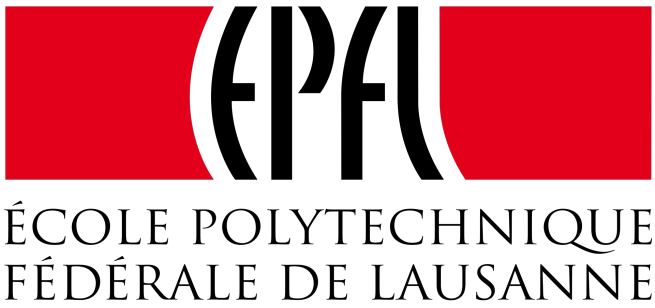



# Introducing Vehicle Dynamic Models in Dynamic Networks for Navigation in UAVs

Master Thesis

January 26, 2019

Author :

*Kenneth Joseph Paul*

Supervisors :

*Jan Skaloud*

*Dario Floreano*

Assistant :

*Davide Antonio Cucci*

**Abstract**

Estimation of the trajectory is a fundamental problem in robotics. Introduction of additional measurements in a robotic platform reduces the uncertainty in the trajectory estimate. The limitations on the power and payload in a UAV platform advocates for the usage of already existing inputs such as Vehicle Dynamic Model over the addition of new sensors. These inputs from the Vehicle Dynamic Model are independent from the lighting and texture of the surrounding environment. Different methods exist in the literature for multi-sensor fusion problems. The conventional methods of sensor fusion are outperformed by the graph-based methods such as Dynamic Network, which could model observations by constraining them in space or time. In this work we investigate the improvement gained by introducing the Vehicle Dynamic Model as measurements in a Dynamic Network for a Quadcopter. We investigate two separate studies - i) estimation of the trajectory with the known sensor parameters, ii) estimation of trajectory with unknown sensor parameters. In the second study the effect of correlation between the parameters on the estimated parameter value is studied. The observation model required for using Vehicle Dynamic Model as measurement was derived and was tested on simulated data.

# *Acknowledgments*

First and foremost, I would like to express my gratitude to my thesis advisor, Prof. Jan Skaloud, for giving me the opportunity to work on this project. Being someone who is extremely passionate about research and is committed to giving his students the best academic experience possible, he engaged me in new ideas and provided suggestions whenever I encountered hurdles in the project. I would like to take this opportunity to thank him again for it. Secondly, I would like to thank my project assistant, Davide Antonio Cucci. The tips and the suggestions he gave me during the thesis, not only helped me in solving various problems but also to improve myself as a person. I also would like to thank him for the patience he showed when clarifying some aspects of the project to me. I also would like to thank Mehran Khaghani, who took the time to introduce the topic of Vehicle Dynamic Models to me. I am also grateful for his help that he provided me during the thesis. I also use this opportunity to express my gratitude towards Prof. Dario Floreano who agreed to supervise my project from the Mechanical Engineering Department. I would also like to thank Dr. Ismael Colomina for agreeing to be the external expert for my defense.

The completion of this project and the results presented here were possible because of the constant support of my lab mates at TOPO. I would like to express my gratitude towards Iordan Doytchinov for the valuable feedback and insightful comments on my report. Philip Clausen, who is also doing his Ph.D. defense on the same day as mine, was a source for light-hearted discussions whenever you needed a break from the thesis. I would like to use this opportunity to thank him. I would like to give my kind regards to the other members of TOPO - Prof. Bertrand, Dr. Pierre-Yves, Emmanuel, Pinar, Gabriel, Pasquale. The friendly and helpful atmosphere created in the lab by them helped me in successfully completing my thesis. The other interns in the lab who shared the room with me - Laura, Janody, and Micheal, also deserves a special thanks.

Finally, I would like to acknowledge my friends and family for supporting me. I would like to thank my Mom, Dad, and my sister Adeline for providing encouragement and support to chase my dreams throughout my life. Throughout the project, my family provided constant support and encouraged me to tackle the problems I faced head-on. I also would like to thank Chris, Tiago, Matteusz, Kirtan, Sharbatanu, Akhilesh and Atri for making my time in Switzerland a lot more fun. I also would like to thank Vivek, Jophin, Dipin, Atul, and Navaneeth who has provided encouragement for my dreams from my bachelor days.

# Contents

# Chapter 1

# Introduction

## 1 Background

Determining the position and orientation of a robot is an essential requirement in many applications involving autonomous mobile robots [1]. For projects such as planetary rovers, self-driving cars, and unmanned aerial vehicles, knowing the location and orientation of robot accurately is essential. Broadly, the positioning methods could be classified into two - *relative* and *absolute* position measurements. The relative position measurements involve methods like dead-reckoning using wheel encoders and IMUs, while absolute position measurements involve GNSS, beacon navigation, landmark navigation, etc. Since relying on a single type of measurement is not ideal, the usual approaches involve fusing measurements from different sensors to get a position and orientation estimate with less uncertainty. These techniques to fuse data from different sensors are becoming more relevant in the case of Unmanned Aerial System (UAS) and particularly in Micro Aerial Vehicles (MAVs) that are increasingly being used as surveying equipment. With the advent of low cost, off-the-shelf systems, they are increasingly being used in monitoring the vegetation health, surveying, forest fire monitoring, etc [2]. The main challenge when designing a system such as a Micro Aerial Vehicle comes from its reduced size and the maximum weight it can carry. These limitations force the usage of small and light sensors which often provide measurements of insufficient quality [3] for applications such as mapping.

## 2 Available Solutions

The localization of Unmanned Aerial Vehicles is commonly done by fusing the measurements from satellite positioning systems and inertial sensors [4]. Relying solely on these sensors for localization often results in scenarios where the quality of the estimate is below the requirement or cases where the estimate of the position cannot be calculated at all. The most common of these scenarios is when GNSS outage occurs primarily due to the surrounding environment. Various approaches have been proposed to address this issue, some of them involve using dilution of precision [5] or by fusing measurements from additional sensors such as camera [6]. When the issue of GNSS outage is not addressed, the UAVs can quickly get lost in space posing a threat to living beings and objects around. Hence in most of the developed countries the operation of Micro Aerial Vehicles are restricted to Line of Sight (LOS) e.g.:-US [7], Canada [8], and EU [9]. Another approach to address this issue is by using more sophisticated error modeling techniques for IMUs. Such solutions using an improved stochastic error model for MEMS-based inertial sensors [10], by using wavelet analysis [11], or by using an autoregressive process for modeling inertial sensor errors [12] have been proposed. Even though such techniques yield an improved result compared to the conventional methods, GNSS outages for prolonged time periods causes most of the above methods to fail. Apart from improving the error models of the inertial sensors another approach to solving the issue is the addition of more sensors to the platform. Adding sensors like camera and Lidar [13], magnetometer and pitot tube [14], and dual laser scanners [15] yielded improved results. On the other hand, adding additional sensors to the platform reduced the payload and flight time of the MAVs. This limitation inspired researchers to look into other solutions such as using a Vehicle

Dynamic Model(VDM) as an additional measurement. The dynamics of any robot platform is controlled by the control inputs given to the motors. Using these control inputs one could predict the dynamics of motion from a given model which is specific to that platform. Hence the control inputs to the motors could be used as an additional measurement. The advantage of using a Vehicle Dynamic Model over other methods is that it requires no additional sensors. The input to the motors which is used for the measurements is already available in the auto-pilot system. This method could easily be applied to any other platform provided that a suitable Vehicle Dynamic Model is derived. Using VDM was already implemented in [16], [17], [18] and [19]. However, the above implementations used IMU as the process model and VDM as additional measurements. A new method was proposed in [20] where the VDM was used as the process model, this was further extended in [21] to include a practical realization of the method. The advantage of [20] and [21] is their ability to tolerate total IMU failure. This yielded a better result compared to the previous methods where IMU was used as a process model.

# 3 Proposed Solution

In the works [20] and [21], the fusion of the different measurements were done using an Extended Kalman Filtering (EKF). However, as demonstrated in the [22], graph-based methods are superior compared to the methods based on filtering. It was also shown in the same study that the graph-based based methods were superior compared to filtering methods in every aspect except in cases where the available processing budget was small. Although this was the case, the popularization of the graph-based methods was often impeded due to the complexity of computation associated with them. But with the advances in Sparse Linear Algebra [23] and the advent of cheap processing chips with higher computational powers, those problems have been addressed. One such graph-based method was developed in [24] is the Dynamic Network. In a Dynamic Network, residuals associated to measurements are minimized using a non-linear least squares approach. The aim of this study is to implement the Vehicle Dynamic Model (VDM) as a measurement in the sensor fusion in Dynamic Network in order to reduce the uncertainty in the trajectory estimate. The observation models that are used for adding these measurements depends on some parameters associated with that sensor. Determining the value of these parameters beforehand is difficult due to numerical complexity and observability issues. The project conducts two studies on the estimation of the trajectory. The first study will be with known sensor parameters while in the second study the parameters are known and need to be estimated.

# Chapter 2

# Dynamic Network

Most autonomous robots gather data about their current state through the various sensors on them. This data could be about their position and orientation with respect to a reference frame, physical quantities such as temperature, or information related to their immediate surroundings. Hence in situations where one need to perform functions such as localization and navigation, the correct interpretation of the data from the sensors is crucial. In such situations, relying on a single sensor more often than not produce unsatisfactory results. If only a single infrared sensor is used in a robot for obstacle detection, it won't be able to detect the obstacles on its sides and might result in the robot crashing into them. Likewise, a sensor like GPS operates at a lower frequency and hence won't be suitable for an application such as mapping. Hence relying on a single sensor often results in a limited spatial and temporal coverage In order to overcome these difficulties, the measurements from various sensors are combined to get an estimate with less uncertainty through a process called *sensor fusion*. By fusing data from different sensors we get extended spatial and temporal coverage, redundancy in the system which increases the reliability, and the increased confidence about the measurements.

Various methods exist in the literature to fuse measurements from different sensors. They could broadly be classified into two. The filter-based methods which include the Kalman Filtering and the graph-based methods such as Dynamic Networks. In a filtering based method such as Kalman Filtering, it is difficult to formulate constraints between states at different epochs. But in a dynamic network, such constraints could be modeled between states at different epochs. Also as shown in [22], the graph-based methods are superior to the filter based methods. In this study, a graph-based method called *Dynamic Network (DN)* will be used. In a *Dynamic Network*, the joint probability of sensor readings given the current estimate of the state variables, i.e., the robot poses and the sensor calibration parameters are represented by a factor graph [25]. Before getting into the structure of a Dynamic Network, it is imperative to familiarize the readers with some definitions. Those terms will be in the underlying sections followed by an overview of the *Dynamic Network* approach.

## 1 Definitions used in Dynamic Network

### 1.1 Reference Frames in Dynamic Network

This section will introduce the different reference frames that are relevant to this study and the notations used to represent them.

#### Body Frame

The body frame is a reference frame which is fixed to the body of the platform under consideration. In the current study, the axes are fixed as shown in Figure(2.1). The X-axis is in the forward direction of the drone, the Z-axis points upwards and the Y-axis points towards the left side of the drone forming a right-handed coordinate system. This will be denoted by the letter $b$

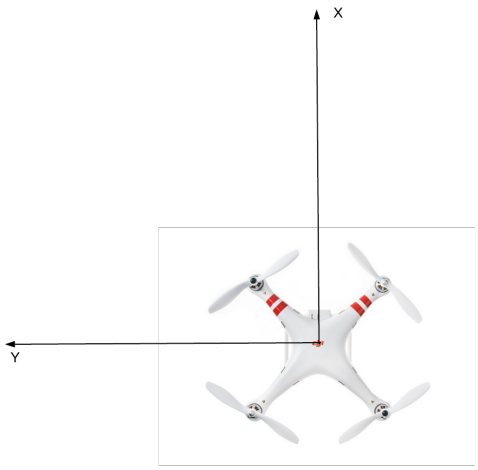



Figure 2.1: The Body Frame used in the project. The Z axis is defined by the right-hand coordinate system

#### World Frame

For this study, world frame is a reference frame that is fixed in an arbitrary point in space. For the project we used the simulated measurements generated using a *MATLAB* simulator. The rotation of the earth was ignored and acceleration due to gravity was taken to be a constant. The GPS coordinates, position, and orientation of the UAV are expressed in this frame of reference. For the GPS coordinates, the position of the quadcopter in world frame was sampled at regular intervals. This is also a right-handed coordinate system with Z axis pointing upwards. It will be denoted by the letter $W$

#### Sensor Frame

Sensor frame is a reference frame that is fixed to the particular sensor under consideration. It is denoted by the letter $s$

## 1.2 State Variables in Dynamic Network

The position and orientation of a robot are collectively referred to as pose. The state variables consist of the poses and the sensor calibration or geometric parameters. Each variable has a domain, which may be euclidean or non-euclidean depending on the type of the variable. In order to deal with non-euclidean domains, the operators $\boxplus$, $\boxminus$ are defined so that euclidean increments could be applied to the non-euclidean variable. This technique is called manifold encapsulation [26]. There are two types of state variables, fixed and not fixed. Fixed variables are state variables whose value is known and is not estimated. State variables are also classified on the basis of their time dependency. *Constant Variables* are variables whose value does not change with time or is constant in the time window considered. Whereas *time variant with limited bandwidth* variables vary with a specified frequency as a function of time. When solving the dynamic network, the unknown state variables are estimated. If $t_k$ is the time stamp of the measurements from the sensor that is used as the process model, then the unknowns that need to be estimated are

- Position and Orientation of the MAV at each $t_k$
- Parameters in the error models such as biases in the inertial sensors
- Parameters in the observation model that can be constant or varying with respect to time such as proportional motor constants.

In our study, the Cartesian coordinates are used to represent positions while unit quaternions are used to represent orientations. A bias was introduced to the observation model (explained in section (1.4)) of accelerometer and gyroscope. These random yet time-constant variables, $bf$ for accelerometer and $b_\omega$ for gyroscope, are unknowns that need to be estimated when solving the

dynamic network. More details about the bias and other error models could be found in section (1.2) in Chapter (3). Apart from these, there are different sensor calibration parameters that need to be estimated. Some of these parameters are the drag coefficients, components of the inertia matrix. These parameters will be listed in detail in section (3) in Chapter (3).

### 1.3 Backward Augmented State Estimator

Consider three robot poses $\Gamma_b^W(t_{i-2})$,$\Gamma_b^W(t_{i-i})$, $\Gamma_b^W(t_i)$ at time $t_{i-2} < t_{i-2} < t_i$. Let $s$ be a generic sensor on the robot with

$$\Gamma_s^b = \begin{bmatrix} s^b & R_s^b \end{bmatrix} \tag{2.1}$$

For each Sensor S, the augmented state is defined to be the following.

$$\hat{x}_s(t_i) = \begin{bmatrix} s_i^W & R_{s_i}^W & v_{s_i}^{s_i} & \omega_{Ws_i}^{s_i} & a_{s_i}^{s_i} & \alpha_{Ws_i}^{s_i} \end{bmatrix} \tag{2.2}$$

where $\hat{x}_s(t_i)$ is a function of $\Gamma_b^W(t_{i-2})$,$\Gamma_b^W(t_{i-2})$,$\Gamma_b^W(t_i)$, $\Gamma_s^b$ and of $\Delta t_{i-1}^{i-2} = t_{i-1} - t_{i-2}$ and $\Delta t_i^{i-1} = t_i - t_{i-1}$. Here the components of Equation (2.2) are the following.

- $s_i^W$ Position of the sensor at time $t_3$ with respect to the world frame.
- $R_{s_i}^W$ Orientation of the sensor frame $s$ at time $t_i$ expressed in world frame.
- $v_{s_i}^{s_i}$ Linear velocity of the sensor at time $t_i$ expressed in sensor frame S at time $t_i$
- $\omega_{Ws_i}^{s_i}$ Angular velocity of the sensor at time $t_i$ with respect to the world frame expressed in sensor frame at time $t_i$.
- $a_{s_i}^{s_i}$ Linear acceleration of the sensor at time $t_i$ expressed in sensor frame s at time $t_i$
- $\alpha_{Ws_i}^{s_i}$ Angular acceleration of the sensor at time $t_i$ with respect to the world frame expressed in sensor frame at time $t_i$.

The terms in the equation (2.2) are calculated using the estimates of the poses and sensor calibration parameters available at that point.

### 1.4 Observation Models in Dynamic Network

In this section the concepts of observation models which is used to relate the measured sensor values and the state variables are discussed. The observation model of a sensor is a mathematical relation that calculates the error between the measured value of a sensor and the predicted value of the sensor. An observation model should be specified for each type of sensor that needs to be used. The observation model has the following general representation. Let $s$ be a general sensor.

$$e(\breve{x}) = f(t; \hat{x}_S(t), z, \xi) + \eta \tag{2.3}$$

here $\hat{x}_S(t)$ is the augmented state that is defined in section (1.3), $\breve{x}$ is the vector of unknowns that needs to be estimated, $\xi$ is the sensor calibration parameters and $\eta$ is the Gaussian white noise. The function $f(t; \hat{x}_S(t), z, \xi)$ is the difference between the measured value from the sensor and the predicted value of that sensor. A mathematical model is used to predict the sensor value from certain inputs. These inputs are a subset of the terms in the backward augmented state estimator introduced in Section (1.3) of that particular sensor. such models were already derived for IMU and GNSS sensors. In this paper we derive the same for vehicle dynamic measurements.

## 2 Overview of Dynamic Network

### 2.1 Structure of a Dynamic Network

In a *Dynamic Network*, the unknown poses, sensor calibration parameters, and measurements are expressed in the form of a graph. In the graph, the unknown nodes are the robot poses and sensor calibration parameters. Each measurement is then connected by an edge to the state variables and parameters it is related to. These edges have a general form as shown in Equation (2.3). In

order to have more clarity, a small example is considered. $f(t; \hat{x}_S(t), z, \xi)$ for a gyroscope could be written as the following.

$$f(t; \hat{x}_S(t), z, \xi) = z_{\omega,j} - \omega^s_{Ws,j} - b_\omega \tag{2.4}$$

Here $z_{\omega,j}$ is the gyroscope measurement at time $t_j$, $\omega^s_{Ws,j}$ is the difference between the orientation of the sensor at time $t_j$ and $t_{j-1}$, and $b_\omega$ is the bias in the gyroscope. As seen from Equation (2.4), the observation model is written in terms of quantities in the sensor frame. But the unknown poses in the dynamic network are in the body frame of the robot. These quantities are related by the Equation (2.6).

$$Lever - arm \quad s^W = b^W + R^W_b s^b \tag{2.5}$$

$$Bore - sight \quad R^W_s = R^W_b R^b_s \tag{2.6}$$

Here $s^W$ and $R^W_s$ are the position and orientation of the sensor in world frame, $b^W$ and $R^W_b$ are the position and orientation of the robot in the world frame, and $s^b$ and $R^b_s$ are the position and orientation of the sensor in body frame. In this way, the edges are constructed for each measurement. Figure (2.2) shows how these edges are represented graphically for a general sensor $s$.

