Design and collection of a benchmarking data set for attitude and pose estimators in UAV applications

Sixue Zhao
U5071410

Supervised by Dr. Jochen TRUMPF

November 2013

A thesis submitted in part fulfilment of the degree of
Bachelor of Engineering
The Department of Engineering
Australian National University

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university. To the best of the author's knowledge, it contains no material previously published or written by another person, except where due reference is made in the text.

Sixue Zhao
1 November 2013

Acknowledgements

I would first like to thank my supervisor Dr. Jochen Trumpf for providing me with the opportunity to undertake this project. The constant guidance and support he has shown me throughout the year was instrumental in the development of this thesis and my enhanced understanding of research based engineering. For that I am truly grateful.

I am also grateful to Professor Robert Mahony for the support and motivation. Your passion on researches and willingness to help really impressed me.

All the people in the Vision Robotics group are gratefully acknowledged. Especially I want to thank Mr Moses BANGURA, Mr Juan Adarve and Mr Xiaolei Hou who consistently support my research project and link me to the related research literatures.

Finally I would like to thank my parents, Qiuyi Zhao and Shunling Guo, for supporting me through my entire undergraduate degree.

Abstract

There are not many publicly available benchmarking data sets for performance comparisons between different attitude and pose estimation algorithms. This project aims at the design and collection of a benchmarking data set for the above benchmarking suite. The Mikrokopter-based quadrotor platform and four different attitude estimation algorithms are investigated in this thesis. A design criterion for the benchmarking data set is derived for further work on a performance comparison of the attitude filters. General sources of time synchronisation problems and different approaches to this problem are analysed. A time synchronisation solution using interpolation is proposed. The proposed solution has been implemented and simulated in MATLAB.

CONTENTS

List of Figures

List of Tables

Glossary of Terms

ANU	Australian National University
UAV	Unmanned Aerial Vehicle
MATLAB	A mathematical software package
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
VICON	Vicon Motion Capture System
CPU	Central Processing Unit
A/D converter	Analogue-to-Digital Convertor
SA-ADC	The Successive Approximation Analogue-to-Digital Convertor
Host PC	The computer in a Vicon system architecture that contains a dedicated Ethernet port for Vicon system communications and on which the core Vicon application software is installed.
EKF	the Extended Kalman Filter
MEKF	the Multiplicative Extended Kalman Filter
USQUE	the Unscented Quaternion Estimator
GAME filter	the Geometric Approximate Minimum-Energy filter
LCF	the Linear Complementary Filter
PI	Proportional-Integral

Chapter 1 Introduction

This chapter provides an introduction to the thesis topic. Section 1.1 details the motivation for investigating the problem and then details the aims of the thesis. In Section 1.2, an introduction into the area of benchmarking data sets, as well as a justification of the problem being addressed in this thesis are given. Lastly, in Section 1.3 the organisation of this thesis is described.

1.1 SCOPE AND MOTIVATION

Attitude and pose estimation for flying robots is a major research topic undertaken by the ANU vision robotics group. It is the problem of how to estimate the orientation and location of a rigid body with respect to an inertial reference frame. One main method to estimate the orientation of an aircraft is through attitude filters or estimators. These filters statistically combine measurements from several sensors in order to achieve an estimate of the attitude. For decades, different attitude estimation algorithms have been created and developed. There are no true optimal attitude filters, however, yet known in the literature (Zamani, 2003). It is therefore valuable to undertake a performance comparison study that provides a suggestion of optimal attitude filter for real flight scenarios.

This research project aims at the design of a benchmarking data set for the performance comparison of attitude and pose estimation algorithms for flying robots. It encompasses the collection and analysis of data sets from an Inertial Measurement Unit (IMU) and a Vicon Motion Capture System (VICON) for a Mikrokopter-based quadrotor, and pre-processing the above sensor data used for the measurement inputs of attitude filters for performance comparison. To compare the performance of different attitude filters with varying types of measurement inputs and to choose the filter with best performance for one particular type of quadrotor, a benchmarking data set suitable for filters with different characteristics should be applied. This research project does not physically compare different attitude filters, but aims to enable a user to perform benchmarking, without the requirement of the whole environment of Mikrokopter-based quadrotor and VICON system. The designed benchmarking data set is used both as measurement inputs for attitude filters and ground-truth data for performance evaluation and should help to provide a suggestion for an optimal attitude filter in a real flight scenario.

1.2 OVERVIEW OF BENCHMARKING DATA SET

A benchmarking data set is the data set within a benchmark suite. The problem of attitude estimation for Unmanned Aerial Vehicles (UAVs) has been actively researched by a large number of researchers. This is because the attitude state of the system provided by an attitude estimation algorithm will be used in the control process of the above vehicles to attain a certain automated navigation task (Zamani, 2013). For decades, different attitude estimation algorithms have been created and developed. The most reliable and convenient way to assess an attitude filter would be to test it on offline data sets collected from a sensor suite (Lee, Achtelik, Fraundorfer, Pollefeys, and Siegwart, 2010) in a real flight scenario through the environment. This approach could make it easy for researchers especially when they do not have access to real applications. Among the research literature, however, there are not many publicly available benchmarking data sets for performance comparisons between different attitude and pose estimation algorithms. Thus, it is necessary to research on a new benchmarking data set for evaluating attitude and pose estimation algorithms in UAV applications. The data sets are collected from the Mikrokopter-based quadrotor in a real flight scenario hovering within a space of approximately $2\text{m}\times 2\text{m}\times 2\text{m}$, and flying randomly within the sight of the Vicon system. The datasets consist of linear accelerations, angular velocities, absolute rotation angles and absolute operation directions from the IMU, and the ground truth measurements from the Vicon system with their respective timestamps.

1.3 THESIS CONTRIBUTION

The following are the main contributions of the thesis. The research literatures on each topic are provided and analysed in the related chapter. Chapter 2 covers the background material relevant to this research project including the Mikrokopter-based quadrotor platform, performance comparison of attitude filters, and the data sets. This examines the overall Mikrokopter platform in terms of hardware and software, two sensor groups, the representations of orientation, attitude determination, the meaning of performance comparison, sampling, and the function of Analogue-to Digital converters. Chapter 3 provides the comparison analysis of filters, sensor data, and issues identified for the sensor data measurements. Three types of input characteristics used by four attitude filters, as well as issues identified with the sensor measurements are presented. Chapter 4 discusses the time synchronisation problem. This examines all the possible reasons for two clocks to be unsynchronised. Chapter 5 details the time synchronisation problem on the Mikrokopter-based quadrotor platform, proceeds to discuss different approaches to solving the time synchronisation problem, and suggests an interpolation solution. Finally in Chapter 6 conclusions are derived from the findings of this thesis and suggestions for further work are discussed. The appendixes contain the initial sensor data collected within the ANU vision robotics group and the MATLAB code written for this project.

Chapter 2 Background

This chapter provides background information to the thesis topic. In Section 2.1, an introduction into the area of unmanned aerial vehicle (UAV), especially the Mikrokopter-based quadrotor platform, as well as the characteristics of the sensor groups is given. Section 2.2 details the performance comparison of attitude and pose estimators, along with the information of representations of orientation. Lastly, Section 2.3 exams the background information for data sets in terms of sampling and Analogue-Digital converter.

2.1 UAV APPLICATIONS AND THE QUADROTOR PLATFORM

An unmanned aerial vehicle (UAV) is an aircraft with no pilot on board. Historically, UAVs were controlled remotely while advanced techniques like feedback control and autonomous control are increasingly employed and researched (Lim, Park, Lee, and Kim, 2012). The UAV platforms in use at the ANU robotics group are quadrotors. A quadrotor is one type of aerial robot that is lifted and propelled by four rotors. This type of aerial robot is one of the most flexible and adaptable platforms for undertaking aerial robotics research. For decades, different quadrotor platforms have been designed and constructed to suit specific purposes. The Mikrokopter is one type of quadrotor unmanned aerial vehicle platform and is used for this research project.

