issue: 4, Page(s): 1333 – 1364, 2003.
14. Singer, R.A., **Estimating Optimal Tracking Filter Performance for Manned Maneuvering Targets,** Aerospace and Electronic Systems, IEEE Transactions on, Volume: AES-6 , Issue: 4, Page(s): 473 – 483, 1970.

15. Singer, R.A., Behnke, K.W., **Real-Time Tracking Filter Evaluation and Selection for Tactical Applications,** Aerospace and Electronic Systems, IEEE Transactions on, Volume: AES-7 , Issue: 1, Page(s): 100 – 110, 1971..
16. Gustafsson, F., Isaksson, A.J., **Best Choice of Coordinate System for Tracking Coordinated Turns** Decision and Control, Proceedings of the 35th IEEE, Volume: 3, Page(s): 3145 - 3150, 1996.
17. Mazor E., Averbuch A., Bar-Shalom, Y., Dayan J., **Interacting Multiple Model Methods in Target Tracking: A Survey,** Aerospace and Electronic Systems, IEEE Transactions on, Volume: 34, Issue: 1, Page(s): 103 – 123, 1998.
18. Rong Li, X., Jilkov V.P., **Survey of Maneuvering Target Tracking. Part V: Multiple-Model Methods,** Aerospace and Electronic Systems, IEEE Transactions on, Volume: 41 , Issue: 4 Page(s): 1255 – 1321, 2005.
19. Rong Li, X., Jilkov, V.P., **A Survey of Maneuvering Target Tracking—Part IV: Decision-Based Methods**, Aerospace and Electronic Systems, IEEE Transactions on, Volume: 41 , Issue: 4 Pages 1255 - 1321, 2005.
20. Wayne Schmaedeke, Keith Kastella, **Sensor Management using Discrimination Gain and Interacting Multiple Model Kalman Filters,** Lockheed-Martin Tactical Defense Systems, IEEE Transactions on Systems, Man and Cybernetics, January, 1998.