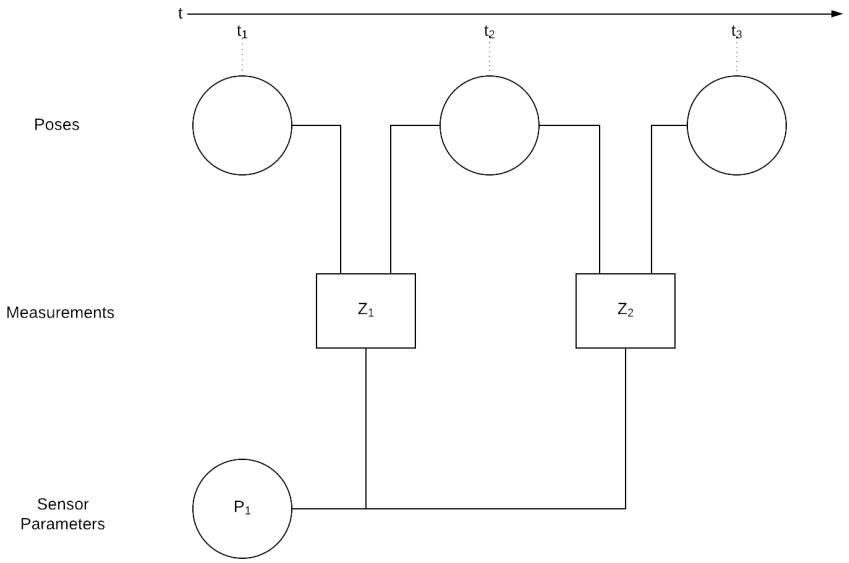



Figure 2.2: A small example of a factor graph

In Figure (2.2) there are three unknown poses at time $t_1$,$t_2$, and $t_3$. Two measurements $z_1$ and $z_2$ and a parameter $p_1$. Here the Equation (2.3) is applied for both measurements $z_1$ and $z_2$. In the first case when calculating $f(t; \hat{x}_S(t), z, \xi)$ in equation (2.3), the sensor calibration parameter $p_1$ and poses at $t_1$ and $t_2$ are used. In the second case, $p_1$ and poses at $t_2$ and $t_3$ are used. This process is continued for each available measurements.

In *Dynamic Network* the unknown quantities to be estimated are the poses and sensor parameters. But often the measurements from sensors like accelerometers and gyroscopes are related to a higher order of derivatives of the poses. For example, a gyroscope gives a reading which is the rate of change of orientation between two consecutive poses. These higher order derivatives are obtained by applying finite order differencing to the body poses. A finite difference operator $\Xi$ is introduced to address this. A finite difference can be a forward, backward, or central operator depending on the scheme implemented. A finite difference operator of the order $i$ requires upto $i$ samples before and after the element under consideration. We define

$$v^W_{s,j} = \Xi(s^W_j, s^W_{j\pm1}, ..., s^W_{j\pm O}) \tag{2.7}$$

$$\omega^W_{s,j} = \Xi(R^W_{s,j}, R^W_{s,j\pm1}, ..., R^W_{s,j\pm O}) \tag{2.8}$$

where $v^W_{s,j}$, $\omega^W_{s,j}$ are the velocity and angular velocity of the body at time $t_j$ in world frame. For the second order, we have

$$a^W_{s,j} = \Xi(v^W_{s,j}, v^W_{s,j\pm1}, ..., v^W_{s,j\pm O}) \tag{2.9}$$
$$\alpha^W_{s,j} = \Xi(\omega^W_{s,j}, \omega^W_{s,j\pm1}, ..., \omega^W_{s,j\pm O}) \tag{2.10}$$

where $a^W_{s,j}$, $\alpha^W_{s,j}$ are the acceleration and angular acceleration of the body at time $t_j$ in world frame. Here $\Xi$ is a finite differencing operator of the order O. For the study a finite backward differencing of the order 1 is used. For eg:-

$$v^W_{s,j} = \Xi(s^W_j, s^W_{j-1}) = \frac{s^W_j - s^W_{j-1}}{t(j) - t(j-1)} \tag{2.11}$$

This could be represented graphically as shown in Figure (2.3)

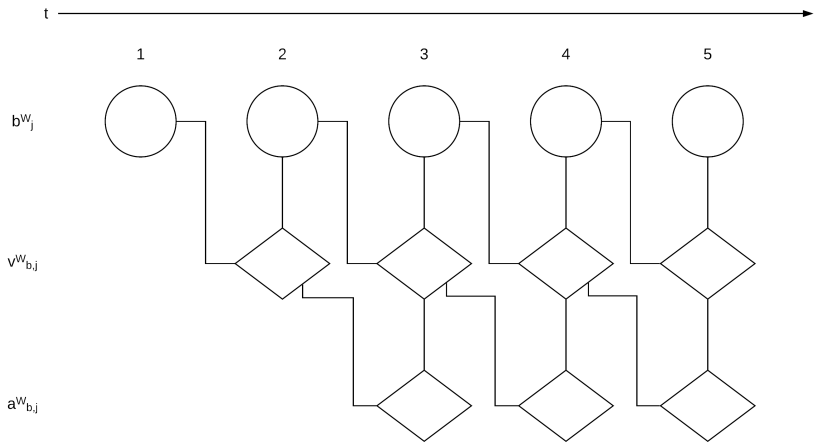



Figure 2.3: The Finite Differencing expressed graphically. Here $b^W_j$, $v^W_{b,j}$ and $a^W_{b,j}$ are respectively the pose, velocity, and acceleration of the body at time $t_j$ expressed in world frame.

## 2.2 Solving a Dynamic Network

Once the *Dynamic Network* is constructed, we have the unknown poses and sensor calibration parameters associated with each measurement through equation (2.3). The unknown quantities could be estimated by minimizing all the $e_i(t)$ through non-linear least square since most of the equations in the observation model are non-linear. The unknowns are listed in Section (1.2) of Chapter (2) and (3) of Chapter (3). Recalling the definition of $e_i(\breve{x})$, we have

$$e_i(\breve{x}) = f(t; \hat{x}_S(t), z, \xi) + \eta \tag{2.12}$$

here i $\in$ 1,2,...$K$, where $K$ is the total number of measurements. The objective is to find the unknowns such that the sum of the squares of all the $e_i(\breve{x})$ is minimum. This could be written as

$$\mathcal{P} : \underset{x}{arg\ min} \sum_{i=1}^{N} \bar{e}_i(\breve{x}_i)\Omega_{e_i}\bar{e}_i(\breve{x}_i) \tag{2.13}$$

here $\Omega_{e_i} = \Sigma^{-1}_{e_i}$ is the information matrix of the $i^{th}$ measurement. From the noise characteristics of a sensor, $\Sigma^{-1}_{e_i}$ could be determined using

$$\Sigma_{e_i} = J_{i,\eta}\Sigma_\eta J^T_{i,\eta}\big|_{x_i=\breve{x}_i,\eta=0} \tag{2.14}$$

Using Taylor series one can define

$$e_i(\breve{x} + \Delta\breve{x}) \approx e_i(\breve{x}) + J_{i,\Delta\breve{x}}\Delta\breve{x} \tag{2.15}$$

where $J_{i,\Delta x}$ is jacobian of $e_i$ with respect to the elements in $\Delta\breve{x}$. We define it as

$$J_{i,\Delta\breve{x}} = \begin{bmatrix} \frac{\partial e_i}{\partial \breve{x}_1} & \cdots & \frac{\partial e_i}{\partial \breve{x}_N} \end{bmatrix} \tag{2.16}$$

and

$$J_{\Delta\breve{x}} = \begin{bmatrix} \frac{\partial e_1}{\partial \Delta\breve{x}_1} & \cdots & \frac{\partial e_1}{\partial \Delta\breve{x}_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial e_J}{\partial \Delta\breve{x}_1} & \cdots & \frac{\partial e_J}{\partial \Delta\breve{x}_N} \end{bmatrix} \tag{2.17}$$

The elements of $\Delta\breve{x}$ are the increments corresponding to the elements in $\breve{x}$. We calculate the $\Delta\breve{x}$ by

$$\Delta\breve{x} = (J_{\Delta\breve{x}}^T \Omega J_{\Delta\breve{x}})^{-1} J_{\Delta\breve{x}}^T \Omega \Delta y \tag{2.18}$$

here $\Delta y$ is the difference between the measured value and the estimated measurement from the current estimates of the unknowns. Then we apply

$$\breve{x} = \breve{x} + \Delta\breve{x} \tag{2.19}$$

This is continued until a termination criterion is reached.

## 2.3 Advantages of a Dynamic Network

There are a couple of advantages by using the factor graphs, some of them are listed below

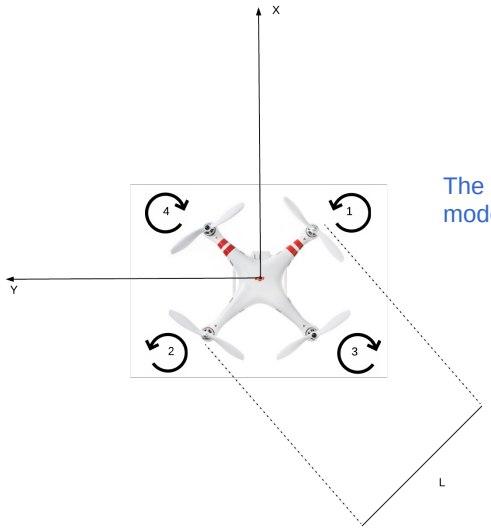
- An arbitrary number of sensors could be handled. Any type of sensor could be modeled and inserted into the graph appropriately. This makes it easier to insert a new sensor like VDM into the framework.
- Sensors could be turned on or off easily
- The network can handle out of the order measurements. This helps to address the lag in the control if it arises. Also, different measurements are processed simultaneously.
- The graph methods were shown to be more accurate and in some cases faster than traditional filter-based methods [22].
- No information is lost since marginalization of the variables is not done.
- Constraints between multiple poses at different time instants could be expressed.

# Chapter 3

# Vehicle Dynamic Model

This chapter introduces the observation model for using the control input given to the motors of quadcopter through its vehicle dynamic model. As introduced in section (1.4), the observation models relate the measured sensor values and the unknown state variables. In order to add a new type of measurement to the dynamic network, it is necessary to develop an observation model for it.

## 1 Model of the UAV




The motors 3 and 4 are interchanged in the current model

Figure 3.1: The Quacopter model used for this project with the motor configurations

As shown in Figure (3.1), the $x$- axis of the drone will be pointed in the forward direction between the two arms, the $y$-axis will pointing towards the left side when viewed from the top and the $z$-axis will be pointing in the upward direction. This type of configuration is known as x-configuration in the literature. The motors of the UAV are numbered in the manner shown in Figure (3.1). The motors 1 and 2 rotate counterclockwise and motors 3 and 4 rotate clockwise when viewed from the top.

## 2 Derivation of Dynamic Equations for a UAV

In order to derive the dynamic equations of a UAV, we need to start with the motors attached to it. In a UAV, brushless motors are usually used. All the four motors in the UAV are identical and

hence will analyze a single motor. The torque produced in a motor is given by

$$\tau = K_t I \tag{3.1}$$

Here $I$ is the current through the motor, $\tau$ is the motor torque, and $K_t$ is a proportional constant. The voltage across the motor is given by

$$V = IR_m + K_v\omega \tag{3.2}$$

Here $R_m$ is the motor resistance, $K_v$ is the proportional constant, and $\omega$ is the angular velocity. Hence using equations (3.1 ) and (3.2), the power consumed by the motor is defined as

$$P = IV = \frac{K_v}{K_t}\tau\omega \tag{3.3}$$

## 2.1 Thrust Force from the motor

Let us consider a single motor, i ∈ 1,2,3,4. From the conservation of energy we have

$$P_i.dt = F_{T,i}.dx \tag{3.4}$$

from this we get

$$P_i = F_{T,i}\frac{dx}{dt} = F_{T,i}v_h \tag{3.5}$$

here $v_h$ is the air velocity during hovering. From momentum theory for propellers, $v_h$ can be written as

$$v_h = \sqrt{\frac{F_{T,i}}{2\rho A}} \tag{3.6}$$

here $\rho$ is the density of the air and $A = \pi R^2$ is the area swept by a propeller with radius $R$. From [27] the torque is proportional to the thrust generated with a proportional constant $K_\tau$ that depends on the blade geometry and hence can be written as

$$\tau_i = K_\tau F_{T,i} \tag{3.7}$$

Using equations (3.3) and substituting (3.5),(3.6), and (3.7) we get,

$$P_i = \frac{K_v K_\tau}{K_t}F_{T,i}\omega = \frac{F_{T,i}^{3/2}}{2\rho A} \tag{3.8}$$

solving for $F_{T,i}$, we get

$$F_{T,i} = \left(\frac{K_v K_\tau\sqrt{2\rho A}}{K_t}\omega_i\right)^2 = b\omega_i^2 \tag{3.9}$$

Here $b$ is proportional constant. Hence the thrust force on the UAV is

$$F_T = \begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^{N} b\omega_i^2 \end{bmatrix} \tag{3.10}$$

## 2.2 Drag forces

A UAV experiences drag forces when it moves through the air. This is modeled as a force proportional to the square of air speed and it acts in the opposite direction to the motion. A proportional constant $C_x$, $C_y$, and $C_z$ for each axis are defined. Hence the drag force is defined as

$$F_D^b = -\begin{bmatrix} C_x(V_x^b)^2 Sgn(V_x) \\ C_y(V_y^b)^2 Sgn(V_y) \\ C_z(V_z^b)^2 Sgn(V_z) \end{bmatrix} \tag{3.11}$$

here the $V = \left[V_x^b; V_y^b; V_z^b\right]$ is the air speed. It is defined as

$$V^b = V_d^b - V_w^b \tag{3.12}$$

here $V_d^b$ is the velocity of drone and $V_w^b$ is the velocity of the wind.

## 2.3 Gravity

The UAV also experiences the gravity by the earth. For the study, the gravity is modeled as a constant with a value $g = 9.8m/s$. Hence

$$F_G^b = R_W^b \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \tag{3.13}$$

This model could be extended to include a gravity model.

## 2.4 Torque contributed by the motor propeller

The torque experienced in a UAV is contributed by each motor. When the propellers are spinning, they experience a drag force on them. This drag force is given by

$$F_D = \frac{1}{2}\rho C_D A v^2 \tag{3.14}$$

here $\rho$ is the density of the surrounding air, $C_D$ is a proportional constant, and $A$ is cross-sectional area of the propeller. For the sake of simplicity, let us assume that the drag force acts on the tip of the propeller. Then the torque due to the drag can be written as

$$\tau_D = R \times F_D = \frac{1}{2}R\rho C_D A v^2 = \frac{1}{2}R\rho C_D A(R\omega)^2 = d\omega^2 \tag{3.15}$$

here R is the radius of the propeller. Even though we made an assumption that the force is acting on the tip of the propeller, the end result is that the torque is proportional to the angular velocity of the motor. For propellers rotating in the clockwise direction when viewed from the top, this torque is positive and it is negative for propellers rotating in the anti-clockwise direction.

## 2.5 Vehicle Dynamic Model - Combined Forces and Moments

In this section, the Vehicle Dynamic Equation which will be used to give the inputs to the motors of the UAV as a measurement in the dynamic network will be derived.
The total force acting on the UAV in its body frame is sum of thrust ($F_T$), drag force ($F_D$) and gravity ($F_G$).

$$F = F_T + F_D + F_G^b \tag{3.16}$$

From Equations (3.11), (3.13), and (3.10) and dividing both sides of the equation with the mass of UAV $m$, the acceleration of the center of mass of UAV, $a_b$ can be determined as

$$a_b = \frac{1}{m}\begin{bmatrix} 0 \\ 0 \\ \sum_{i=1}^{N} b\omega_i^2 \end{bmatrix} - \frac{1}{m}\begin{bmatrix} C_x(V_x^b)^2 Sgn(V_x) \\ C_y(V_y^b)^2 Sgn(V_y) \\ C_z(V_z^b)^2 Sgn(V_z) \end{bmatrix} + R_W^b \begin{bmatrix} 0 \\ 0 \\ -g \end{bmatrix} \tag{3.17}$$

From Figure (3.1), the torque acting on each axis can be calculated, it can be written as

$$M_b = \begin{bmatrix} \frac{\sqrt{2}Lb}{4}(-\omega_1^2 + \omega_2^2 + \omega_3^2 - \omega_4^2 \\ \frac{\sqrt{2}Lb}{4}(-\omega_1^2 + \omega_2^2 - \omega_3^2 + \omega_4^2 \\ d(-\omega_1^2 - \omega_2^2 + \omega_3^2 + \omega_4^2 \end{bmatrix} \tag{3.18}$$

here L is the distance between the propellers turning in the same direction as shown in Figure (3.1), $b$ is the proportional constant for the thrust of the motor, and $d$ is the proportional constant for the torque of each motor. The rotational equations of motion are derived using the Eulers equations of rigid body dynamics. We have

$$I\dot{\omega} + \omega \times (I\omega) = M_b \tag{3.19}$$

Solving for $\dot{\omega}$, we have

$$\dot{\omega}^b = I^{-1}(M_b - \omega \times (I\omega)) \tag{3.20}$$

here $I$ is the moments of inertia matrix of the form

$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{xy} & I_{yy} & I_{yz} \\ I_{xz} & I_{yz} & I_{zz} \end{bmatrix} \tag{3.21}$$

The $f(t; \hat{x}_s(t), z, \xi)$ in Equation (2.3) could now be derived for using Vehicle Dynamic model as an additional sensor. Using Equations (3.17) and (3.20)

$$f(t; \hat{x}_S(t), z, \xi) = \begin{bmatrix} a^b - \hat{a}(X_s) \\ \dot{\omega}^b - \hat{\dot{\omega}}^b(X_s) \end{bmatrix} \tag{3.22}$$

Here $\hat{a}(X_s)$ and $\hat{\dot{\omega}}^b(X_s)$ are the predicted linear and angular acceleration using the current knowledge about the backward augmented state of the sensor. The inputs to the observation model given by equation (3.22) are the angular velocities $\omega_1$, $\omega_2$, $\omega_3$, and $\omega_4$ as seen in Equations (3.17) and (3.18).

# 3 Parameters in the model

In the previous section, the vehicle dynamic model was derived. In this model, there are parameters that are inserted as sensor parameters of the VDM in the dynamic network. These parameters are sometimes known in advance or need to be estimated online. The section will introduce the parameters used in this model.

## 3.1 Wind Parameters

Three parameters are used to model the wind, one for each axis. It is denoted by $w_1$, $w_2$, and $w_3$. Their unit is m/s. For this project the parameters of wind was fixed to be zero.

## 3.2 Drag Parameters

The drag parameters are the proportional constants in Equation (3.11). Three parameters are used to model the proportional constant of the drag force in each axis. It is denoted by $C_x$, $C_y$, and $C_z$. The unit is $N.s^2/m^2$

## 3.3 Copter Parameters

The copter parameters consist of the mass and length of the drone. The length of the drone is measured as indicated in Figure (3.1). The length is denoted by $L$ and mass by $m$. The units are $m$ and $kg$ respectively.