2.1.1 The Mikrokopter Platform

The Mikrokopter is a quadrotor platform developed by a subsidiary of HiSystems GmbH in 2006 and is shown in Figure 2.1.

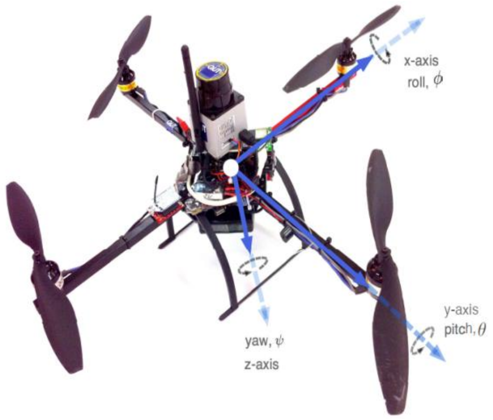


Figure 2.1: Mikrokopter (Sa, I. 2011)

The hardware and software configurations of the Mikrokopter platform are analysed below, with the visual representations taken from Lim et al. (2012).

The overall hardware configuration of a typical quadrotor UAV system contains four main parts: flight actuators, motor controllers, flight controller and ground control systems. This can be seen in Figure 2.2. The flight control platform (Lim et al., 2012) is used for flight avionics in the Mikrokopter. Flight-Ctrl is the main board of the Mikrokopter where the microcontroller and the sensors are mounted along with input/output (I/O) pins. The main chips being used are Microcontroller (ATmega644V), Gyroscope (ADXRS610), Accelerometer (LIS344ALH), and Magnetometer (KMZ51). Figure 2.3 below shows the Mikrokopter Flight-Ctrl board.

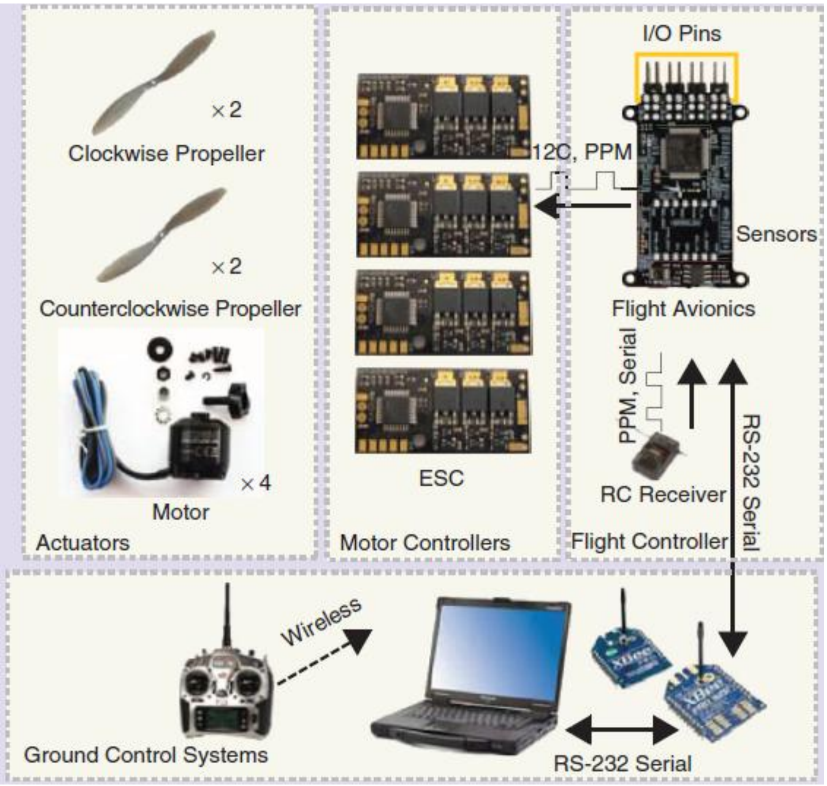


Figure 2.2: Configuration of a typical quadrotor UAV system (Lim et al., 2012).

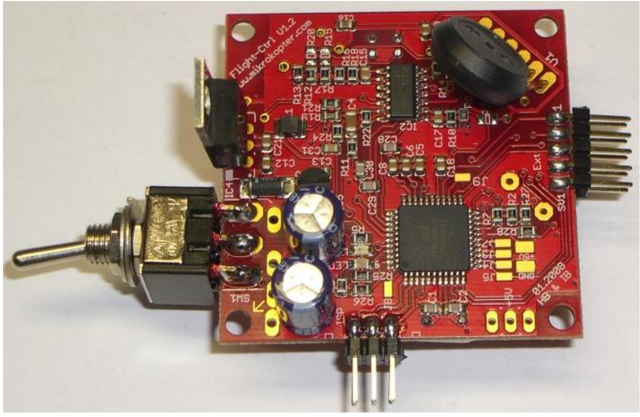


Figure 2.3: ‘CIMG8456_FC’ Flight-Ctrl board (HiSystems GmbH, 2008)

Along with the above hardware components, the Mikrokopter platform utilises software components for effective flying functions. These include attitude estimation and feedback control loop mechanisms.

The attitude estimation algorithm of the Mikrokopter is designed to fuse multiple independent noisy measurements of the same signal that have complementary spectral characteristics (Lim et al., 2012) through the implementation of a linear complementary filter (LCF). Figure 2.5 and Figure 2.5 are visual representations of the LCF mechanism (Pascoal, Kaminer, and Oliveira, 2000). The former structure represents the scenario that there is no steady-state estimation error, while the later structure compensates the variation of gyro bias over time. The measurement inputs of the accelerometer (notation y_u) and the gyroscope (notation y_x) on the same axis are compiled in LCF. The notation K_p, K_i, and $\frac{1}{s}$ are the gain parameters of the LCF and the notation $\hat{b}$ represents the bias measurement being estimated. The crossover frequency is determined via the structure $C(s)$. The attitude of the Mikrokopter (notation $\hat{\theta}$) at that point in time is able to be estimated through the above implementation.

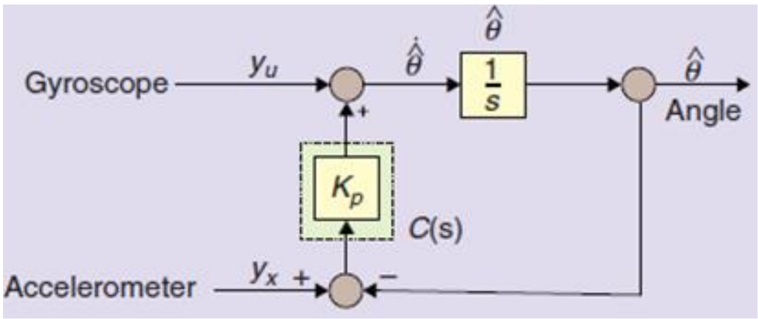


Figure 2.4: LCF without bias compensation (Pascoal et al., 2000).

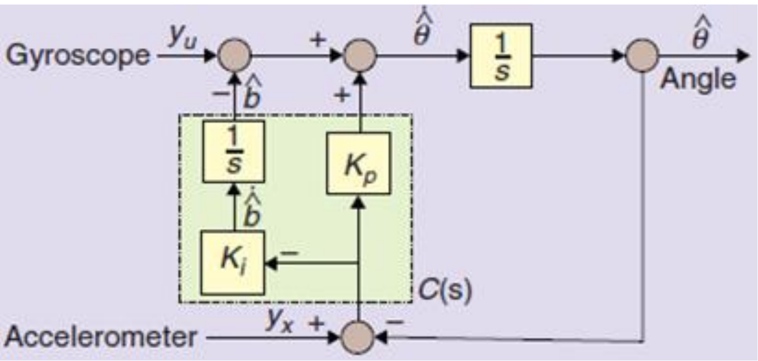


Figure 2.5: LCF with bias compensation (Pascoal et al., 2000).