## 3.4 Motor Parameters

The motor parameters consist of the proportional constants that we derived for the thrust and torque produced by the body. They are denoted by $b$ and $d$ respectively. The unit of $b$ is $N.s^2$ and $d$ is $N.m.s^2$.

## 3.5 Inertia Matrix Parameters

The inertia matrix parameters consist of the terms from Equation (3.21). They could be classified into two, diagonal elements and off-diagonal elements. The diagonal elements are indicated by $I_{xx}$, $I_{yy}$, and $I_{zz}$ and the off-diagonal elements by $I_{xy}$, $I_{xz}$, and $I_{yz}$. Their unit is $kg.m^2$.

# Chapter 4

# Experimental Method

The chapter will explain the method of study adopted for this project. All the experiments in this study were conducted using simulated data. The chapter is divided into three sections. In the first section, an overview of the different sensors used in the UAV is provided together with their stochastic parameter. This is followed by a section which explains how the measurements were generated and the experimental procedure used. The final section provides an explanation about the tool that was used to implement the dynamic network.

## 1 Experimental Setup

### 1.1 Sensors Simulated on UAV

The following sensors were simulated on the UAV

#### GPS Sensor

The GPS measurements were simulated with the help of the *MATLAB* simulator. The trajectory of the quadcopter was obtained from the simulator and the coordinates of the quadcopter in the world frame was sampled at specific intervals to simulate the GPS measurements. Te sampling interval was fixed to be 1s.

#### IMU Sensor

An IMU sensor was also simulated. The IMU is composed of three accelerometers and three gyroscopes, one for each axis. The sampling interval of the IMU sensors was fixed to be 0.01s.

#### VDM sensor

The inputs that were given to the motors during the flight was used as an additional measurement in the trajectory estimation of the drone. No noise was given to the measurements for the VDM sensor.

### 1.2 Noise Models used for Experiments

In the real world, the measurements from the sensors would contain noise from different sources. Hence when simulated data is used for testing a system, it becomes imperative to account for these noises. In the study we use, two types of noise models to emulate the real data

#### White Noise

The measurements from the sensors are sometimes perturbed by noise. These noises fluctuate at a higher rate than the sampling rate. The noise could be modeled as a zero-mean Gaussian variable with a variance $\sigma^2$. This noise was used for measurements from GPS and IMU. The variance $\sigma^2$ used for the sensors will be specified in Chapters (5) and (6).

### Bias

A bias is a constant offset in the measured value. This type of error is commonly found in accelerometers and gyroscopes. This error is modeled as a random constant value. For the study, the random constant value was generated for each experiment which was distributed with a variance $\sigma^2$.

# 2 Experimental Approach

## 2.1 Generating the Measurements

The data for the experiments in this study was generated using the *MATLAB* simulator used in [20]. The simulator uses a simple PID controller to take the UAV through the waypoints that the user specified. One of the key challenges in the parameter estimation is the observability of the parameters. The observability of a system is defined as the possibility to compute the initial state of the system given a sequence of inputs and measurement. Hence the trajectory chosen for analysis should be such that the states in the system are observable. A trajectory that satisfies this condition is the lemniscate trajectory. Hence it was chosen for this study. The following equation is the parametric form of lemniscate trajectory

$$x = \frac{a\cos(t)}{1+\sin^2(t)} \tag{4.1}$$

$$y = \frac{a\cos(t)\sin(t)}{1+\sin^2(t)} \tag{4.2}$$

The trajectory used had a span of 80m in $x$-direction and 60m in $y$-direction. The time of flight of the drone was 400s. Using the simulator the above trajectory was specified and the waypoints were sampled at equal distances. The trajectory used and waypoints generated could be seen in Figure (4.1).

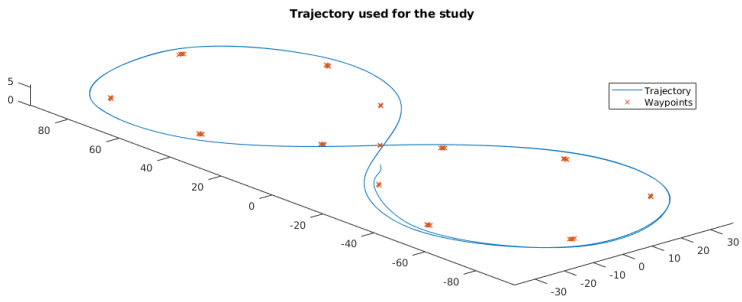



Figure 4.1: The Trajectory used for the simulation

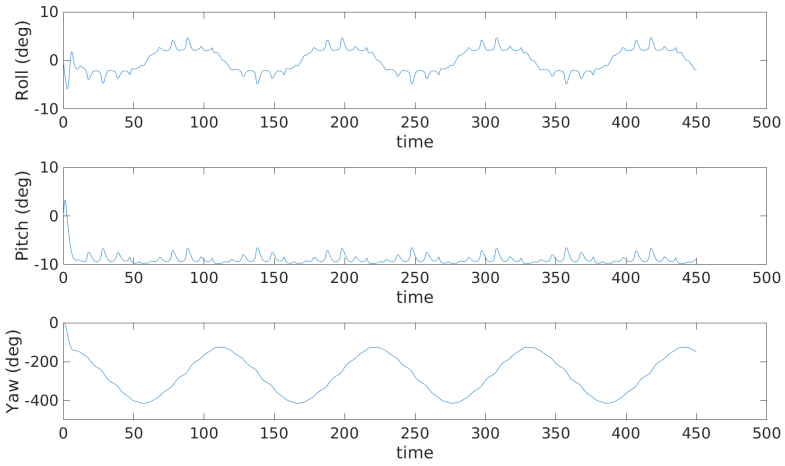



Figure 4.2: The Roll, Pitch, and Yaw of the Trajectory used

The above trajectory was used to generate the measurements for the experiment. The IMU measurements and VDM measurements had a frequency of 100Hz while the GPS measurements had a frequency of 1Hz. The inputs to the motors for the VDM sensor was obtained from the controller.

## 2.2 Monte Carlo Simulation

Monte Carlo simulations are widely used in the field of engineering to model the outcome of a system that depends on the random variable. They rely on repeated random sampling to obtain numerical results. Monte Carlo experimentation is the use of simulated random numbers to estimate some functional of a probability distribution [28]. Several realization of the experiment is carried out and for each iteration, a set of random variables generated with the help of a probability distribution is used as input. From the output of these realizations, the characteristics of the solution could be inferred.

The data generated from the *MATLAB* simulator (section (2.1 )) was for the ideal case and hence the measurements didn't have noises. In order to perform a Monte Carlo Simulation, noise vectors were generated using the noise models given in section (1.2). These noise vectors were added to the ideal measurements to produce the measurements with noise. 100 such realizations were generated for the Monte Carlo simulations. The weight was given through the information matrix $\Omega = \Sigma^{-1}$. The information matrix is diagonal and its entries are taken as the inverse of the variances associated with them. Since the knowledge about the variances associated with the VDM sensor is not available, the experiments have to be done against different variances and the effect on the output studied. 6 different cases were studied and the diagonal elements of $\Sigma$ for each case are specified below. The first three elements are the variance of VDM measurements for linear acceleration and the remaining are the same for the angular acceleration.

- Case 1 : $\Sigma_1^D = \left[1.69e^{-6}\mathrm{m}^2/\mathrm{s}^4, 1.69e^{-6}\mathrm{m}^2/\mathrm{s}^4, 5.04e^{-5}\mathrm{m}^2/\mathrm{s}^4, 4.35e^{-4}\mathrm{rad}^2/\mathrm{s}^4, 4.35e^{-4}\mathrm{rad}^2/\mathrm{s}^4, 4.35e^{-4}\mathrm{rad}^2/\mathrm{s}^4\right]$
- Case 2 : $\Sigma_2^D = 10 \times \Sigma_1^D$
- Case 3 : $\Sigma_3^D = 50 \times \Sigma_1^D$
- Case 4 : $\Sigma_4^D = 100 \times \Sigma_1^D$
- Case 5 : $\Sigma_5^D = 150 \times \Sigma_1^D$
- Case Disable : VDM measurements are not used

The Monte Carlo simulations were done for each case.

# 3 Experimental Tool

A tool was needed to implement the dynamic network and to estimate the trajectory from the measurements generated from the Monte Carlo simulations. For this purpose, an open source sensor fusion tool called *ROAMFREE* [29] was selected. The *ROAMFREE* was developed in [30] and is used to estimate the pose and do self sensor calibration. The framework to use the measurements from GPS and IMU was already been implemented in *ROAMFREE* [31] . We extended this open source tool to include an additional measurement type which could be given by the VDM measurements.

# Chapter 5

# Trajectory Estimation with fixed VDM Coefficients

In this chapter, the results obtained from the trajectory estimation using VDM for two different IMU is presented. This is followed by a study on the effect of different IMUs on the final trajectory estimate. The values of the parameters were known before-hand. Here the wind velocity in all three different axes was set to zero. The VDM parameters had the following values

- $Drag\ Parameters = \left[d_1, d_2, d_3\right] = \left[0.1\mathrm{N\,s^2/m^2}, 0.1\mathrm{N\,s^2/m^2}, 0.2\mathrm{N\,s^2/m^2}\right]$
- $Copter\ Parameters = \left[l, m\right] = \left[0.4\mathrm{m}, 1.454\mathrm{kg}\right]$
- $Motor\ Parameters = \left[b, d\right] = \left[1.6e^{-4}\mathrm{N\,s^2}, 7.5e^{-7}\mathrm{N\,m\,s^2}\right]$
- $Inertia\ Matrix\ Diagonal = \left[I_{xx}, I_{yy}, I_{zz}\right] = \left[5.8e^{-3}\mathrm{kg\,m^2}, 6.0e^{-3}\mathrm{kg\,m^2}, 1.1e^{-2}\mathrm{kg\,m^2}\right]$
- $Intertia\ Matrix\ Off-Diagonal = \left[I_{xy}, I_{xz}, I_{yz}\right] = \left[0\mathrm{kg\,m^2}, 0\mathrm{kg\,m^2}, 0\mathrm{kg\,m^2}\right]$

Using these parameters and using the trajectory shown in Figure (4.1)), the measurements for the experiment was generated as described in Section (2.1) of Chapter (4). After adding noise to these measurements, a Monte Carlo simulation was performed to analyze the distribution of errors in the trajectory estimate. For this study, separate experiments were conducted using two different IMU's. In the following sections, the results from these experiments are shown followed by a comparison between them. For each experiment performed in this study, the Monte-Carlo simulations were done for six different cases. The cases are specified in Section (2.2) of Chapter (4). The different cases will be denoted by the numbers 1,2,3,4,5 and *Disable* respectively in the subsequent plots.

## 1 Distribution of Error in the Trajectory Estimate using IMU 1

The first IMU that was used, had the following noise properties.

### Accelerometer

- White noise $\sigma = 4.02e^{-3}\ m/s^2/\sqrt{Hz}$
- Random Bias $\sigma = 7.84e^{-2}\ m/s^2$

### Gyroscope

- White noise $\sigma = 4.7e^{-5}\ rad/s/\sqrt{Hz}$
- Random Bias $\sigma = 3.5e^{-3}\ rad/s$

Using the above noise properties, the noise was generated using the noise models in Section (1.2) in Chapter (4). A white noise error of $\sigma = 1m$ was added to the GPS measurements. 100 sets of noise were generated and were added to the ideal measurements from the *MATLAB* simulator to obtain 100 sets of measurements for the Monte Carlo simulation. The Figure (5.1) and (5.2) shows the results obtained from a single test.

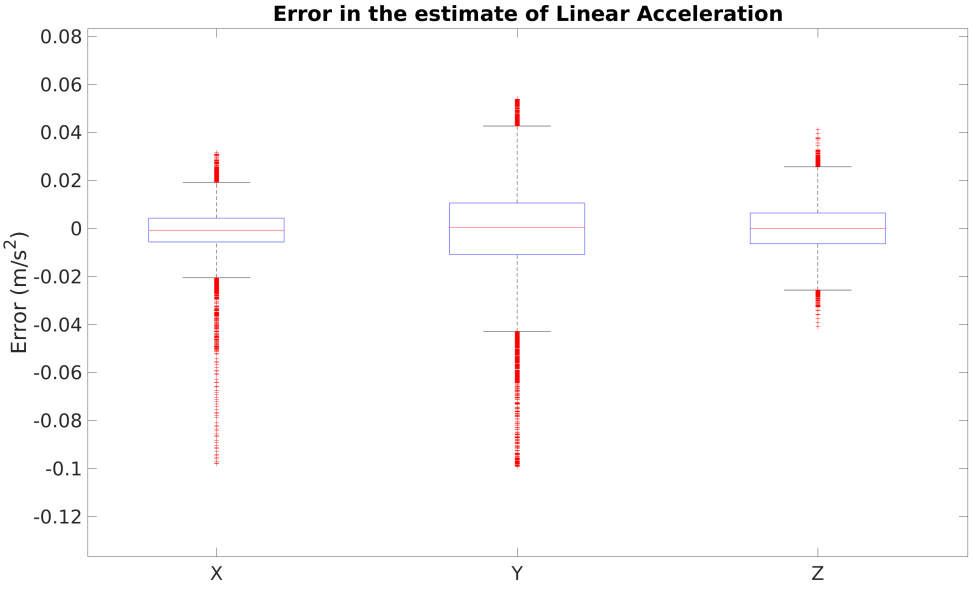



Figure 5.1: Error in the linear acceleration of the estimate

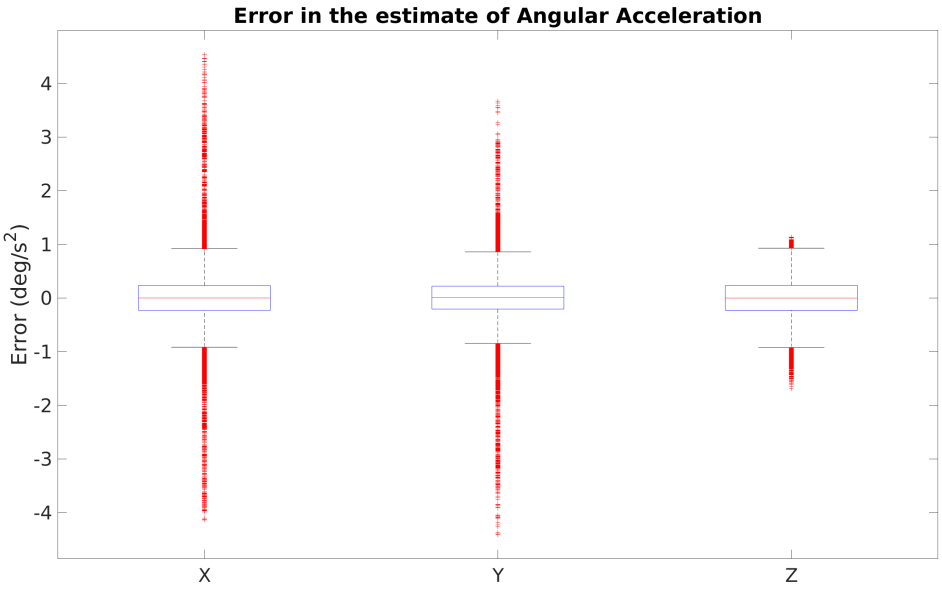



Figure 5.2: Error in the angular acceleration of the estimate

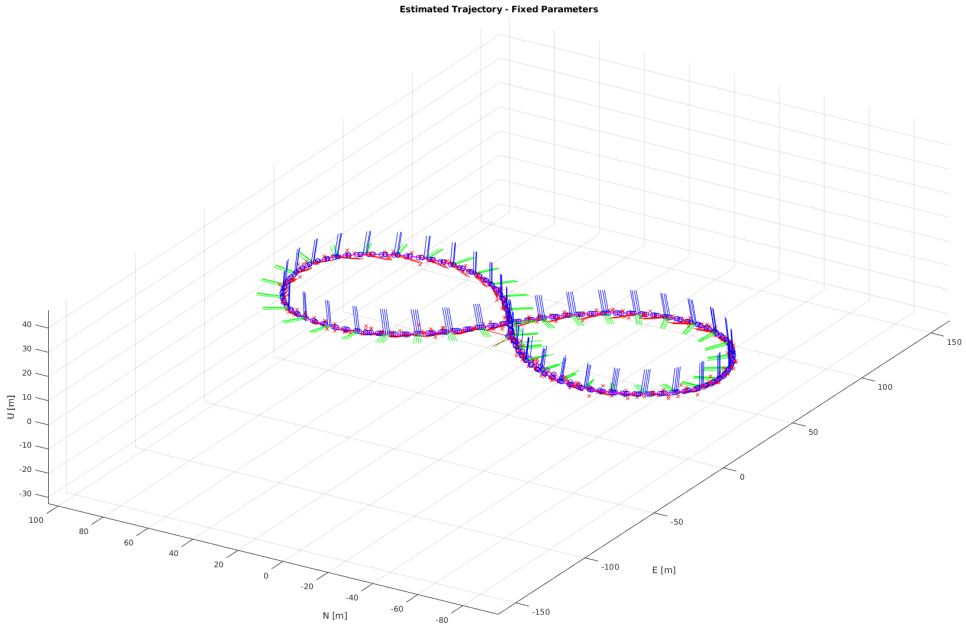
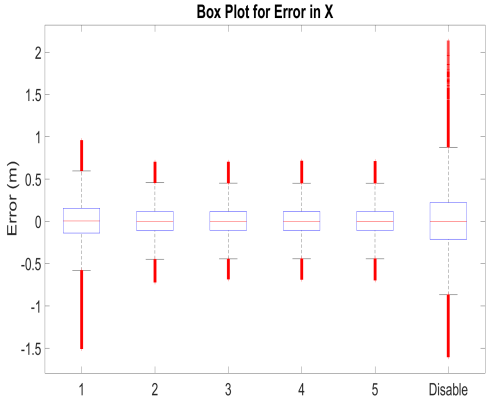
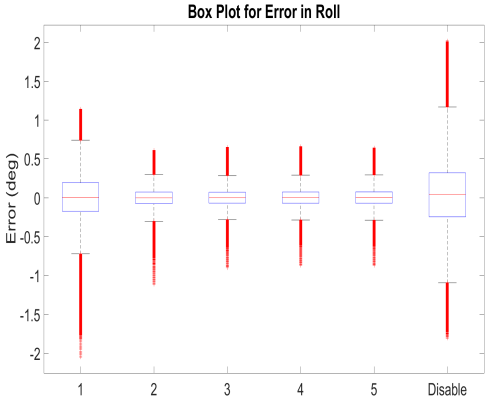



Figure 5.3: The Trajectory Estimated using Dynamic Network

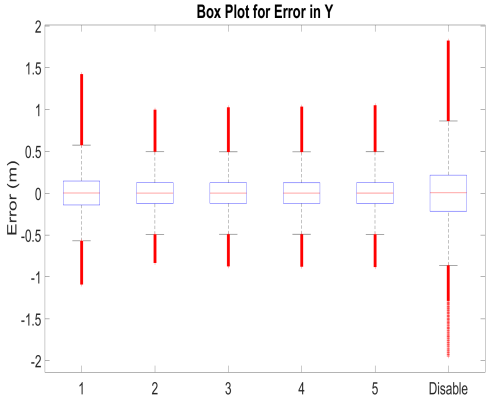
The reader can see from Figure(5.1) and (5.2), error in the estimated linear and angular acceleration is very low. Similar tests were done on the remaining set of measurements to study the distribution of the error in the final estimate. The results from the tests are summarized in Figure (5.4).