The attitude state of the system provided by the above attitude estimation algorithm will then be used in the control process of the Mikrokopter to attain a certain automated navigation task. The configuration of the Mikrokopter controller is proportional-integral (PI) feedback control to achieve forward attitude error compensation. The structure of the Mikrokopter controller is shown in Figure 2.6.

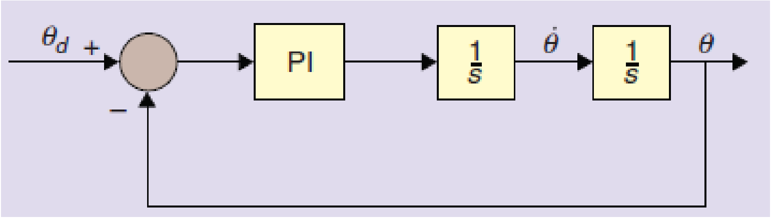


Figure 2.6: Mikrokopter controller (Lim et.al, 2012).

The next subsection provides more detail on the Inertial Measurement Unit (IMU) used in the Mikrokopter platform.

2.1.2 Inertial Measurement Unit (IMU)

An Inertial Measurement Unit, or IMU, is an electronic device combining gyroscopes, accelerometers, and sometimes also magnetometers. An IMU measures and reports the angular velocity, linear acceleration, and the magnetic field of an object with respect to a particular reference frame. In less expensive, small and light-weight IMUs, the inertial sensors produce estimation errors including any superimposed sensor drift and noise which tend to grow unbounded (Crassidis, Markley, and Cheng, 2007).

The IMU used in the Mikrokopter platform employs three one-axis gyroscopes and a three-axis accelerometer. The three one-axis gyroscopes are arranged into a 3-dimensional Cartesian orthogonal frame which follows the right hand rule. Each gyroscope channel measures the instantaneous angular velocity at a given point in time around one of the axis x, y, and z with respect to inertial frame in body-fixed coordinates. The information of the inertial frame and the body-fixed coordinates will be discussed in Section 2.2.1. The three-axis accelerometer has a configuration of a 3-dimensional Cartesian orthogonal coordinate system. Each accelerometer channel measures instantaneous acceleration indirectly through a force that is applied to that direction of the sensor. Similar to the gyroscope, the measurements of the accelerometer are with respect to the inertial frame, expressed in body-fixed coordinates. Due to the effect of gravitational force, the magnitude and the direction of the linear acceleration of the Mikrokopter in a flight scenario are dominated by the gravity acceleration. In other words, the linear accelerations apart from the direction of vertically downwards are relatively small. Thus, the overall linear acceleration of the Mikrokopter is approximately equal to the gravitational acceleration with a small angle and direction deviation.

As mentioned in the hardware configuration of the Mikrokopter platform above, the gyroscope being used is an ADXRS610 and the accelerometer is a LIS344ALH. Both of these two sensor types will output an analogue data, which is ratio metric with respect to a provided reference supply voltage. The analogue data will be converted to a digital value via an analogue to digital converter (A/D converter) as compiled in the attitude estimation algorithms. The A/D converter in the Mikrokopter platform is controlled by the microcontroller ATmega644V, and is discussed in section 2.3.2. The detailed specifications of the gyroscope and the accelerometer chips can be found in the sensor datasheets (Analogue Devices, 2010 and Stmicroelectronics, 2008, respectively). Table 2.1 below extracts the most important specifications of the gyroscope and the accelerometer from the sensor datasheets. This table provides the basis for the accuracy of sensor measurements, which will be discussed in section 3.4.

Table 2.1: Specifications of the IMU sensors.

Sensor Type	**Parameter**	**Conditions**	**Min.**	**Typ.**	**Max.**	**Unit**
Gyroscope Chip Name: ADXRS610 Outputs: Analogue Voltage Axis: One axis	Temperature Drift			±2		%
	Nonlinearity	Best fit straight line		0.1		% of FS
	Rate Noise Density	TA ≤ 25℃		0.05		°/sec/ √Hz
	Bandwidth		0.01		2500	Hz
	Turn-on time	Power on to ±½ °/sec of final			50	ms
Accelerometer Chip Name: LIS344ALH Outputs: Analogue Voltage Axis: Three axis	Bandwidth				1.8	kHz
	Sensitivity change Vs Temperature	Delta from +25 ℃		±0.01		%/℃
	Zero-g level change Vs Temperature	Delta from +25 ℃		±0.4		mg/℃
	Non linearity	Best fit straight line; Full-scale = ±2 g		±0.5		%FS
	Acceleration noise density	Vdd = 3.3 V; Full-scale = ±2 g		50		μg/√Hz
	Turn-on time at exit of Power-down mode	Cload expressed in μF		550×Cload+0.3		ms

2.1.3 VICON Motion Capture System

A VICON motion capture system (VICON in short) is an infrared marker-tracking system that is used to track the trajectories of objects in real time (The Johns Hopkins University, 2013). This system includes two main parts: the camera and the reflective dots. The camera is located in the experiment environment and is equipped with an array of Infrared (IR) LEDS and IR optical filters. The dots, also known as markers, which will reflect the IR radiation, are connected on the detected object, the Mikrokopter platform. At least four markers will be attached to the Mikrokopter in a configuration that is neither collinear nor symmetrical to ensure accurate tracking (Davis, 2013). The cameras use an onboard processor and IR optical filters to capture the location of the dots. Through the above processes, the VICON system tracks the trajectories of objects. Figure 2.7 below is an image of the Vicon System tracking an UAV platform. Further information about the VICON motion capture system can be found in http://www.vicon.com.

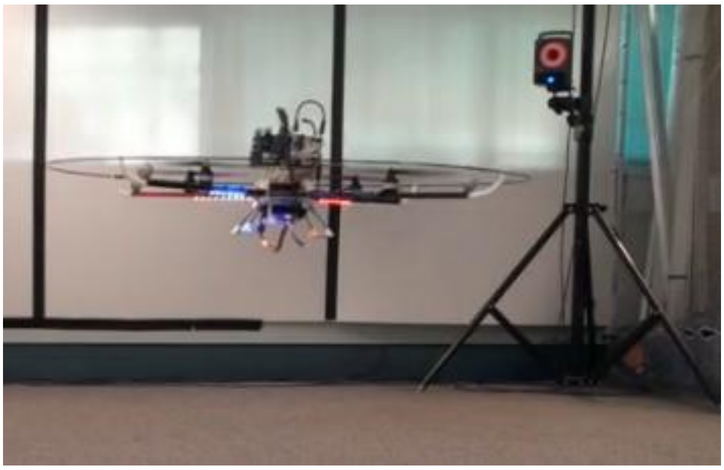


Figure 2.7: Vicon Motion Capture System (Hou, 2013).

2.2 PERFORMANCE COMPARISON OF ATTITUDE FILTERS

Attitude and pose determination is the problem of how to estimate the orientation of a rigid body, for example an aircraft or a spacecraft, with respect to an inertial reference frame. One main method to estimate the orientation of an aircraft is through attitude filters or estimators. These filters statistically combine measurements from several sensors in order to achieve an estimate of the attitude. For decades, different attitude estimation algorithms have been created and developed. However, there is no true optimal attitude filter yet known in the literature (Zamani, 2003). Besides, different quadrotor platforms could favour different attitude filters. It is therefore valuable to undertake a performance comparison study that provides a suggestion for good attitude filters for real flight.

2.2.1 Orientation and Rotation Representations

In geometry, the term orientation of an object such as a line, plane or rigid body refers to the angular position or attitude with respect to a reference frame (Twiss and Moores, 1992). A conventional way to describe the orientation of a rigid body is to attach a frame to it. The orientation of the rigid body is physically meaningful only when a reference coordinate system is defined and the orientation of its axis is expressed relative to the reference frame.

A rotation is an operation that changes the orientation of an object from one to another. In other words, given a coordinate system, the orientation of a rigid body can be represented as a rotation from a reference orientation. Here we define two coordinate systems: the world coordinate system and the body-fixed coordinate system. The world coordinate system is fixed in inertial space and is denoted by {A}. The world coordinate system is also called the universe coordinate system/frame or the base frame in the literature. The body-fixed coordinate system is a moving coordinate frame that is rigidly attached to the aircraft whose attitude is to be described. The body-fixed coordinate system is denoted by {B}. Figure 2.8 below shows these two coordinate systems. The notation A_{RB}, or the term rotation, represents the relationship between these two coordinate systems, and is expressed in the form of rotation matrix.

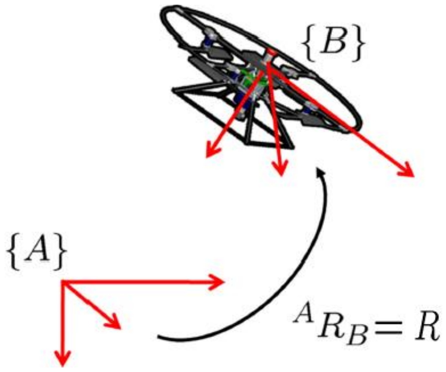


Figure 2.8: Two coordinate systems (Mahony, 2013).

There are many different ways to describe a rotation: rotation matrix, Euler angles, axis-angle, rotation vector and unit quaternion. Each rotation representation has its advantages and disadvantages and is used for different purposes (Diebel, 2006). It is possible to convert from one rotation representation to another. The rotation representation of IMU is Euler angle while that of the VICON system is unit quaternion. This thesis discusses three types of rotation representations which best match the hardware platforms and the four attitude estimation algorithms. These three rotation representations are rotation matrix, Euler angle and unit quaternion.

2.2.1.1 Rotation Matrix

A rotation matrix describes the relative orientation of the above two frames. It is commonly used to represent the kinematics of the attitude on the Lie group SO (3). The orientation of the rigid body is defined by a (3 × 3) rotation matrix (X_E, Y_E, Z_E) where the unit vectors X_E, Y_E and Z_E describe the axis of the rigid-body frame. Thus, a rotation matrix representing the orientation {B} relative to the frame of reference {A} is as follows.

$$ {}^{A}R_{B} = \left({}^{A}X_{B} \; {}^{A}Y_{B} \; {}^{A}Z_{B} \right) = \begin{pmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{pmatrix} $$

Formula 2.1: Rotation Matrix (Craig, 2005)

2.2.1.2 Euler Angles

The Euler angles are commonly used in aerospace engineering to represent the orientation of quadrotors. The Euler angles are represented as Roll-Pitch-Yaw angles, which are the rotation around x, y, z axis, respectively. Two main disadvantages of Euler angles are singularity and less accuracy than unit quaternion when they are used to integrate incremental changes in attitude over time. The conversion between rotation matrix and Euler angles is shown in Formula 2.2.

$$ R_{123}(\phi,\theta,\psi) = R_1(\phi)R_2(\theta)R_3(\psi) = \\ \begin{bmatrix} c_\theta c_\psi & c_\theta s_\psi & -s_\theta \\ s_\phi s_\theta c_\psi - c_\phi s_\psi & s_\phi s_\theta s_\psi + c_\phi c_\psi & c_\theta s_\phi \\ c_\phi s_\theta c_\psi + s_\phi s_\psi & c_\phi s_\theta s_\psi - s_\phi c_\psi & c_\theta c_\phi \end{bmatrix} $$

Formula 2.2: Conversion between Rotation Matrix and Euler Angles (Diebel, 2006).

The frame of the Mikrokopter has one bar being painted in red with the rest being in black. This is an approach to distinguish the four rotors. We define that the red bar is the front and the opposite is the rear. The coordinate system of the Mikrokopter is shown in Figure 2.9. Pitch angle represents the forwards and backwards movements while roll angle represents the left and right movements.

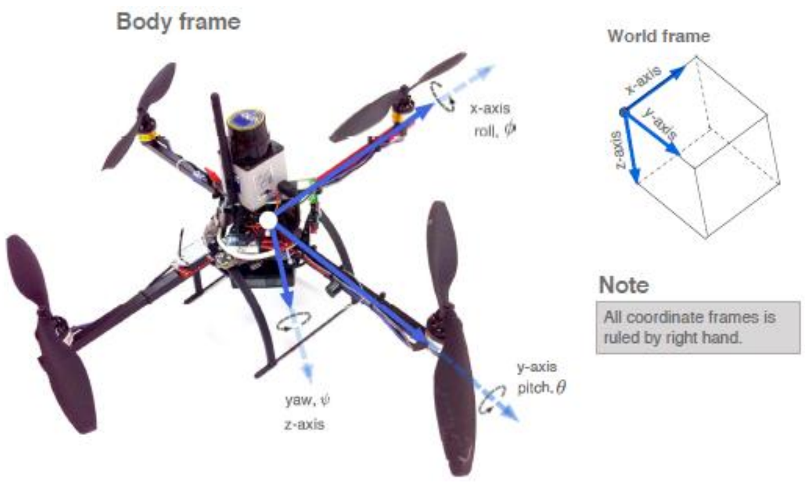


Figure 2.9: The coordinate system of Mikrokopter (Sa, I. 2011)

The IMU sensors measure the rotation around the axis x, y, and z with the data being expressed as a rotation matrix. These are determined in the body-fixed coordinates with respect to the inertial frame.

2.2.1.3 Unit Quaternion

Quaternions were first devised by William Rowan Hamilton, a 19th-century Irish mathematician (Diebel, 2006). Unit quaternions are quaternions with unity norm, a representation which can be used to represent the attitude of a rigid body. Unit quaternion has two main disadvantages. Firstly, the four quaternion parameters have less intuitive physical meanings. Secondly, the unity norm of the representation could bring problem in encoding the algorithms. The conversion between rotation matrix and unit quaternion is shown in Formula 2.3.

$$
\begin{aligned}
4q_0^2 &= 1 + r_{q11}(\mathbf{q}) + r_{q22}(\mathbf{q}) + r_{q33}(\mathbf{q}) \\
4q_1^2 &= 1 + r_{q11}(\mathbf{q}) - r_{q22}(\mathbf{q}) - r_{q33}(\mathbf{q}) \\
4q_2^2 &= 1 - r_{q11}(\mathbf{q}) + r_{q22}(\mathbf{q}) - r_{q33}(\mathbf{q}) \\
4q_3^2 &= 1 - r_{q11}(\mathbf{q}) - r_{q22}(\mathbf{q}) + r_{q33}(\mathbf{q})
\end{aligned}
$$

Formula 2.3: Conversion between Rotation Matrix and Unit Quaternion (Diebel, 2006)

The ground-truth measurements estimated by VICON system is expressed in unit quaternion. The measurements including the position and rotation of the rigid body are expressed in Earth-fixed coordinates in relation to the body-fixed frame.

2.2.2 Attitude Determination Methods

There are two main methods to determine attitude: deterministic solutions and estimation algorithms or filters. The former one is a point-by-point method based on two or more vector observations from a single point in time, while the latter one is a recursive stochastic method that statistically combines measurements from several sensors and predictions from dynamic and/or kinematic models as an approach to achieve an estimate of the attitude.

The attitude determination method being used in this research project is the estimation algorithms. Four attitude estimation algorithms are examined: Extended Kalman Filter (EKF) (Anderson and Moore, 1979), the Multiplicative Extended Kalman Filter (MEKF) (Markley, 2003), the Unscented Quaternion Estimator (USQUE) (Crassidis and Markley, 2003), and the Geometric Approximate Minimum-Energy (GAME) filter (Zamani, 2013). Detailed analysis and comparison of the above four attitude filters will be discussed in Chapter 3.2.

2.2.3 Performance Comparison

For decades, different attitude estimation algorithms have been created and developed. Existing filter studies involve theoretical and simulation comparisons (Zamani, 2003) where the former one is based on comparing the design properties of filters and the later one is based on comparing the simulation outcomes from a desired scenario. There is not much literature on performance comparison of the above four attitude estimation algorithms.