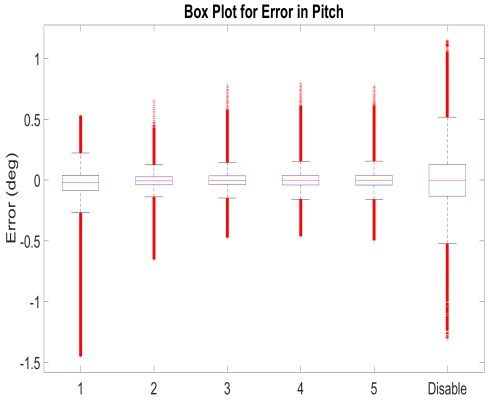

(a) Error in position X



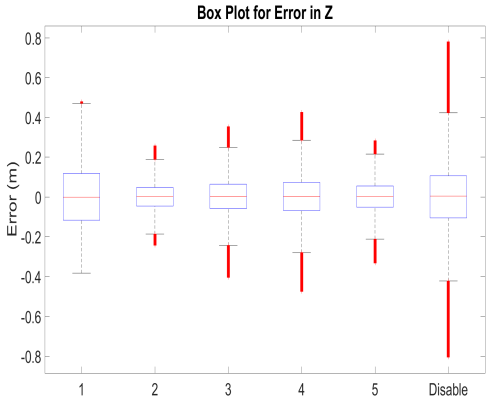

(b) Error in Orientation Roll



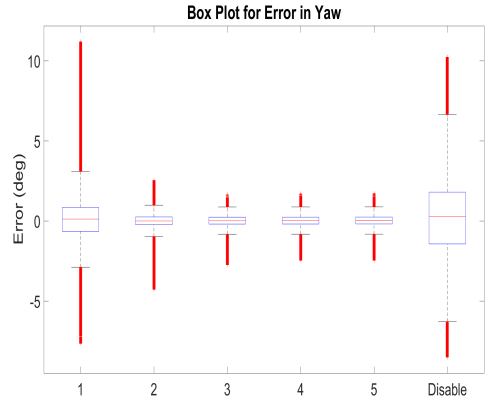

(c) Error in position Y




(d) Error in Orientation Pitch




(e) Error in position Z



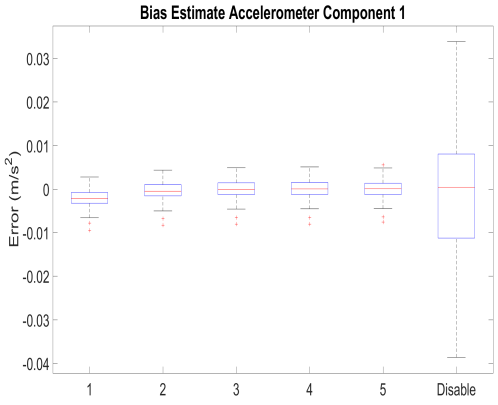

(f) Error in Orientation Yaw

Figure 5.4: Box Plots of Errors in Position and Orientation with IMU 1

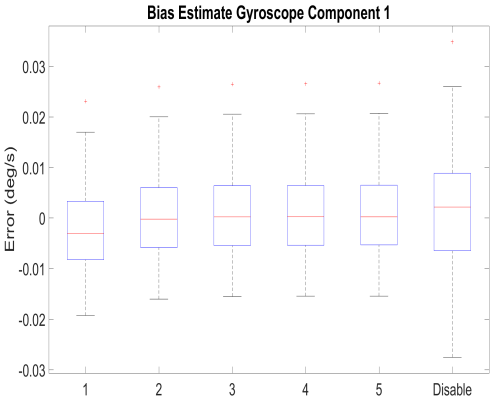
From the Figure (5.4), certain trends can be observed. Variance given to the VDM measurements plays a role in the errors in the final estimate of the trajectory. The general trend we can see in this figure is that there is an optimal variance for the VDM measurement. The error distribution at either end of the graph has a higher variance compared to the rest. When the VDM measurements are not used or are given a high variance, the resulting estimated trajectory has an error distribution of higher variance. Here the number of sources of measurement decreases resulting in a higher uncertainty of the final estimate. When the variance of the VDM measurements is low, the dynamic network places higher confidence on these measurements. This might result in neglection of other measurement sources if they high variance with respect to the variance of the measurement. This results in a higher uncertainty in the estimate.

## 1.1 Bias Estimation

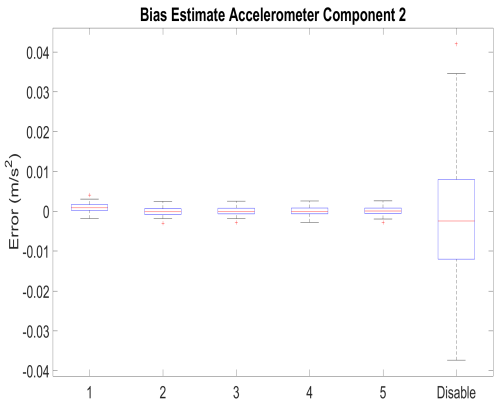
When the noisy measurements were generated, a bias was added to the measurements from accelerometer and gyroscope. The Bias that was added was stored and was later compared with the estimated bias. The results could be seen in Figure (5.5).



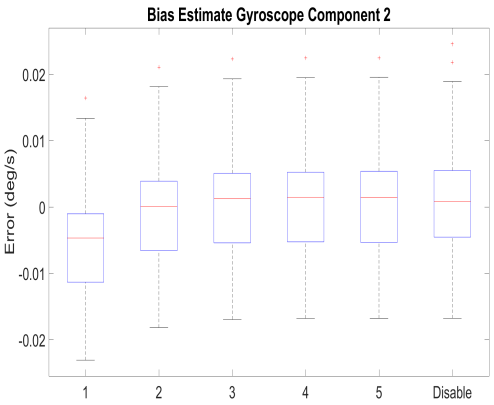

(a) Error in Bias Estimate Accelerometer X



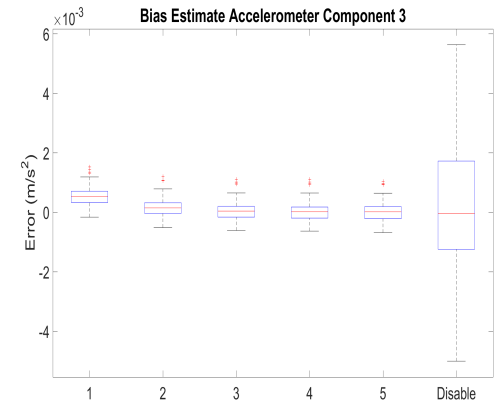

(b) Error in Bias Estimate Gyroscope X



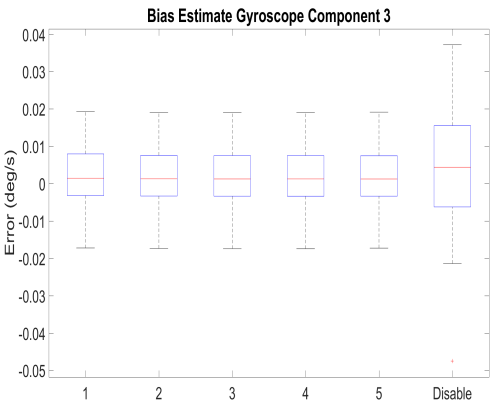

(c) Error in Bias Estimate Accelerometer Y




(d) Error in Bias Estimate Gyroscope Y




(e) Error in Bias Estimate Accelerometer Z




(f) Error in Bias Estimate Gyroscope Z

Figure 5.5: Box Plots of Errors in Estimated Bias of accelerometer and gyroscope with IMU 1

From Figure (5.5), one can see that the bias in the Accelerometer measurements is accurately determined when VDM measurements are used. As mentioned earlier with the VDM, an additional measurement is being provided to the dynamic network which decreases the variance in the estimate. Another trend that could be observed is the effect of the VDM measurements with low variance on the variance of the error in estimated bias. In such cases, the mean of the error is

shifted from zero. This happens because in those cases, the measurements from IMUs are given less weight making the bias estimation difficult. In the case of bias estimation of gyroscope component Y, there isn't a noticeable change in the variance of the estimate with the addition of VDM measurements. This may be due to the observability issues arising from the trajectory that we are using.

# 2 Distribution of Error in the Trajectory Estimate using IMU 2

The next experiment was done using an IMU of higher quality. The aim was to see whether the accuracy of the IMU would have any effect on the variance used for VDM measurements. The IMU had the following properties

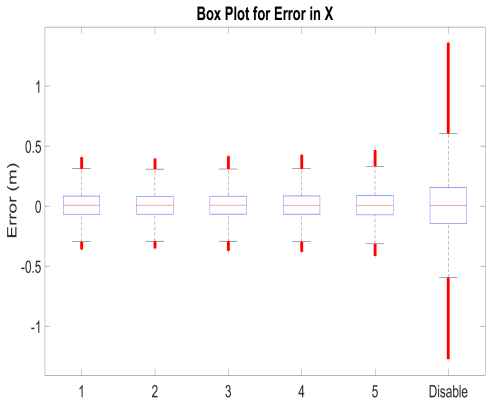
## Accelerometer

- White noise $\sigma = 4.02e^{-4} m/s^2/\sqrt{Hz}$
- Bias $\sigma = 7.84e^{-2} m/s^2$

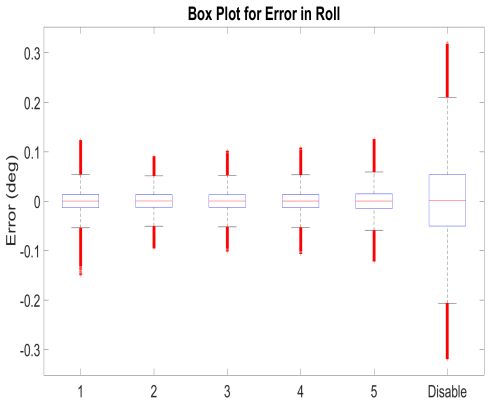
## Gyroscope

- White noise $\sigma = 4.7e^{-6} rad/s/\sqrt{Hz}$
- Bias $\sigma = 3.5e^{-3} rad/s$

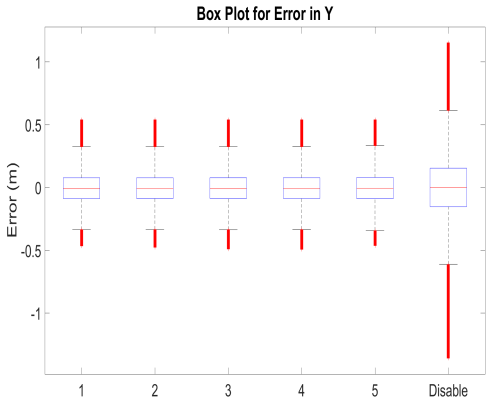
Like in the previous experiment, noisy measurements were generated for a Monte Carlo simulation with 100 iterations. The results of this experiment could be seen in Figure (5.6).



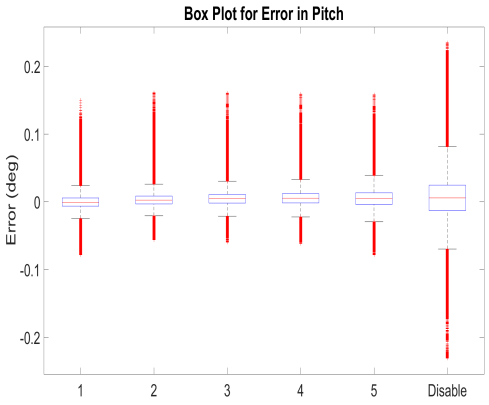

(a) Error in position X



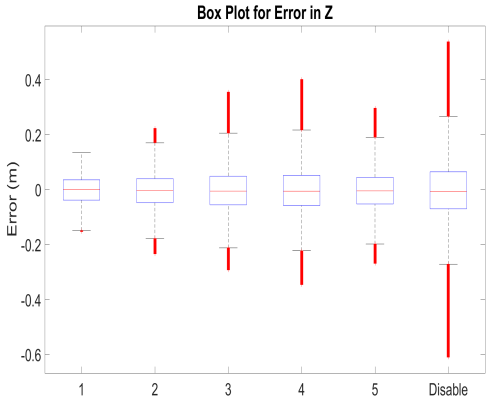

(b) Error in Orientation Roll




(c) Error in position Y




(d) Error in Orientation Pitch




(e) Error in position Z

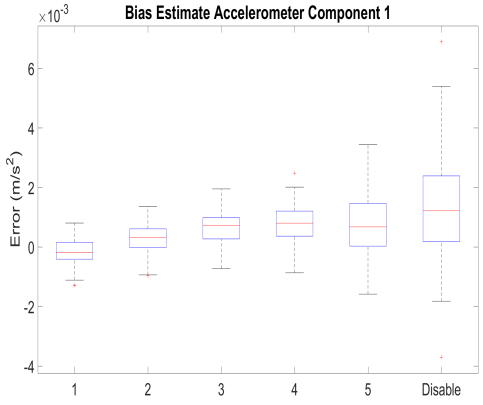

(f) Error in Orientation Yaw

Figure 5.6: Box Plots of Errors in Position and Orientation with IMU 2

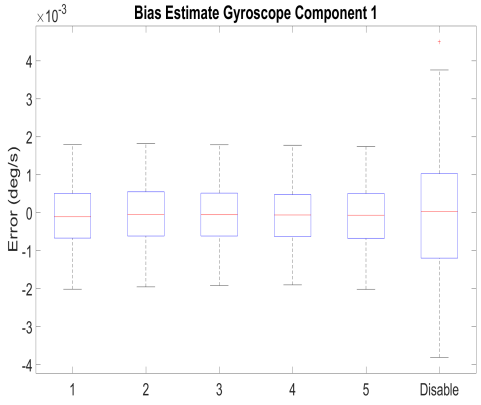
Like in the previous experiment, the variance in the error of the estimate decreased with the addition of VDM measurements. Unlike the previous case where using a low variance for the VDM measurement gave a bad performance, here all the variances that were tested gave comparable performances. In the previous case with the low variance for VDM measurements, the measurements from other sensors were ignored or given less weight. Here we are using an IMU with less noise and hence its measurements have lower variance. Hence even with a low variance for VDM measurements, the IMU measurements are still used for sensor fusion and hence a performance drop is not seen.

## 2.1 Bias Estimation

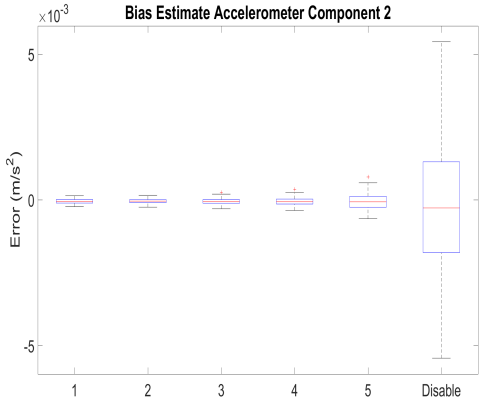
Like in the previous experiment, a bias was added when the noisy measurements were generated. This Bias was estimated by solving the dynamic network and the error in the estimate was computed. The results are shown in Figure (5.7).