The performance comparison of attitude estimation algorithms is primarily assessed based on a real flight scenario. During the flight, the attitude of the object, represented through the rotation representations, is estimated as a result of the attitude filter on IMU or some data through Mikrokopter-based quadrotor. Meanwhile, the Vicon system estimates the quadrotor attitude as the ground-truth data. By comparing how far the attitude angle from filtered IMU data derives from the ground-truth attitude angle, we can compare the performance of the above four attitude estimation algorithms.

2.3 DATA SETS FOR ATTITUDE FILTERS

To compare the performance of different attitude filters and choose the filter with best performance for one particular type of quadrotor, a minimal benchmarking data set should contain ground truth information (a quaternion q provided by the VICON system) as well as angular velocity and vector directions provided by IMU measurements. Using the same data set for all filters ensures that filter comparison results vary only in the quality of filters not in other circumstances. This project could be expanded to further explore other auxiliary sensors such as audio and visual sensors for the benchmarking data set.

2.3.1 Sampling

Sampling is the process of recording the values of an analogue signal at certain time intervals (White and Clare, 1992). A sample refers to the values or measurements at a given point in time. There are two types of samples: analogue samples (which are continuous in both time and amplitude), and digital signals (which are discrete in both time and amplitude). For digital signals, it involves the term sampling frequency, which refers to the number of signals taken during one second. Figure 2.10 below is a signal sampling representation. The continuous curve represents the analogue sample while the cross nodes on the curve represent the discrete samples. The conversion from the analogue sample to a digital one is through an A/D converter, which is discussed in detail in 2.3.2.

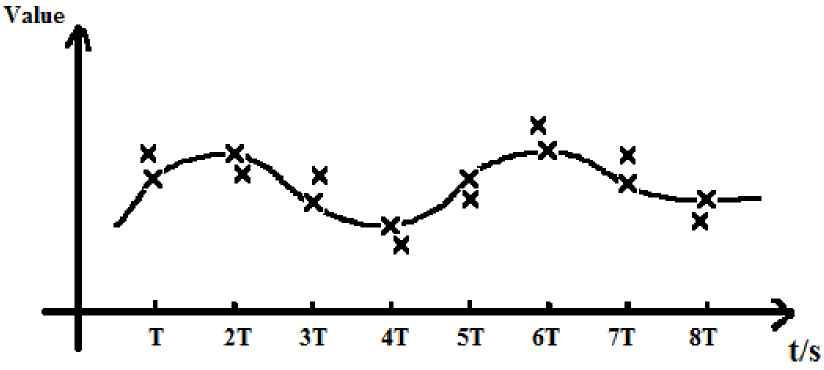


Figure 2.10: Sampling

Theoretically, the digital signals converted from the A/D converter attain the value of the analogue signals at given points in time. In reality, however, the measurements of the digital signals could be different from the analogue signals at that given points in time. This issue is visualised as the cross dots derivate from the analogue signal curve in Figure 2.10. This deviation effect can be explained as sampling noise, where a random fluctuation is observed in most mechanical, electrical and optical systems (White and Clare, 1992).

2.3.2 Analogue-to-Digital Converter

An analogue to digital converter (A/D converter) is the interface between analogue signals and digital signals where an analogue signal is changed into a digital signal. The general structure of an A/D converter is a two-step conversion model as shown in Figure 2.11. This structure first held the input voltage to the ADC at a constant level during conversion through the sample-and-hold circuit. It then quantises the signal from the first step in amplitude and converts it to digital zeros and ones.

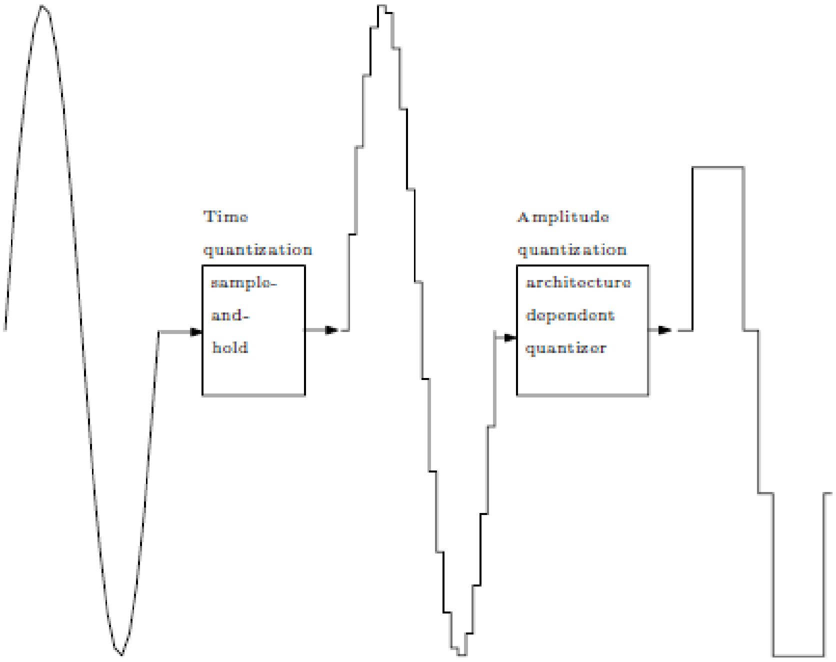


Figure 2.11: Two Step Conversion from Analogue to Digital Signal (Elbornsson, 2003).

The A/D converters can be implemented in many different ways, thus there are many structures of A/D converter. The structure being used in the ATmega644V microcontroller for the Mikrokopter platform is a successive approximation A/D converter (SA-ADC). The SA-ADC has two main components: a resistance ladder and a comparator, shown in Figure 2.12. After a voltage is applied over the resistance ladder, the digital value is attained by comparing the analogue voltage, u_{sh}, to the voltages between the resistances in the resistance ladder, u_{ri}, until a correct level is found. A *n*-bit SA-ADC represents that n comparisons are needed in the structure.

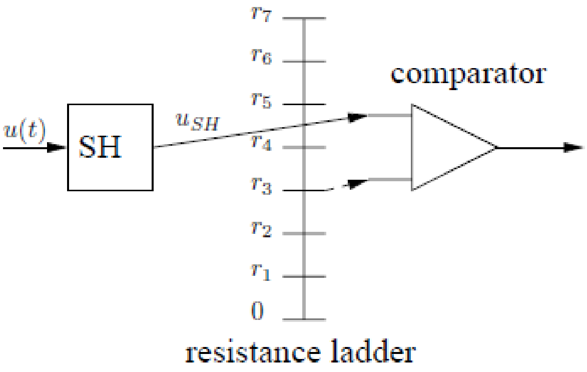


Figure 2.12: Successive Approximation ADC (Elbornsson, 2003).

The advantages of the SA-ADC are the small amount of hardware and low power consumption. The disadvantage of this A/D converter structure is that the conversion time is long due to the fact that the conversion is done in several steps.

2.4 CHAPTER SUMMARY

This chapter provides necessary background for this research project which is the design and collection of a benchmarking data set for attitude and pose estimators in UAV applications. It first analyses UAV applications in general and the Mikrokopter-based quadrotor. It then discusses the attitude and pose estimation through three orientation representations and the definition of the performance comparison. Finally, this chapter examines the characteristics of the benchmarking data on sampling and A/D converter.

Chapter 3 Comparison of Filters and Sensor Data

This chapter analyses four attitude estimation algorithms, as well as two sensor groups. In Chapter 3.1, a research literature in the area of attitude filters is given. Chapter 3.2 details the input characteristics of attitude and pose estimators, the information of which will be covered in the benchmarking data sets. An analysis of sensor data among IMU and VICON is addressed in Chapter 3.3. Lastly, Chapter 3.4 exams the issues where pre-processing the sensor data are needed.

3.1 EXISTING LITERATURE

Four attitude estimation algorithms have been explored in this project. They are the Extended Kalman Filter (EKF), the Multiplicative Extended Kalman Filter (MEKF), the Unscented Quaternion Estimator (USQUE), and the Geometric Approximate Minimum-Energy (GAME) filter.

Existing filter studies involve theoretical comparison and simulation comparison. Theoretical comparison was based on the properties of filters. Lefferts, Markley and Shuster (1982) comprehensively reviewed the methods of the Kalman filter in attitude estimation with the attitude being represented by the quaternion. Julier, Uhlmann and Durrant-Whyte (1995) first proposed the unscented estimator (UF) which was used to estimate the mean and covariance of the state vector. Crassidis and Markley (2003) then derived the USQUE which was a spacecraft attitude estimation approach based on the UF that used unit quaternion to represent the attitude. Crassidis, Markley and Cheng (2007) evaluated the basic assumptions of the first three filters with mathematical processes. This paper lastly discussed the strengths and weaknesses of different filters with a conclusion that the EKF, especially the MEKF, could incorporate different measurements flexibly. The USQUE, however, operated well when the dynamics or measurement models were highly nonlinear, or the state lacked a good priori estimate. Mortensen (1968) introduced a systematic approach to deriving filtering algorithms based on the properties of the value function of the optimal filtering problem. Lastly, Zamani (2013) in his PhD thesis proposed the GAME filter including its performance which is based on Mortensen's deterministic minimum-energy filtering approach.

Among the above four attitude estimation algorithms, some approaches maintained the basic structure of the extended Kalman filter, but employed various modifications providing better convergence or improvement of other performance characteristics. The aim of filter analysis is to prepare adequate information into the designed benchmarking data set through identifying the required inputs and outputs of different filters and mapping sensor data to filters.

3.2 COMPARISON OF FILTERS

A typical attitude estimator involves a two-part process (Crassidis et al., 2007): 1) estimate the orientation of a rigid body from on-board sensor measurements and know reference observations, and 2) filter the noisy measurements.

The input characteristics used by different attitude and pose estimators can be divided into three groups. They are measurements, initial data and parameters. The measurement inputs for the EKF, the MEKF, the USQUE, and the GAME filter are the same while the inputs of initial data and parameters for these four filters are different. The subsections below provide more details on the filter input characteristics. The notations follow the literature of Zamani (2013).

3.2.1 Measurement Inputs

The measurement inputs are the updated data collected through sensor estimation. The measurement inputs for the EKF, the MEKF, the USQUE, and the GAME filter are angular velocity of the quadrotor $\omega_{m,k}$ and its orientation represented by at least two directions $a_{m,k}$. The angular velocity is provided by the gyroscopes. The vector directions are provided by accelerometers and auxiliary sensors such as magnetometers and camera sensors. The above two measurement inputs, which are angular velocity and vector directions can be written as $u_k = \begin{bmatrix} \omega_{m,k} \\ a_{m,k} \end{bmatrix}$. Both of these two measurements are estimated in body-fixed frame.

3.2.2 Initial Data Inputs

Since the attitude filters are recursive stochastic methods that statistically combine measurements from several sensors and predictions from dynamic and/or kinematic models, the data for the initial state should be assigned. The initial information including initial filter gain matrixes is the initial data inputs required for different filters. All the above four attitude filters require initial attitude quaternion $\hat{q}(0)$, gyro-bias estimates $\hat{b}(0)$ and reference vectors $\mathbf{P}(\mathring{y}_i)$. The EKF, the MEKF, and the GAME filter have three initial filter gain matrixes $p_a(0)$, $p_b(0)$, and $p_c(0)$ belong to $R^{3\times3}$, while the USQUE has an initial 6×6 covariance p(0), The notation hat means the estimated value.

3.2.3 Parameter Inputs

The parameter inputs represent the characteristics of attitude filters. All the above four filters require gyro measurements Q_Ω (or B_Ω), bias model Q_b (or B_v), and time step d_t. The USQUE also has parameter inputs a, f, and λ , δ_k.

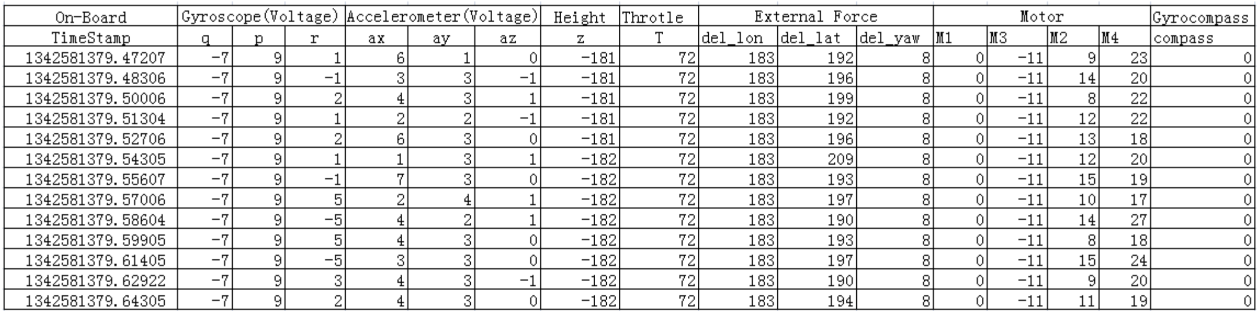
3.3 SENSOR DATA

The sensor data is collected within the ANU Vision Robotics group. The IMU data is collected through Mikrokopter on-board sensors with respect to world coordinate system expressed in body-fixed coordinates. Meanwhile the VICON system collects data in the world coordinate as a ground truth with respect to body-fixed coordinates expressed in world coordinates. Both the IMU data and the VICON data are a series of value at given points in time.

3.3.1 Inertial Measurement Unit (IMU)

The Mikrokopter on-board sensor data contains the following information: time stamp, angular velocities on roll, pitch, and yaw axis, linear acceleration on x, y, z axis, height, throttle, external force inputs on roll, pitch, and yaw axis, motor and gyrocompass information. The collected Mikrokopter data provide neither SI units nor documents explaining the units of the on-board sensors. Thus, these data are unscaled measurements. The collected sensor data is saved in text document as default and is shown in Table 3.1.

Table 3.1: Mikrokopter on-board sensor data.



On-Board	Gyroscope(Voltage)			Accelerometer(Voltage)			Height	Throtle	External Force			Motor				Gyrocompass
TimeStamp	q	p	r	ax	ay	az	z	T	del_lon	del_lat	del_yaw	M1	M3	M2	M4	compass
1342581379.47207	-7	9	1	6	1	0	-181	72	183	192	8	0	-11	9	23	0
1342581379.48306	-7	9	-1	3	3	-1	-181	72	183	196	8	0	-11	14	20	0
1342581379.50006	-7	9	2	4	3	1	-181	72	183	199	8	0	-11	8	22	0
1342581379.51304	-7	9	1	2	2	-1	-181	72	183	192	8	0	-11	12	22	0
1342581379.52706	-7	9	2	6	3	0	-181	72	183	196	8	0	-11	13	18	0
1342581379.54305	-7	9	1	1	3	1	-182	72	183	209	8	0	-11	12	20	0
1342581379.55607	-7	9	-1	7	3	0	-182	72	183	193	8	0	-11	15	19	0
1342581379.57006	-7	9	5	2	4	1	-182	72	183	197	8	0	-11	10	17	0
1342581379.58604	-7	9	-5	4	2	1	-182	72	183	190	8	0	-11	14	27	0
1342581379.59905	-7	9	5	4	3	0	-182	72	183	193	8	0	-11	8	18	0
1342581379.61405	-7	9	-5	3	3	0	-182	72	183	197	8	0	-11	15	24	0
1342581379.62922	-7	9	3	4	3	-1	-182	72	183	190	8	0	-11	9	20	0
1342581379.64305	-7	9	2	4	3	0	-182	72	183	194	8	0	-11	11	19	0

As mentioned in Chapter 2.1.2, the IMU sensor group measures instantaneous angular velocity and linear acceleration of the rigid body. If we plot the IMU data over time, it would have a 2-dimentional configuration with the x-axis indicating the passage of time and y-axis representing the sensor measurements at each recorded time. Figure 3.1 and Figure 3.2 below visualise the three-axis angular velocity and linear acceleration of the Mikrokopter in the real flight scenario over time. Sa (2011) compared the measurements between Mikrokopter IMU and external sensors and gave a suggestion for the Mikrokopter IMU that **Second** is a unit of time, **Radian/s** is a unit of all angular velocity, and **m/s^2** is a unit of all linear acceleration.