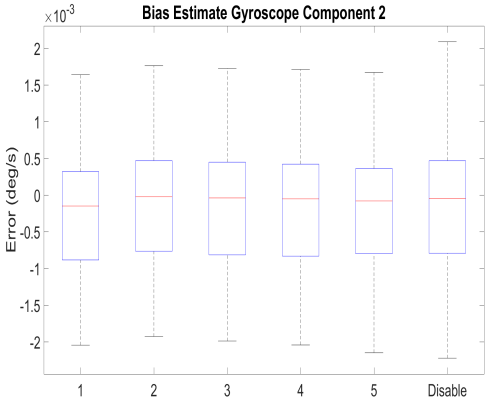

(a) Error in Bias Estimate Accelerometer X



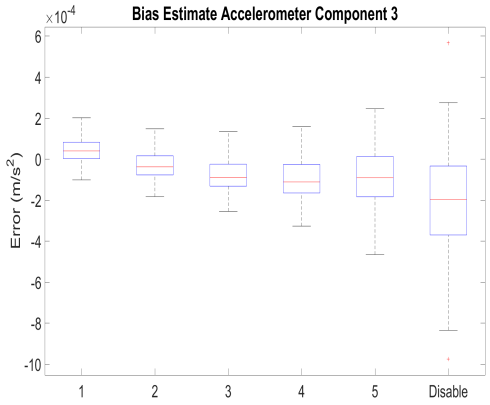

(b) Error in Bias Estimate Gyroscope X



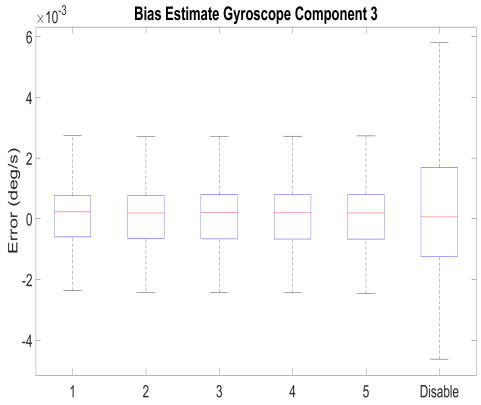

(c) Error in Bias Estimate Accelerometer Y




(d) Error in Bias Estimate Gyroscope Y




(e) Error in Bias Estimate Accelerometer Z



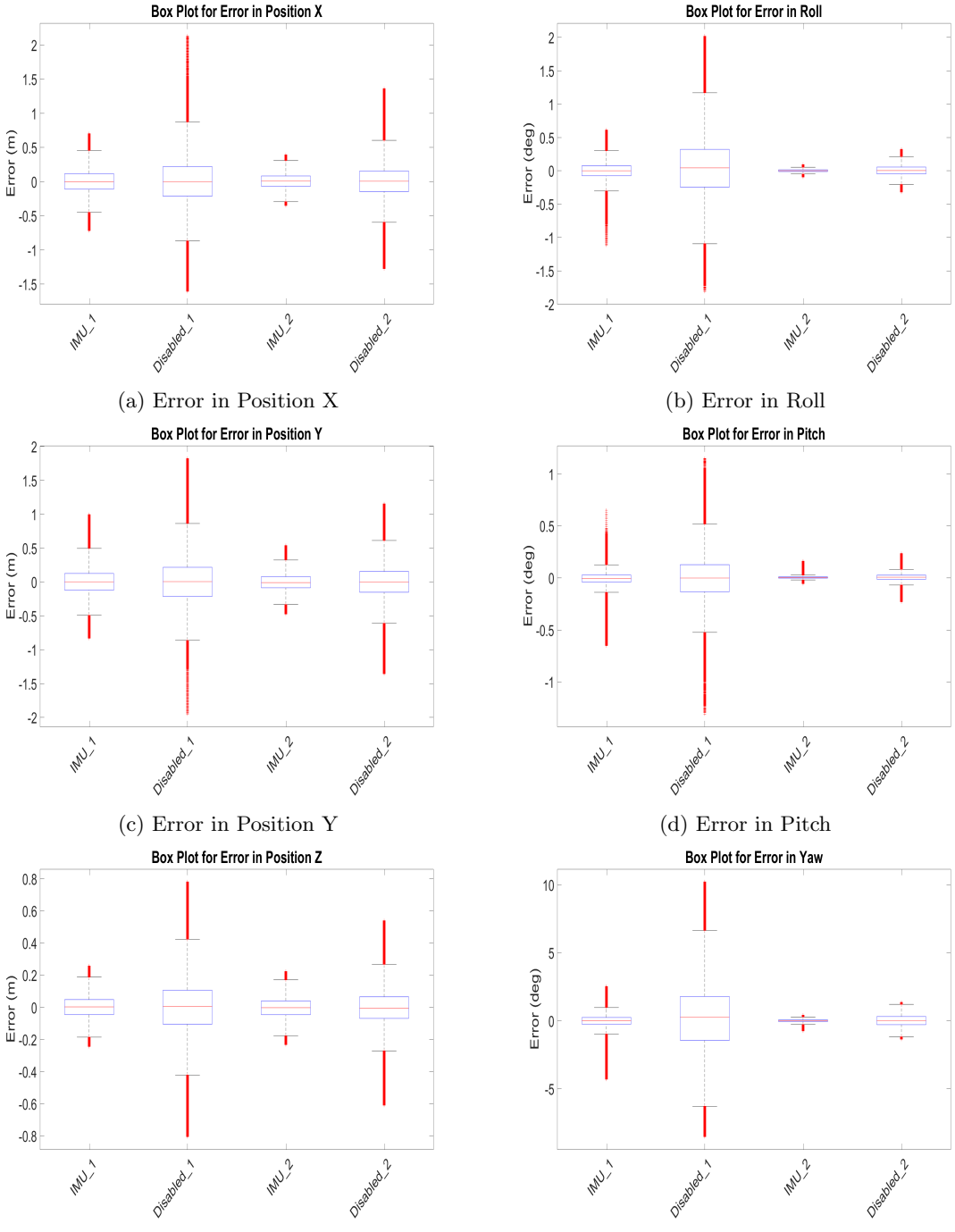

(f) Error in Bias Estimate Gyroscope Z

Figure 5.7: Box Plots of Errors in Estimated Bias of accelerometer and gyroscope with IMU 2

The Bias estimates of the gyroscope follow the same trend as in the previous experiment, no significant improvement in the Y component while there is an improvement for X and Z component. For the accelerometer, we can see a significant improvement in the variance of the estimates as we decrease the variance of the VDM measurements.

# 3 Comparison between the two IMUs

In the previous sections, the effect of the accuracy of the different IMU on different variances of VDM measurements was explored. In this section, the effect of IMUs on the variance of the estimates would be studied. For the study, we chose $VDM_2$ as the variance of the VDM measurements




(a) Error in Position X

(b) Error in Roll

(c) Error in Position Y

(d) Error in Pitch

(e) Error in Position Z

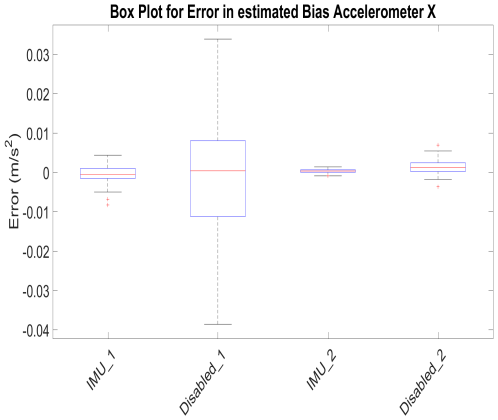
(f) Error in Yaw

Figure 5.8: Error distribution of the estimates by using the IMU 1 and IMU 2 with VDM measurements compared to the case without VDM measurements

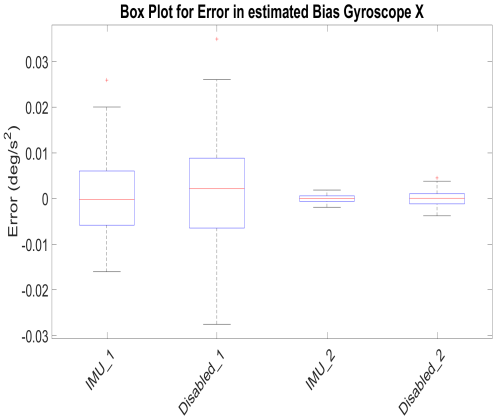
In both cases of the IMUs, there is an improvement in the variance of the estimates when VDM measurements are used. When the IMU is of a higher quality, the variance in the error estimates decreases irrespective of whether the VDM is used. But when an IMU of higher accuracy is used, the improvement in the variance of estimates from using VDM is lower compared to a case with

an IMU having lower accuracy. Hence adding the measurements from the VDM might not have any effect on systems containing sensors of high accuracy.

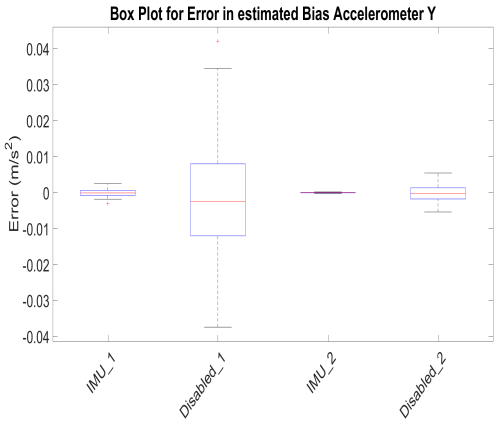
In the Figure (5.9), the effect of the noise in IMU on the accuracy of the Bias Estimate is shown. As noted in the previous sections, the bias estimate in the Y component of the gyroscope is not affected by the addition of VDM measurements. In the other cases, the variance of the estimates improve with the addition of the VDM measurements. But when an IMU of higher accuracy id used, the improvement gained from VDM measurements decreases.



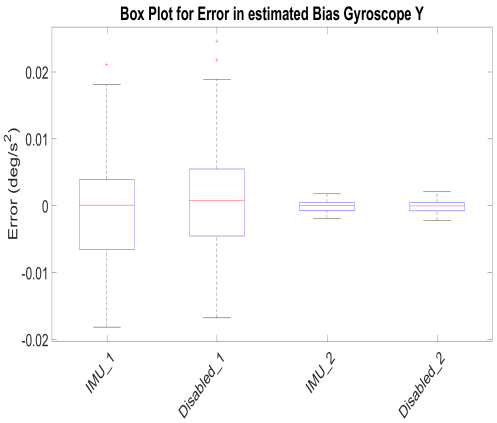

(a) Error in Bias Estimate Accelerometer X



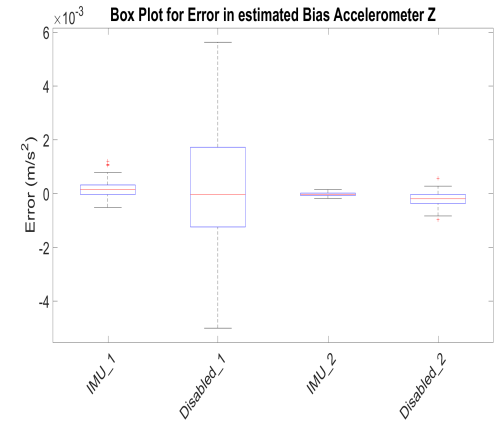

(b) Error in Bias Estimate Gyroscope X



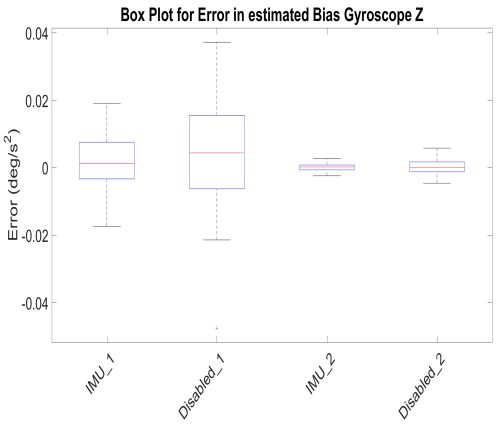

(c) Error in Bias Estimate Accelerometer Y




(d) Error in Bias Estimate Gyroscope Y




(e) Error in Bias Estimate Accelerometer Z




(f) Error in Bias Estimate Gyroscope Z

Figure 5.9: Error Distribution of the Bias Estimate using IMU 1 and IMU 2 with VDM measurements compared to the case without VDM measurements

# Chapter 6

# Trajectory Estimation with VDM-model calibration

In the previous chapter, the experiments were done with known parameters for the VDM sensor. More often than not, it is difficult to determine the value of the parameter before the flight. Hence the value of these parameters has to be estimated along with the estimation of trajectory. This chapter will deal with such scenarios. The following values for parameters were used to generate the measurements from the *MATLAB* simulator. These are the same values that were used in the previous chapter, it is listed here again for the convenience of the reader.

- $Drag\ Parameters = \begin{bmatrix} d_1, d_2, d_3 \end{bmatrix} = \begin{bmatrix} 0.1\mathrm{N\,s^2/m^2}, 0.1\mathrm{N\,s^2/m^2}, 0.2\mathrm{N\,s^2/m^2} \end{bmatrix}$
- $Copter\ Parameters = \begin{bmatrix} l, m \end{bmatrix} = \begin{bmatrix} 0.4\mathrm{m}, 1.454\mathrm{kg} \end{bmatrix}$
- $Motor\ Parameters = \begin{bmatrix} b, d \end{bmatrix} = \begin{bmatrix} 1.6e^{-4}\mathrm{N\,s^2}, 7.5e^{-7}\mathrm{N\,m\,s^2} \end{bmatrix}$
- $Inertia\ Matrix\ Diagonal = \begin{bmatrix} I_{xx}, I_{yy}, I_{zz} \end{bmatrix} = \begin{bmatrix} 5.8e^{-3}\mathrm{kg\,m^2}, 6.0e^{-3}\mathrm{kg\,m^2}, 1.1e^{-2}\mathrm{kg\,m^2} \end{bmatrix}$
- $Intertia\ Matrix\ Off-Diagonal = \begin{bmatrix} I_{xy}, I_{xz}, I_{yz} \end{bmatrix} = \begin{bmatrix} 0\mathrm{kg\,m^2}, 0\mathrm{kg\,m^2}, 0\mathrm{kg\,m^2} \end{bmatrix}$

In order for the dynamic network to estimate the sensor parameters, an initial guess should be provided to them. The initial guess was generated from a Gaussian distribution centered at the true value of the parameter. Since the length of the copter arm and mass of the copter can easily be measured precisely, those parameters were not estimated on flight. The following standard deviation ($\sigma$) was used for each parameter when they were generated for the Monte Carlo Simulation.

- $Drag\ Parameters = \begin{bmatrix} d_1, d_2, d_3 \end{bmatrix} = \begin{bmatrix} 1e^{-1}\mathrm{N\,s^2/m^2}, 1e^{-1}\mathrm{N\,s^2/m^2}, 1e^{-1}\mathrm{N\,s^2/m^2} \end{bmatrix}$
- $Motor\ Parameters = \begin{bmatrix} b, d \end{bmatrix} = \begin{bmatrix} 1e^{-4}\mathrm{N\,s^2}, 1e^{-7}\mathrm{N\,m\,s^2} \end{bmatrix}$
- $Inertia\ Matrix\ Diagonal = \begin{bmatrix} I_{xx}, I_{yy}, I_{zz} \end{bmatrix} = \begin{bmatrix} 1e^{-2}\mathrm{kg\,m^2}, 1e^{-2}\mathrm{kg\,m^2}, 1e^{-2}\mathrm{kg\,m^2} \end{bmatrix}$
- $Intertia\ Matrix\ Off-Diagonal = \begin{bmatrix} I_{xy}, I_{xz}, I_{yz} \end{bmatrix} = \begin{bmatrix} 1e^{-8}\mathrm{kg\,m^2}, 1e^{-8}\mathrm{kg\,m^2}, 1e^{-8}\mathrm{kg\,m^2} \end{bmatrix}$

Like in the previous case, experiments were done for 5 different sets of variances for VDM measurements.

## 1 Distribution of Error in the Trajectory Estimation using IMU 1

The IMU has the following properties

### Accelerometer

- White noise $\sigma = 4.02e^{-3}\ m/s^2/\sqrt{Hz}$
- Random Bias $\sigma = 7.84e^{-2}\ m/s^2$

## Gyroscope

- White noise $\sigma = 4.7e^{-5}\ rad/s/\sqrt{Hz}$
- Random Bias $\sigma = 3.5e^{-3}\ rad/s$

The measurements were generated as described in Chapter (5). A Monte Carlo simulation of 100 iterations was performed. Results obtained from it are shown in Figure (6.1) and (6.2)

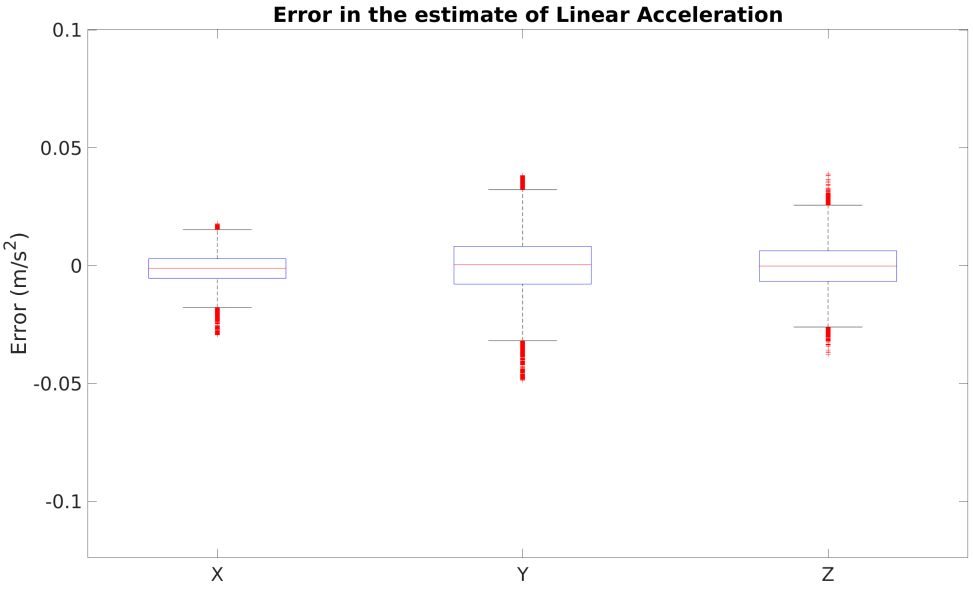



Figure 6.1: Error in the linear acceleration of the estimate

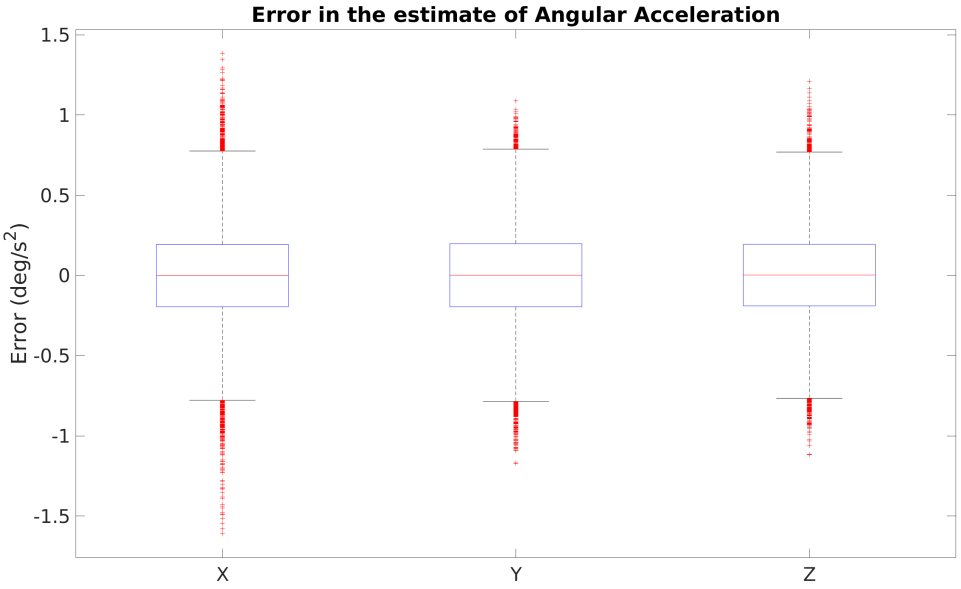



Figure 6.2: Error in the angular acceleration of the estimate

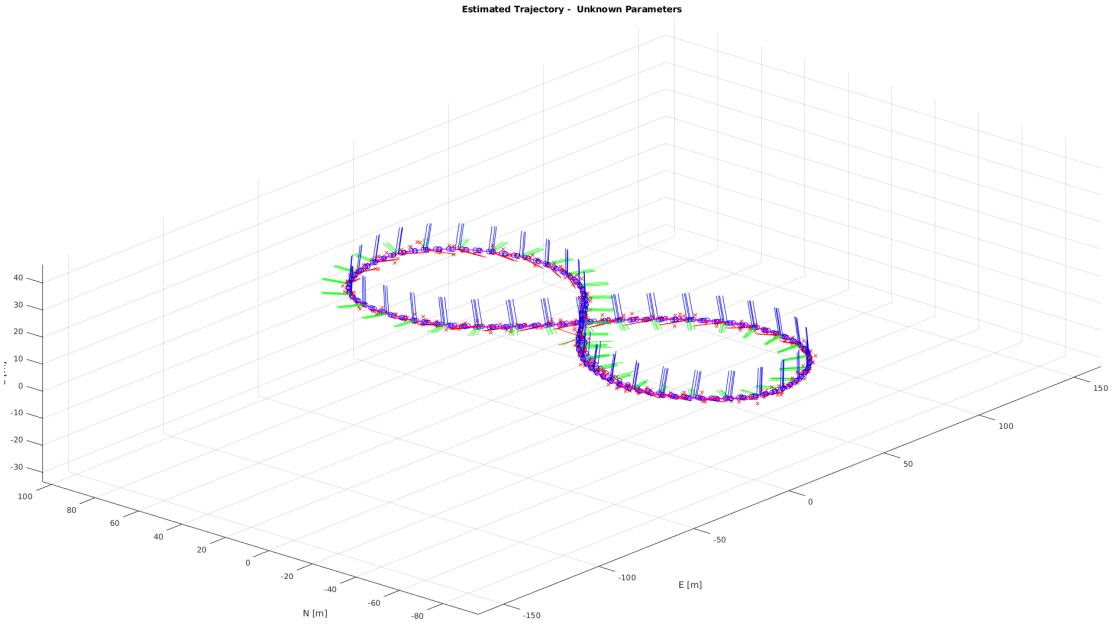



Figure 6.3: The Trajectory Estimated using Dynamic Network

Even when the parameters of the sensor is estimated, the variance in error remain smaller and has a mean closer to the zero. The Figures (6.4) and (6.5) show a comparison plot of the error in the estimate between the two studies, with a known parameter and with an unknown parameter. The variance in the error of the estimate mostly remains the same or is slightly less in the second study.

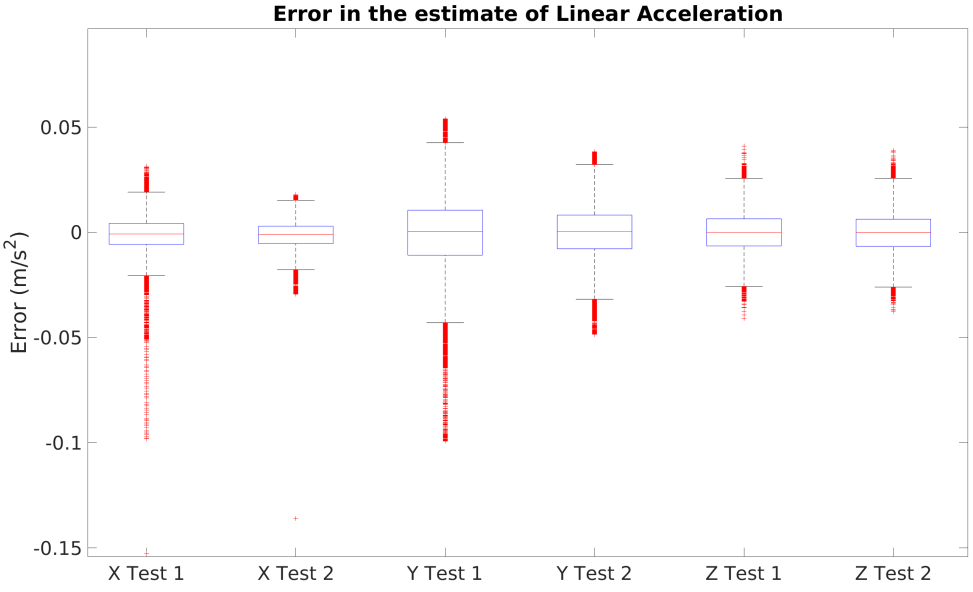



Figure 6.4: Error in the linear acceleration of the estimate

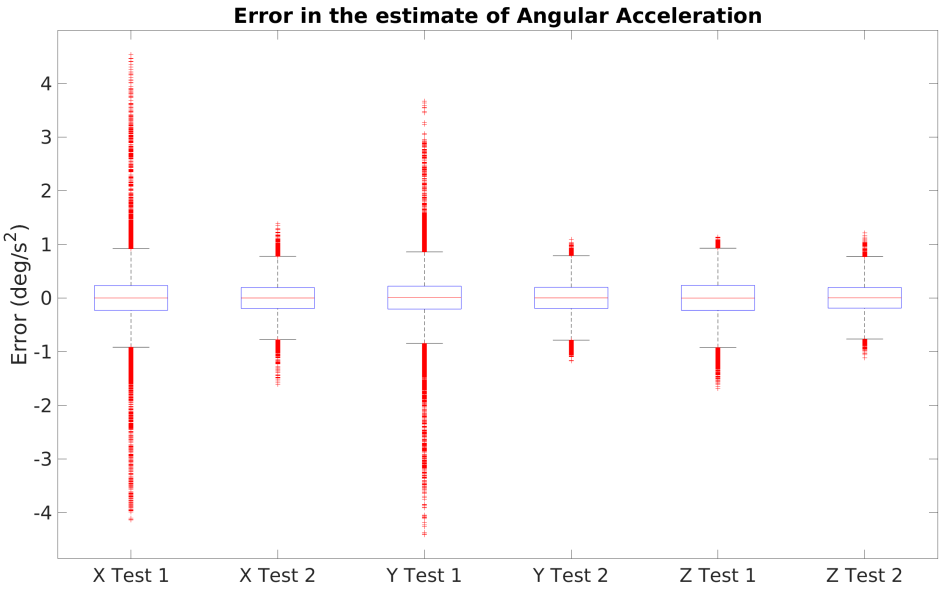
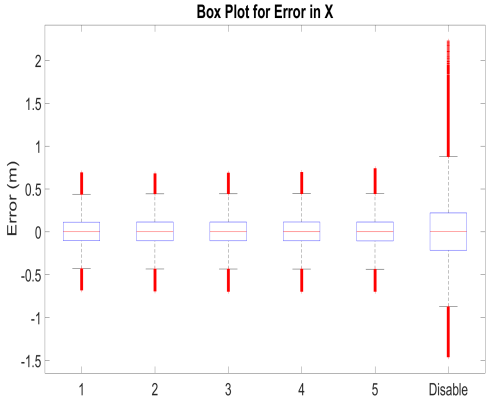
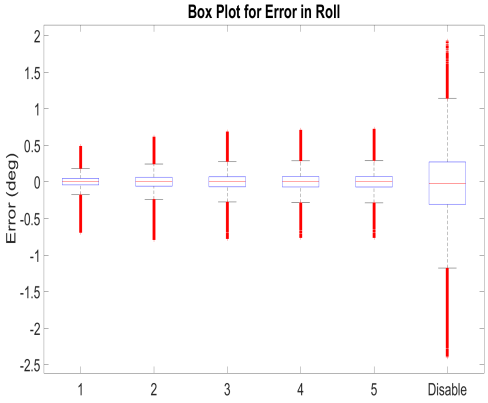



Figure 6.5: Error in the angular acceleration of the estimate

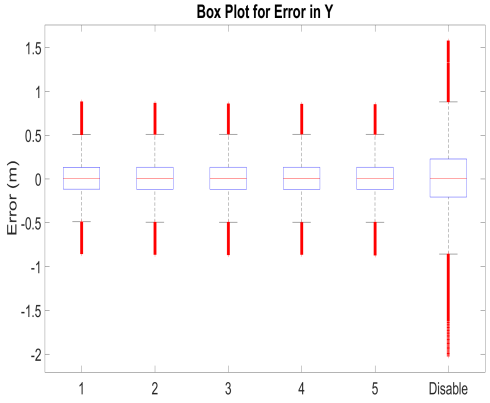
In order to get into a detailed analysis, the error distribution of the trajectory estimate obtained from the Monte Carlo Simulation should be studied. The results from the Monte Carlo Simulation is shown in Figure (6.6). As in the previous cases, the usage of VDM measurements reduces the variance the estimated quantities. But unlike the case where the value of the parameters was known, here the variance of the estimates is invariant with respect to the variance given for VDM measurements. The expected increase in the variance of the estimates might have been offset by the freedom to perturb the sensor parameters. There can be values around the true value of sensor parameters which could give less error to the trajectory estimate.



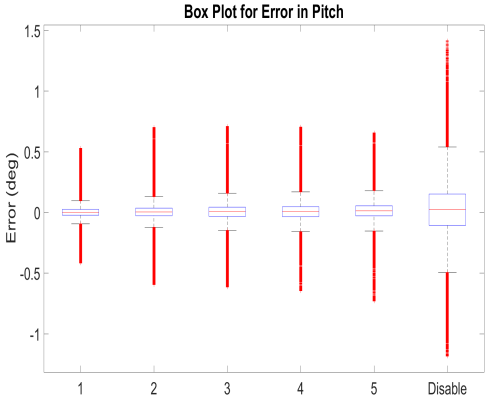

(a) Error in position X



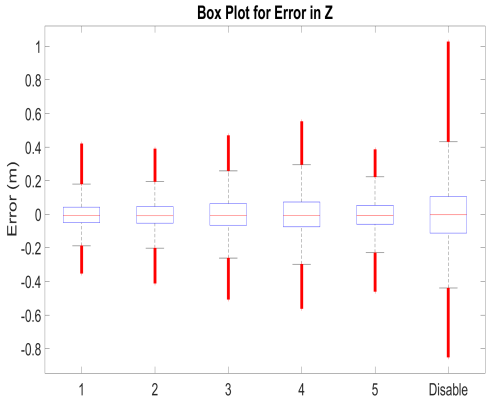

(b) Error in Orientation Roll



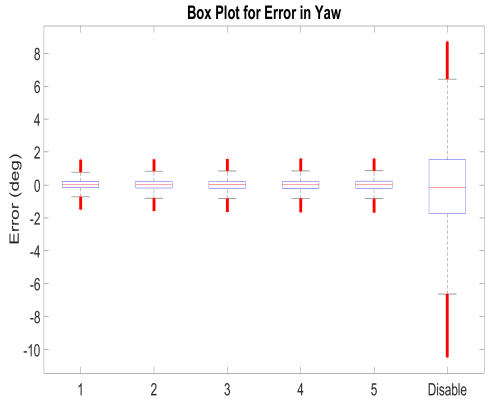

(c) Error in position Y




(d) Error in Orientation Pitch




(e) Error in position Z




(f) Error in Orientation Yaw

Figure 6.6: Box Plots of Errors in Position and Orientation with IMU 1

## 1.1 Parameter Estimation

In this section, a study about the estimation of sensor parameter is done. The term $(J_{\Delta x}^T \Omega J_{\Delta x})^{-1}$ in the Equation (2.18) gives the covariance matrix for the estimated values. This could be used as a metric to determine the confidence of that estimate. In the following figures, the true value and the estimated value of a sensor parameter are plotted. Estimated values are shown with a $3\sigma$ confidence, where $\sigma$ is obtained from the covariance matrix.

### Drag Parameters

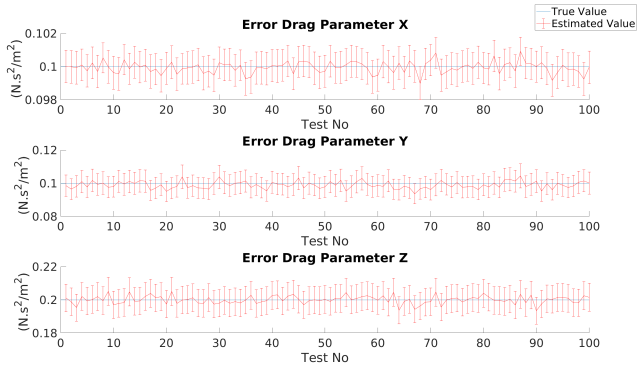



Figure 6.7: Estimated Proportional Constant of Drag Force

### Motor Parameters

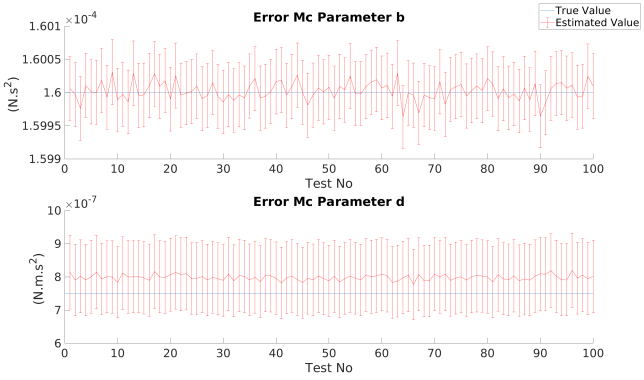



Figure 6.8: Estimated Motor Parameters

### Inertia Matrix - Diagonal Elements

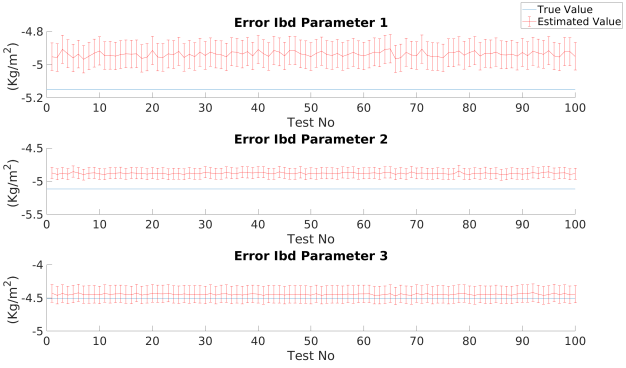



Figure 6.9: Estimated Diagonal Elements of Inertia Matrix

### Inertia Matrix - Off-Diagonal Elements

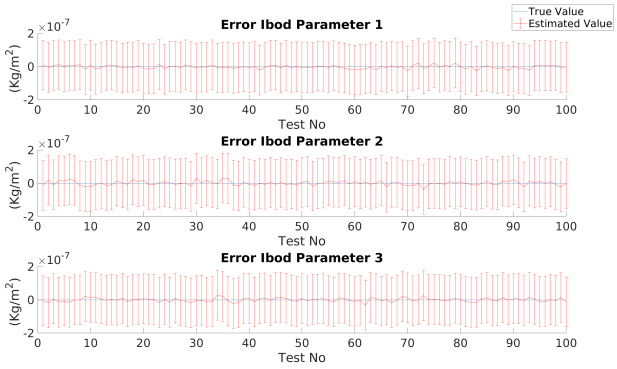



Figure 6.10: Estimated Off-Diagonal Elements of Inertia Matrix

### Box Plot of Errors in the Estimated Parameter

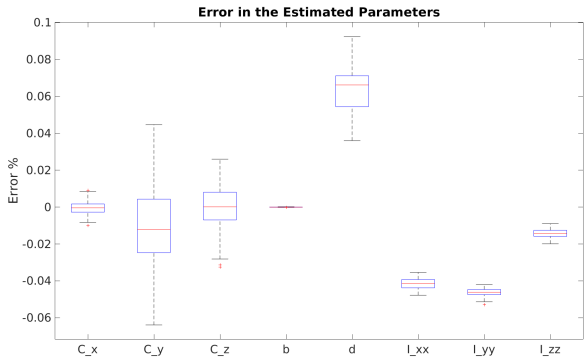



Figure 6.11: Box plot of error in the estimated parameters

From Figure (6.11), one can see that for the $C_x$, $C_z$, and $b$, the errors in the estimate is distributed around zero. In the case of $C_y$, one can see that the mean is shifted from zero. This is due to the observability issues that arise from the trajectory that is used. But for the parameters $d$, $I_{xx}$, $I_{yy}$, and $I_{zz}$ the errors in the estimate are centered around non-zero values. While $d$ is distributed around a positive value, the diagonal elements of inertia are distributed around negative values. This indicates that these parameters might be correlated and hence the positive error in one is offset by the negative error in the other. In the next section, the correlation between the parameters is studied.

### Correlation Matrix

The distribution of errors in the estimated parameters of $d$, $I_{xx}$, $I_{yy}$, and $I_{zz}$ from Figure(6.11) indicates the possible existence of a correlation between them. From the covariance matrix obtained at the end of the least square estimation, the correlation between the parameters is calculated and is plotted in Figure (6.12). As expected there is a correlation between parameter $d$ and diagonal elements of the inertia matrix.

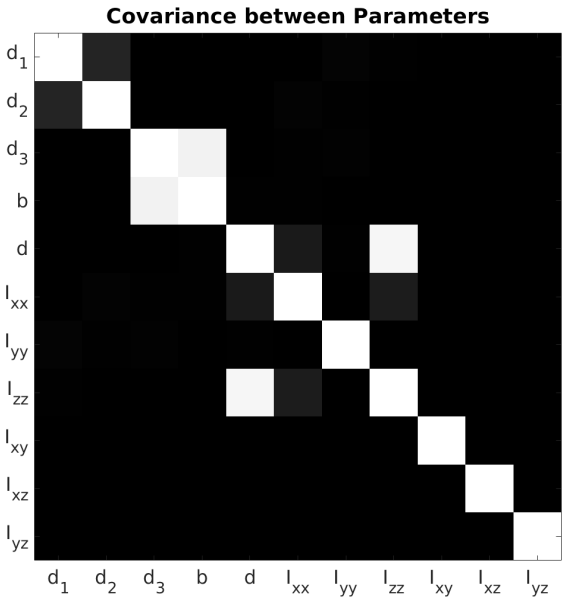



Figure 6.12: Correlation between various parameters in the VDM model

This correlation could also be shown mathematically. For the study, the off-diagonal elements of the inertia matrix were set to zero. Hence the inertia matrix is

$$I = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{6.1}$$

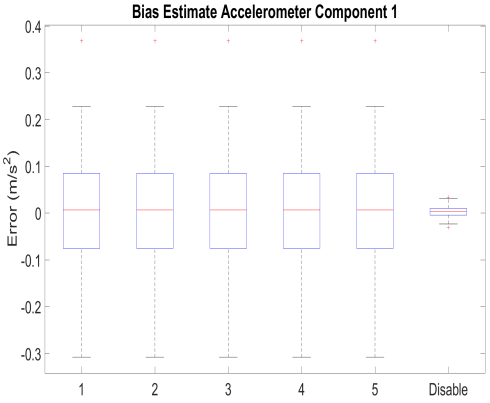
From Equations (3.20) and (3.18) and using a diagonal inertia matrix, one could write the third component of angular acceleration as

$$\dot{\omega}_z^b = \frac{d(-z_1^2 - z_2^2 + \dots}{I_{zz}} \tag{6.2}$$

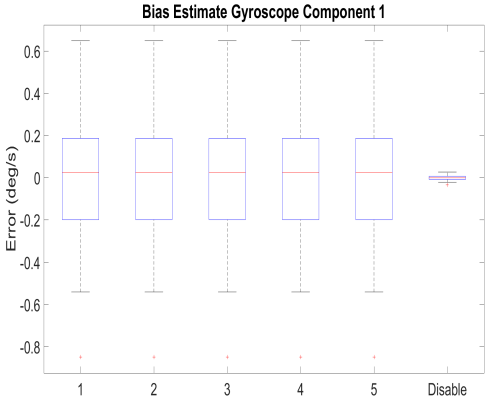
From the above relation, one could see that the positive error in $d$ could be offset by a negative error in the diagonal element of the inertia matrix. In the study, since the off-diagonal parameters are estimated and hence has a non-zero value. So the other elements in the inertia matrix also become slightly correlated with $d$.

## 1.2 Bias Estimation

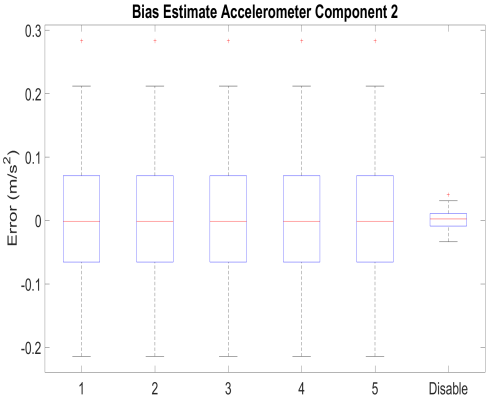
Like in the other experiments, a bias was added to the IMU measurements. But unlike the previous cases, here the addition of VDM measurement is increasing the variance in the trajectory estimates. As shown earlier, if the estimated parameters are correlated, the positive error of one could be offset by the negative error in the other. Hence the reason for this behavior is the correlation between bias and the estimated sensor parameters.