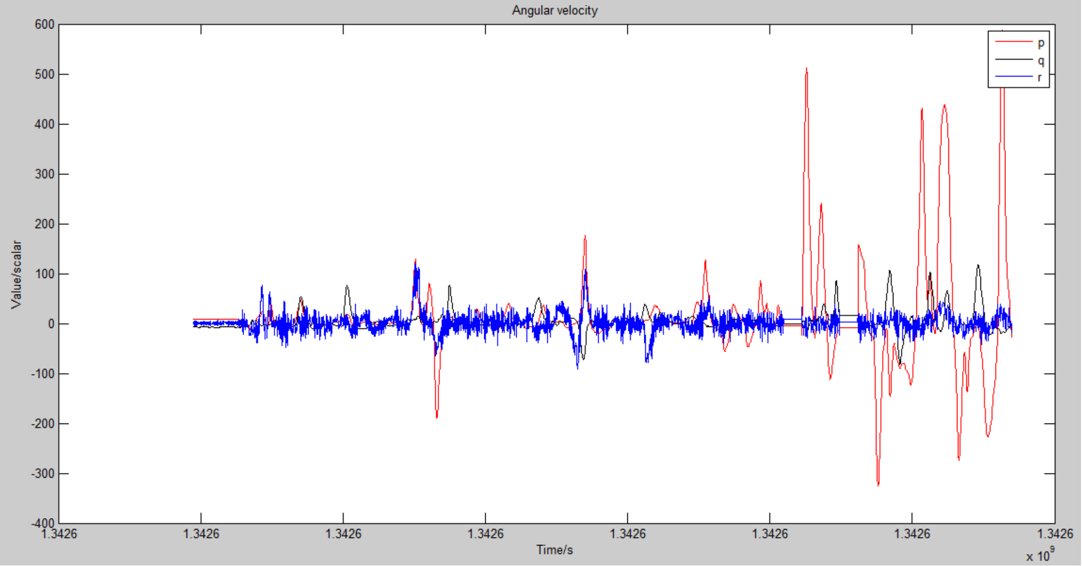


Figure 3.1: Gyroscope sensor data.

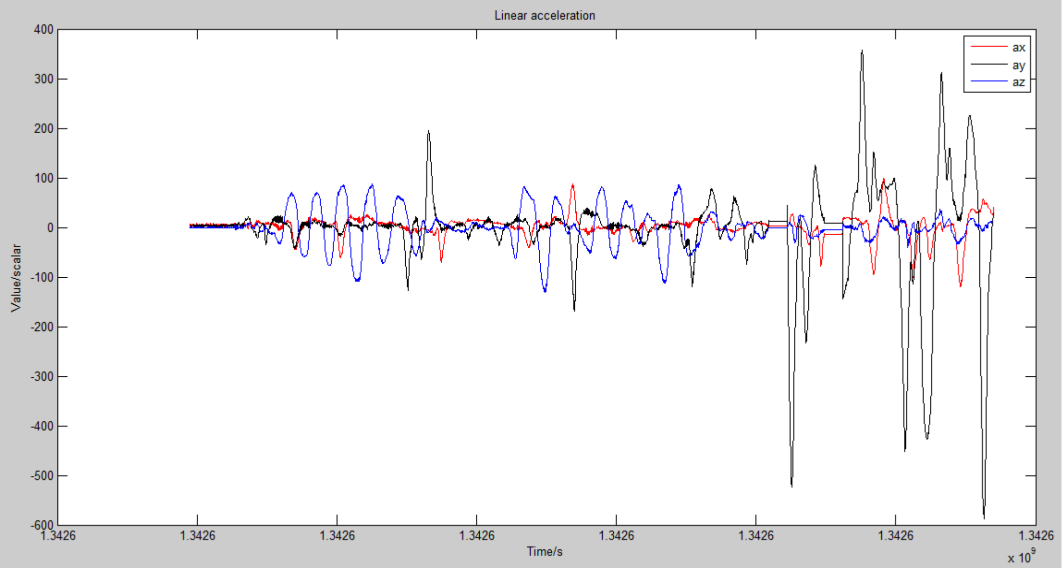


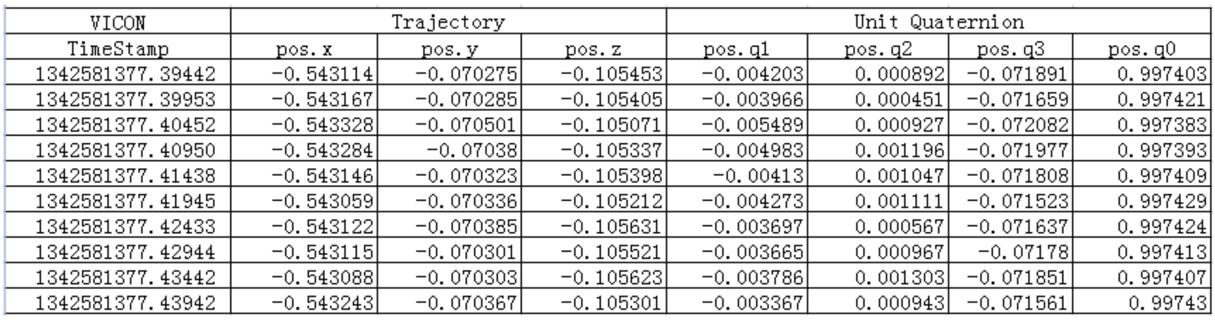
Figure 3.2: Accelerometer sensor data.

Experimental measurements are never perfect. As can be seen from the none-smooth curves of the gyroscope and the accelerometer data, the sensor measurements are noisy. In less expensive, small and light-weight inertial systems, the inertial sensors produce estimation errors, as well as noise which are random unwanted data without meaning. As time increases, the superimposed sensor drift and noise tend to grow unbounded. These sensor measurements will be used as the measurement inputs for the attitude filters.

3.3.2 Vicon Data

The VICON data contains the following information: time stamp, the quadrotor position represented in x, y, and z axis, and unit quaternion values q0, q1, q2, and q3. Similar to the Mikrokopter on-board sensor data, the collected VICON data is shown in Table 3.2.

Table 3.2: VICON sensor data.



VICON	Trajectory			Unit Quaternion			
TimeStamp	pos.x	pos.y	pos.z	pos.q1	pos.q2	pos.q3	pos.q0
1342581377.39442	-0.543114	-0.070275	-0.105453	-0.004203	0.000892	-0.071891	0.997403
1342581377.39953	-0.543167	-0.070285	-0.105405	-0.003966	0.000451	-0.071659	0.997421
1342581377.40452	-0.543328	-0.070501	-0.105071	-0.005489	0.000927	-0.072082	0.997383
1342581377.40950	-0.543284	-0.07038	-0.105337	-0.004983	0.001196	-0.071977	0.997393
1342581377.41438	-0.543146	-0.070323	-0.105398	-0.00413	0.001047	-0.071808	0.997409
1342581377.41945	-0.543059	-0.070336	-0.105212	-0.004273	0.001111	-0.071523	0.997429
1342581377.42433	-0.543122	-0.070385	-0.105631	-0.003697	0.000567	-0.071637	0.997424
1342581377.42944	-0.543115	-0.070301	-0.105521	-0.003665	0.000967	-0.07178	0.997413
1342581377.43442	-0.543088	-0.070303	-0.105623	-0.003786	0.001303	-0.071851	0.997407
1342581377.43942	-0.543243	-0.070367	-0.105301	-0.003367	0.000943	-0.071561	0.99743

If we plot the VICON trajectory over time, it would have a 3-dimentional configuration that represents the Mikrokopter position movement over time. If we transfer the VICON unit quaternion measurements into Euler angle and plot it over time, it would have a 2-dimentional configuration that represents the Mikrokopter attitude movement over time. Figure 3.3 and Figure 3.4 below visualise the position and attitude of the Milrokopter-based quadrotor. The unit of Figure 3.3 is meter and that of Figure 3.4 is second for x-axis and degree for y-axis.