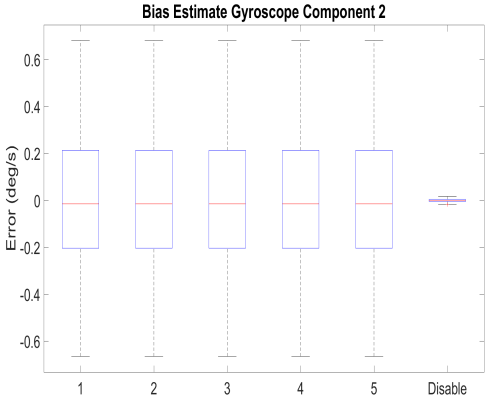

(a) Error in Bias Estimate Accelorometer X



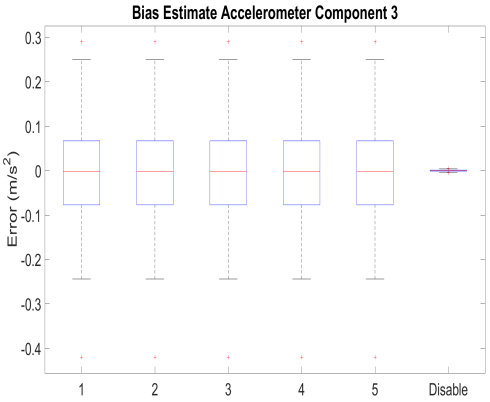

(b) Error in Bias Estimate Gyroscope X



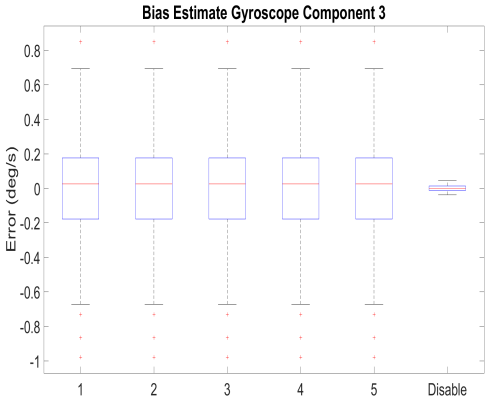

(c) Error in Bias Estimate Accelorometer Y




(d) Error in Bias Estimate Gyroscope Y




(e) Error in Bias Estimate Accelorometer Z




(f) Error in Bias Estimate Gyroscope Z

Figure 6.13: Box Plots of Errors in Estimated Bias of accelerometer and gyroscope with IMU 1

# 2 Distribution of Error in the Trajectory Estimate using IMU 2

The IMU has the following properties

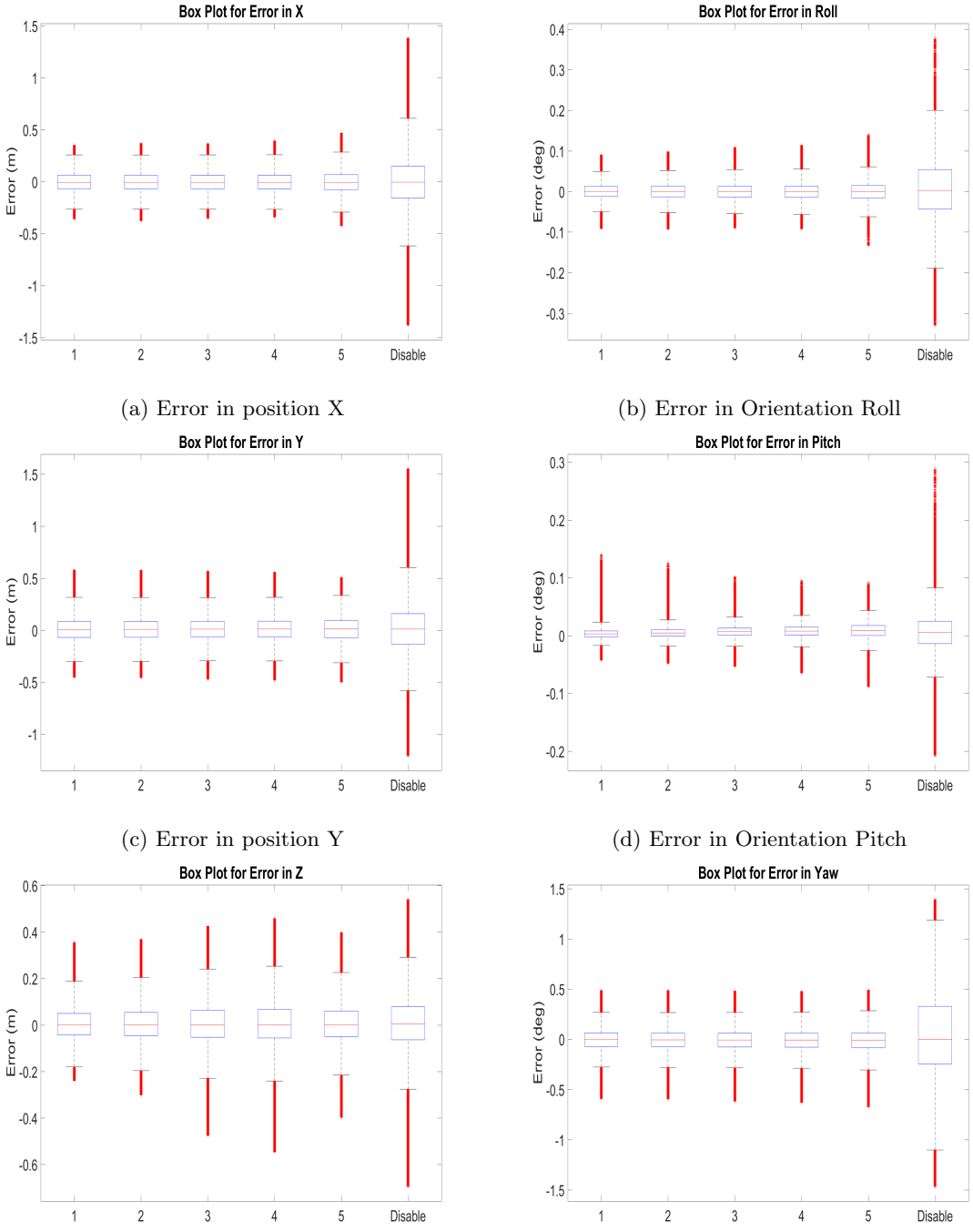
## Accelerometer

- White noise $\sigma = 4.02e^{-4}\ m/s^2/\sqrt{Hz}$
- Random Bias $\sigma = 7.84e^{-2}\ m/s^2$

## Gyroscope

- White noise $\sigma = 4.7e^{-6}\ rad/s/\sqrt{Hz}$
- Random Bias $\sigma = 3.5e^{-3}\ rad/s$

The noisy measurements were generated and a Monte Carlo Simulation of 100 iterations was done. The results are summarized in Figure (6.14)




(a) Error in position X

(b) Error in Orientation Roll

(c) Error in position Y

(d) Error in Orientation Pitch

(e) Error in position Z

(f) Error in Orientation Yaw

Figure 6.14: Box Plots of Errors in Position and Orientation with IMU 2

The behavior is the same as that we saw in the previous experiment in this chapter. There is an improvement in the estimate when VDM measurements are used, but that improvement remains invariant to the variance of the VDM measurements,

## 2.1 Parameter Estimation

In this section, a study about the estimation of sensor parameter is done. In the following figures, the true value and the estimated value of a sensor parameter are plotted. Estimated values are shown with a $3\sigma$ confidence.

### Drag Parameters

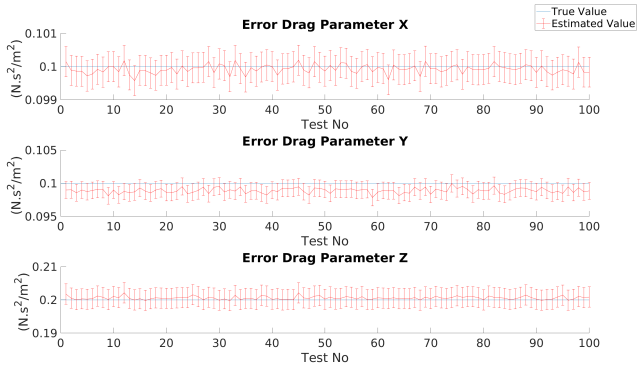



Figure 6.15: Estimated Proportional Constant of Drag Force

### Motor Parameters

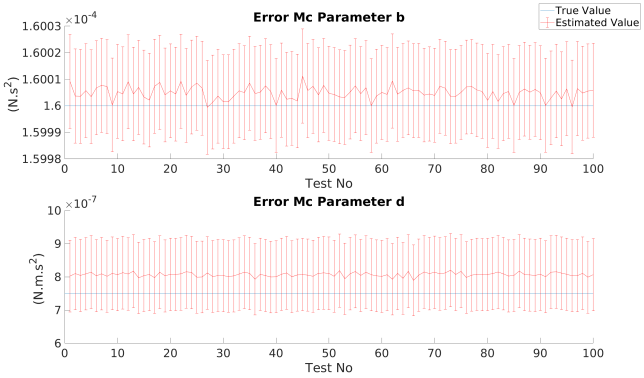



Figure 6.16: Estimated Motor Parameters

### Inertia Matrix - Diagonal Elements

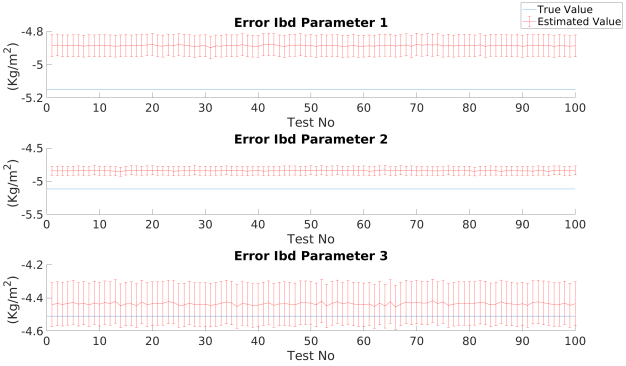



Figure 6.17: Estimated Diagonal Elements of Inertia Matrix

### Inertia Matrix - Off-Diagonal Elements

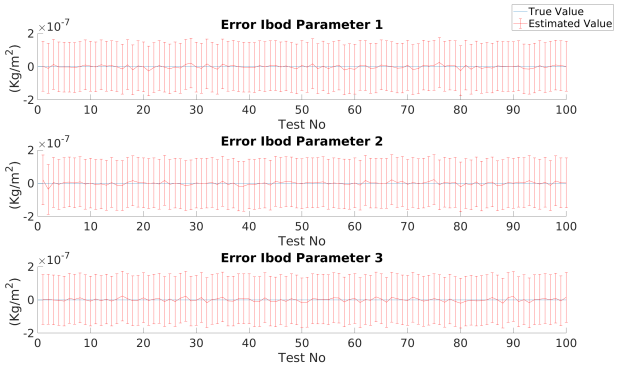



Figure 6.18: Estimated Off-Diagonal Elements of Inertia Matrix

### Box Plot of Errors in the Estimated Parameter

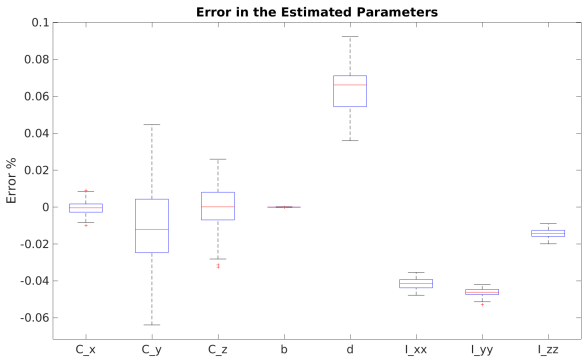



Figure 6.19: Box plot of error in the estimated parameters

The same behavior as seen in Figure (6.11) is seen in Figure (6.11). In order to study the effect of the using a better IMU on the errors in the estimate, box plots of the errors from the two experiments were plotted against each other on Figures (6.20) and (6.21). As expected using a better IMU improves the errors in the estimated parameters.

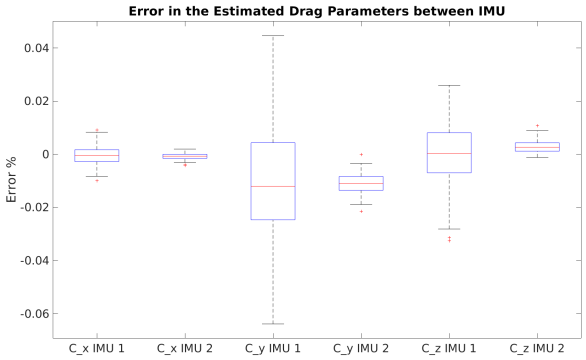



Figure 6.20: Box Plots of Errors in the Estimate of drag parameters in the two experiments

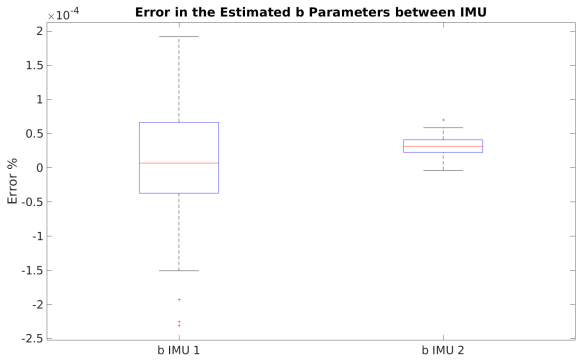



Figure 6.21: Box Plots of errors in the estimate of parameter b in the two experiments

### Correlation Matrix

Like the previous experiment, the diagonal value of the inertia matrix doesn't converge to the true nor it is within $3\sigma$ of the estimated value. The correlation between various parameters could be seen in Figure(6.22).

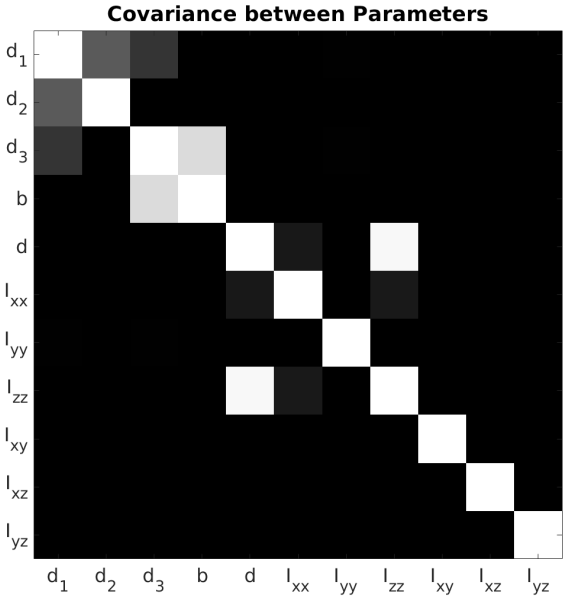
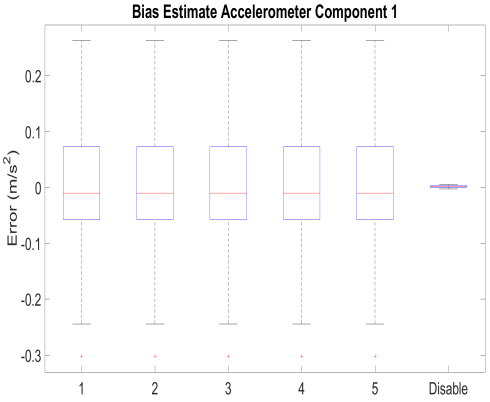
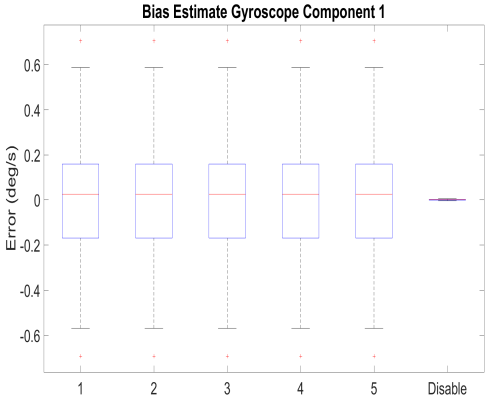



Figure 6.22: Correlation between various parameters in the VDM model

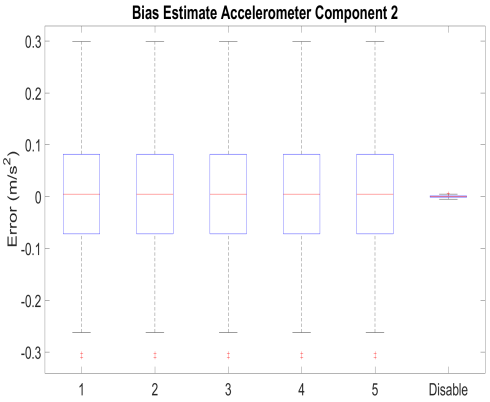
## 2.2 Bias Estimation



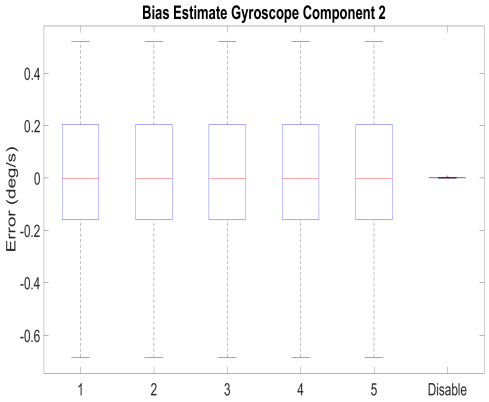

(a) Error in Bias Estimate Accelorometer X



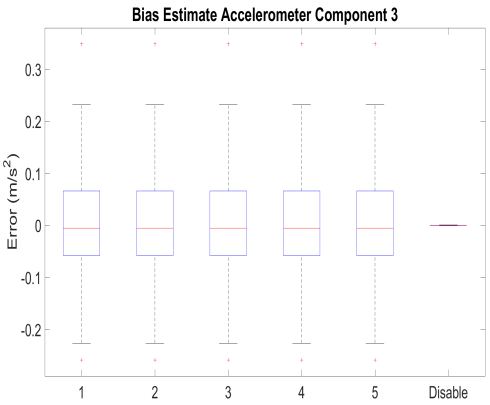

(b) Error in Bias Estimate Gyroscope X



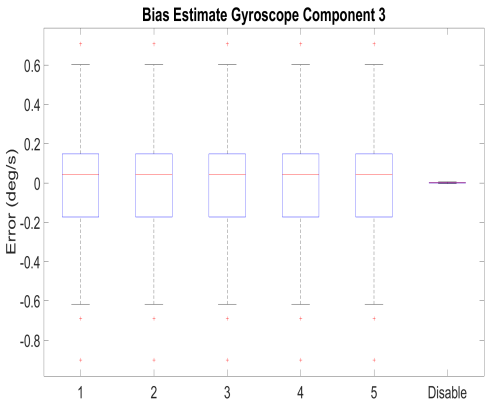

(c) Error in Bias Estimate Accelorometer Y




(d) Error in Bias Estimate Gyroscope Y




(e) Error in Bias Estimate Accelorometer Z

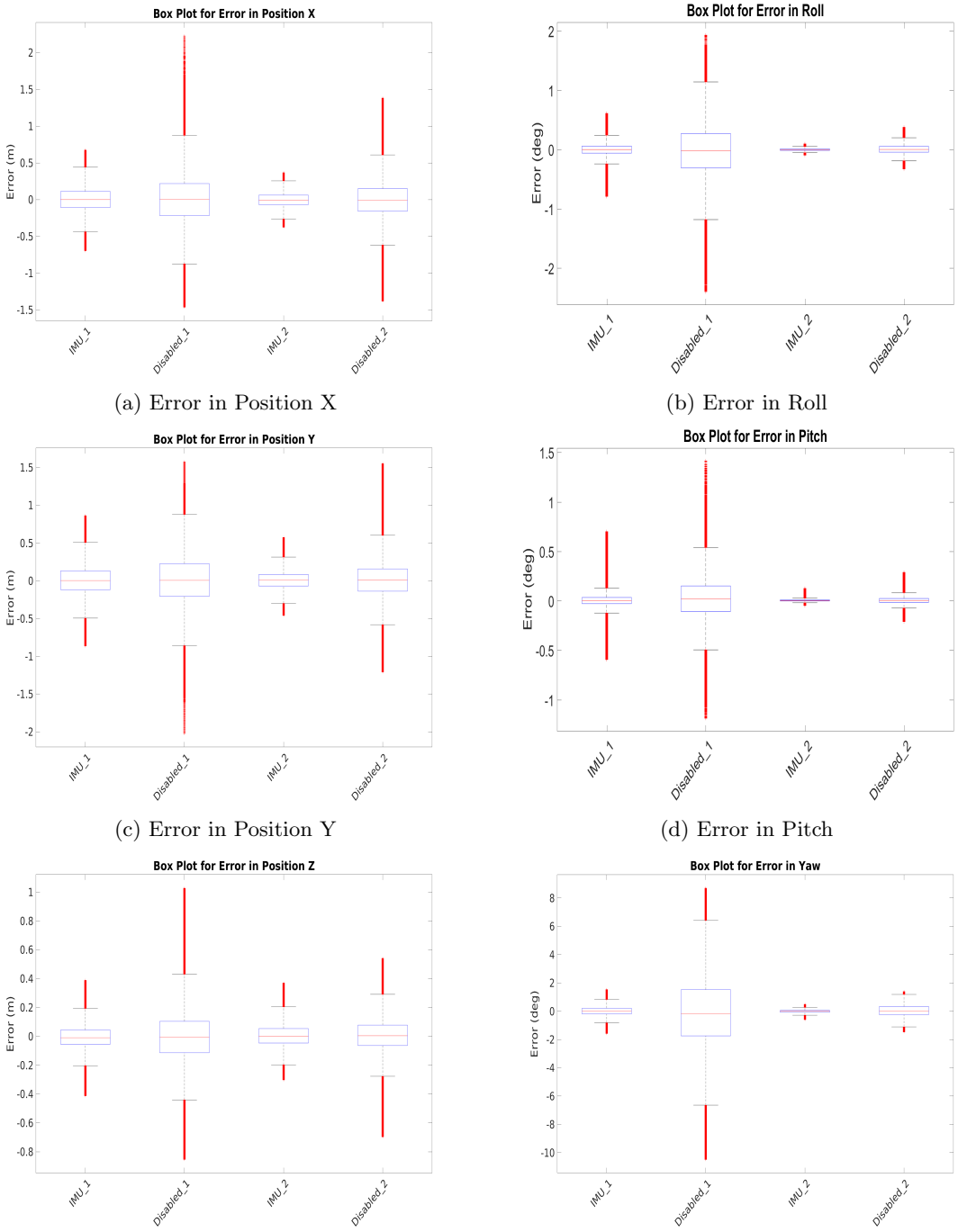



(f) Error in Bias Estimate Gyroscope Z

Figure 6.23: Box Plots of Errors in Estimated Bias of accelerometer and gyroscope with IMU 1

Like the previous experiment, the variance is high for bias estimates when VDM measurements are used. Also, there is no change in the variance of the estimate with respect the variance of the VDM measurements.

# 3 Comparison Between the Two IMUs




(a) Error in Position X

(b) Error in Roll

(c) Error in Position Y

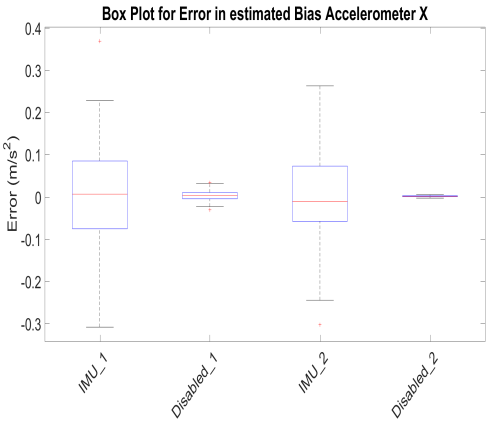
(d) Error in Pitch

(e) Error in Position Z

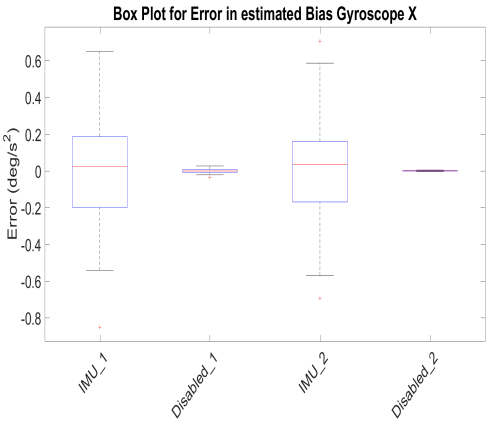
(f) Error in Yaw

Figure 6.24: Error distribution of the estimates by using the IMU 1 and IMU 2 with VDM measurements compared to a case without VDM measurements

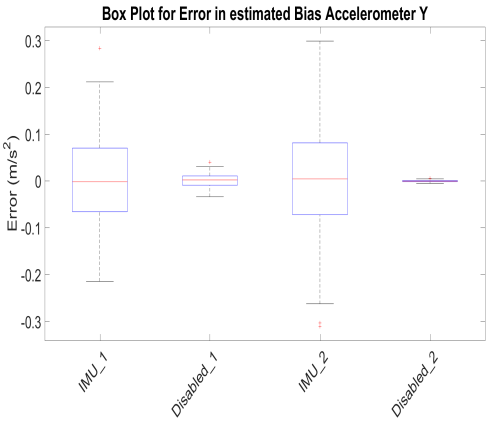
The results are similar to the results in Chapter (5). With VDM measurements, the variance of the estimate is lower. Also when a highly accurate IMU is used, the overall variance in the measurement decreases, but the gain from using VDM also decreases.



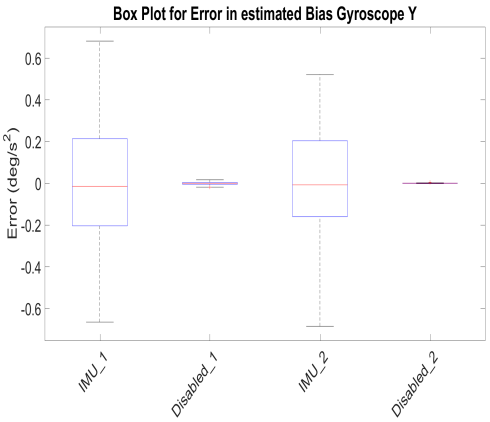

(a) Error in Bias Estimate Accelerometer X



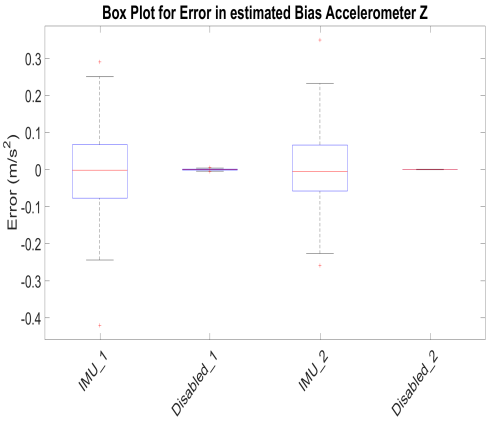

(b) Error in Bias Estimate Gyroscope X



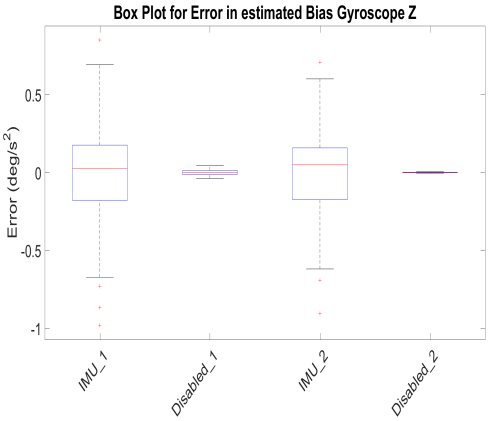

(c) Error in Bias Estimate Accelerometer Y




(d) Error in Bias Estimate Gyroscope Y




(e) Error in Bias Estimate Accelerometer Z




(f) Error in Bias Estimate Gyroscope Z

Figure 6.25: Error Distribution of the Bias Estimate using IMU 1 and IMU 2 with VDM measurements compared to a case without VDM measurements

There is a significant increase in the variance of the bias estimate when VDM measurements are used. When using VDM measurements, there isn't a significant difference in the variance of an estimate when different IMUs are used.

# Chapter 7

# Discussions

## Conclusions from the studies

Two studies were conducted as part of this project. In the first study, the trajectory was estimated with known parameters for the VDM sensors. In the second, the VDM sensor parameters were also estimated along with the trajectory. For each study, two experiments were performed using two different IMUs. The first IMU was noisier compared to the to the second. For each experiment, six cases were analyzed. The first five cases were with different variances for VDM measurement and the sixth case was done without using VDM measurements. Each case was analyzed by performing a Monte Carlo Simulation of hundred realizations. Various trends were observed across these studies. In all the studies, the usage of VDM measurements always reduced the variance in the error of the estimate. It was also observed that using a very high or very low variance for the VDM measurements results in an increase of the variance in the error of the estimate. At very high variance, the VDM measurements are ignored and at very low variance other sensor measurements are ignored, and it leads to an increase in the uncertainty of the estimates. The effect of variance used for the VDM measurements was more profound in the first study, whereas in the second study the effect of the variance was negligible. This may be due to the additional freedom available for perturbing the sensor parameters. When a better IMU was used, the improvement gained from using the VDM measurement decreased. When it comes to bias estimation, the improvement in the error of the estimate is only seen in the study with fixed parameters. Also when the parameters are known, the variance in the error of the bias estimate decreases with both the IMUs. But like in the case of trajectory estimates, the gain in performance is less when an IMU of higher quality is used. It was also noted that when the parameters are unknown, the variance in the error of the estimate increased with the addition of VDM measurements. In the second study where the parameters were unknown, certain trends were observed in the estimation of the parameters. In both the experiments of the second study, the true value of some sensor parameter didn't fall with the $3\sigma$ confidence interval of the estimated value of parameters. Upon further studies, these parameters were found to be correlated with each other. A factor that needs to be considered when doing parameter estimation is the observability of the parameters. This was evident in the estimation of $C_y$ parameter where the trajectory used in the project made it less observable. In the second study we also observed that when an IMU of better quality is used, the variances in the error of the estimate decreases.

## Limitations of the study and future work

The studies that were conducted were done using measurements that were generated using a simulator. The measurements were generated by adding noise to the ideal measurements. For the project, only the White Noise and Bias noise models were considered. As an extension of the work, more complicated stochastic processes like the random walk and Gauss-Markov could be used to study their effect. When the estimation of the sensor parameter was done, the wind was taken to be zero. Although in the current implementation a constant wind could be estimated, a feature to estimate varying wind could be implemented in the future. Another direction that can be taken would be to test the implemented frame on a real system.

# Bibliography


[1] L. Feng, H. Everett, J. Borenstein, U. of Michigan., O. R. N. Laboratory., and U. States., "Where am I?" : sensors and methods for autonomous mobile robot positioning. Vol. 3 / by L. Feng, J. Borenstein and H.R. Everett ; edited and compiled by J. Borenstien. University of Michigan Michigan, 1994.

[2] I. Colomina and P. Molina, "Unmanned aerial systems for photogrammetry and remote sensing: A review," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 92, pp. 79 – 97, 2014.

[3] K. P. Schwarz, M. A. Chapman, M. E. Cannon, and P. Gong, "An integrated ins/gps approach to the georeferencing of remotely sensed data," Photogrammetric Engineering and Remote Sensing, vol. 59, no. 11, pp. 1667–1674, 1993.

[4] M. Bryson and S. Sukkarieh, "UAV Localization Using Inertial Sensors and Satellite Positioning Systems," in Handbook of Unmanned Aerial Vehicles (K. P. Valavanis and G. J. Vachtsevanos, eds.), pp. 433–460, Dordrecht: Springer Netherlands, 2015.

[5] T. K. Lau, Y. Liu, and K. W. Lin, "Inertial-based localization for unmanned helicopters against gnss outage," IEEE Transactions on Aerospace and Electronic Systems, vol. 49, pp. 1932–1949, July 2013.

[6] J. W. A, M. G. B, A. L. C, J. J. W. A, S. Hana, D. S. D, and C. I, "Integration of gps/ins/vision sensors to navigate unmanned aerial vehicles."

[7] "Unmanned Aircraft Systems (UAS)." https://www.faa.gov/uas/.

[8] T. Canada, "Flying your drone safely and legally." https://www.tc.gc.ca/en/services/aviation/drone-safety/flying-drone-safely-legally.html, June 2018.

[9] "Modèles réduits et drones de loisir." http://www.ecologique-solidaire.gouv.fr/modeles-reduits-et-drones-loisir.

[10] A. Noureldin, T. B. Karamat, M. D. Eberts, and A. El-Shafie, "Performance enhancement of mems-based ins/gps integration for low-cost navigation applications," IEEE Transactions on Vehicular Technology, vol. 58, no. 3, pp. 1077–1096, 2009.

[11] S. Nassar and N. El-Sheimy, "Wavelet analysis for improving ins and ins/dgps navigation accuracy," Journal of Navigation, vol. 58, no. 1, pp. 119–134, 2005.

[12] S. Nassar and N. El-Sheimy, "A combined algorithm of improving ins error modeling and sensor measurements for accurate ins/gps navigation," GPS Solutions, vol. 10, no. 1, pp. 29–39, 2006.

[13] S. Yun, Y. J. Lee, and S. Sung, "Imu/vision/lidar integrated navigation system in gnss denied environments," Aerospace Conference, pp. 1–10, 2013.

[14] Y. M. Madany, H. M. Elkamchouchi, and M. M. Ahmed, "Modelling and simulation of robust navigation for unmanned air systems (uass) based on integration of multiple sensors fusion architecture," in Proceedings - UKSim-AMSS 7th European Modelling Symposium on Computer Modelling and Simulation, EMS 2013, pp. 719–724, 2013.

[15] M. U. de Haag, P. Duan, and A. Vadlamani, "Flight test and simulation results of an integrated dual airborne laser scanner and inertial navigator for uav applications," in 2014 IEEE 11th International Multi-Conference on Systems, Signals Devices (SSD14), pp. 1–6, Feb 2014.

[16] J. F. Vasconcelos, C. Silvestre, P. Oliveira, and B. Guerreiro, "Embedded uav model and laser aiding techniques for inertial navigation systems," Control Engineering Practice, vol. 18, no. 3, pp. 262–278, 2010.

[17] N. Dadkhah, B. Mettler, and D. Gebre-Egziabher, "A model-aided ahrs for micro aerial vehicle application," in 21st International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS 2008, vol. 1, pp. 81–89, 2008.

[18] A. A. Author, B. B. Author, and C. Author, "Title of article," Title of Journal, vol. 10, no. 2, pp. 49–53, 2005.

[19] M. Bryson and S. Sukkarieh, "Vehicle model aided inertial navigation for a uav using low-cost sensors," in Proceedings of the Australasian Conference on Robotics and Automation, pp. 1–9, Citeseer, 2004.

[20] M. Khaghani and J. Skaloud, "Autonomous vehicle dynamic model-based navigation for small uavs," Navigation, Journal of the Institute of Navigation, vol. 63, no. 3, pp. 345–358, 2016. Cited By :11.

[21] M. Khaghani and J. Skaloud, "Assessment of VDM-based autonomous navigation of a UAV under operational conditions," Robotics and Autonomous Systems, vol. 106, pp. 152–164, 2018.

[22] H. Strasdat, J. M. M. Montiel, and A. J. Davison, "Real-time monocular slam: Why filter?," in 2010 IEEE International Conference on Robotics and Automation, pp. 2657–2664, May 2010.

[23] T. Davis, Direct Methods for Sparse Linear Systems. Society for Industrial and Applied Mathematics, 2006.

[24] I. Colomina and M. Blazquez, "A UNIFIED APPROACH TO STATIC AND DYNAMIC MODELLING IN PHOTOGRAMMETRY AND REMOTE SENSING," p. 6.

[25] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," IEEE TRANSACTIONS ON INFORMATION THEORY, vol. 47, pp. 498–519, 1998.

[26] C. Hertzberg, R. Wagner, U. Frese, and L. Schröder, "Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds," Information Fusion, vol. 14, no. 1, pp. 57–77, 2013.

[27] J. G. Leishman, Principles of Helicopter Aerodynamics. No. 18 in Cambridge Aerospace Series, Cambridge ; New York: Cambridge University Press, 2nd ed ed., 2006. OCLC: ocm61463625.

[28] J. Gentle, "Computational Statistics," in International Encyclopedia of Education (Third Edition) (P. Peterson, E. Baker, and B. McGaw, eds.), pp. 93–97, Oxford: Elsevier, third edition ed., 2010.

[29] "The ROAMFREE sensor fusion library. Contribute to AIRLab-POLIMI/ROAMFREE development by creating an account on GitHub." AIRLab@POLIMI, Jan. 2019.

[30] D. A. CUCCI, "A general sensor-fusion and parameters self-calibration framework with applications in mobile robotics." https://www.politesi.polimi.it/handle/10589/98423, Dec. 2014.

[31] D. A. Cucci, M. Rehak, and J. Skaloud, "Bundle adjustment with raw inertial observations in UAV applications," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 130, pp. 1–12, 2017.