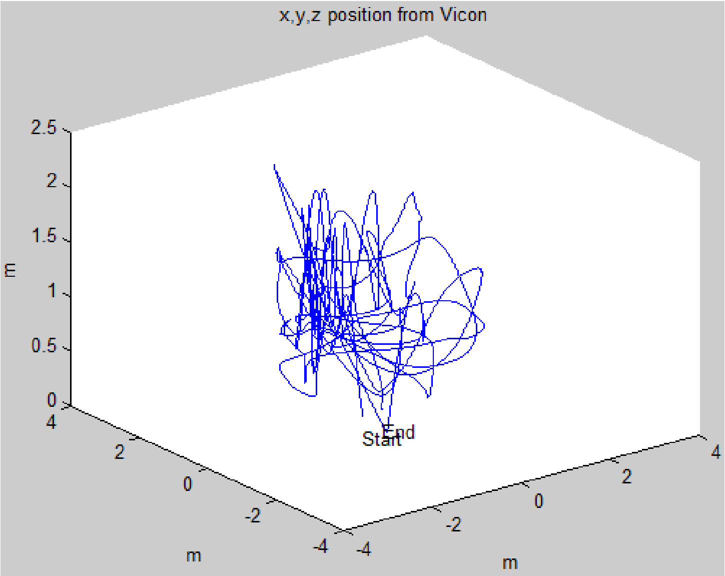


Figure 3.3: VICON data on Mikrokopter trajectory.

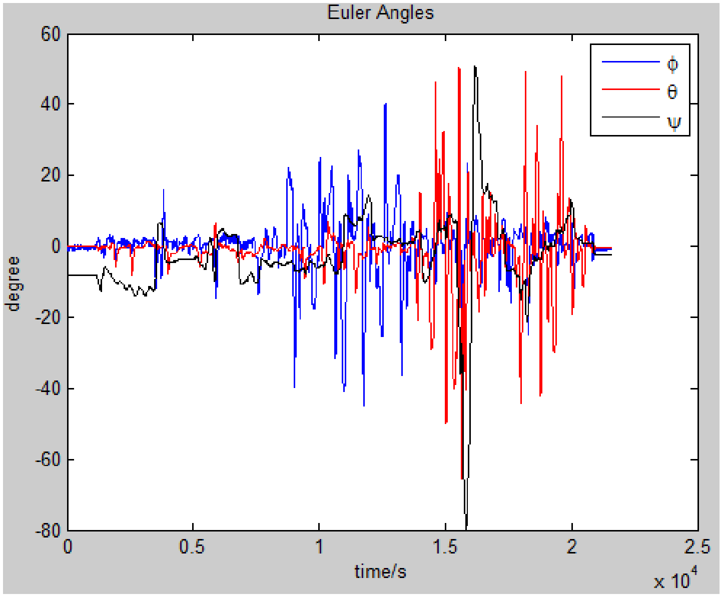


Figure 3.4: VICON data on Mikrokopter attitude.

The VICON measurements are used as the ground truth. The suggestion for the performance comparison of different attitude filters would be to compare the filtered roll-pitch-yaw angles estimated after implementing attitude filters and the ground truth attitude angles.

3.4 ISSUES IDENTIFIED

The data provided by the inertial sensors needs to be accurate if we want to compare IMU measurements with its corresponded ground truth values given by the VICON system measurements. Thus, an initial sensor data pre-processing for the measurement inputs of filters is needed with the following two issues being identified. Firstly, the value or the amplitude of the sensor measurements should be accurate. Recall the sampling image Figure 2.10 in Chapter 2.3.1, that is to say, the value of the discrete sample at a given point in time is correct and lies on the measurement curve. Secondly, the recorded time stamps should be correct. That is to say, the time intervals of successive discrete times are the same and the time stamps when the measurements are recorded in the data sheet represent the time when the measurement object is measured.

Due to the hardware limitations, in less expensive, small and light-weight inertial systems, many factors contribute to degradation of the inertial sensors measurements. In order to collect and design a benchmarking data set for attitude and pose estimators in the Mikrokopter platform, it is necessary to pre-process the initial sensor data for inputs of filters. The subsections below discuss the issues of accuracy, calibration and time synchronisation.

3.4.1 Accuracy

The accuracy of sensors is its ability of a measurement deriving from the actual value of the quantity being measured (Haby, n.d.). Possible sources of errors related to the accuracy of IMU are sensitivity, axis alignment, nonlinearity, thermal effect, sensor tolerance, response time, and A/D converters.

Sensitivity means how much input parameter change is required to produce a standardized output change. For example, sensitivity describes the information of whether the sensor is able to distinguish 64 units and 64.001 units. The axis alignment means the alignment between the three perpendicular axis x-y-z could not be exactly 90 degrees. Thus, the accuracy of sensor data is affected. The linearity expresses the sensor feature that how much the actual measured curve derives from the ideal curve. Thermal variations could cause the misalignment of thermo-mechanical between the attitude sensors, IMU, and payload (Aerospace Control Systems, n.d.), and thus cause inaccuracy. Sensor tolerance is the range when the sensor measurement is trustable. Since the sensor do not change its output state immediately after an input parameter changes, there involves sensor response time which could affect the sensor accuracy. The A/D converter also involves the accuracy issue when (Elbornsson, 2003) it converts a signal from analogue to digital.

3.4.2 Calibration

Calibration is the process to compensate the above sensor accuracy errors. The accuracy and calibration issues are outside the project scope. This chapter below examines different calibration approaches in the research literature and should provide a suggestion for the calibration of the sensor accuracy issue.

Titterton and Weston (1997) pointed out that the sensor data measured by accelerometers and gyroscopes can be numerically integrated based on the initial values of the attitude, velocity and position to get the estimated navigation states. It is evident that bias, scale factor and non-orthogonality errors are the dominant deterministic elements in IMU (Syed, Aggarwal, Goodall, Niu and El-Sheimy, 2007). The less the errors, the less position drift of the navigation system. In order to provide a particular data set for performance comparison, a calibration of sensor errors is needed. Syed et al. (2007) described a calibration method on inertial navigation system (INS), which can be used to determine the bias and scale factors of the sensors, but cannot estimate the axis misalignments (non-orthogonalities). Artese and Trecroci (2008) illustrated a multi-position calibration method, which doesn’t require the precise aligned mounting of the sensors with either local level frame or to the vertical direction. Li, Niu, Zhang, Zhang and Shi (2012) described a calibration method in situ hand, which is able to calibrate the bias and the scale factors of both accelerometer and gyroscope. Instead of specific equipment or device, this method required only motions by hands.

3.4.3 Time Synchronisation

Time synchronisation is the term to represent the sensor measurements having accurate time. The benchmarking data sets contain two main measurement groups: IMU measurements and the ground-truth measurements. In reality, however, there are two things which need to be addressed: 1) the IMU never run at the same frequency, thus there is a time offset (the time offset could be zero) for each node of IMU data between the estimated time and the time we assumed. Thus, the output value could be the neighbour point of the time we read with a bigger or smaller value than the value at that time; 2) since the Vicon system has its own time clock instead of onboard time clock, the start time of the Vicon data and the IMU data need to be synchronised. As a consequence, the time synchronisation issue involves the following two considerations. Firstly, the time series when the IMU record measurements is accurate. Secondly, the time series when the IMU measurements and the ground-truth measurements are well aligned. Since the sensor data is expressed at a given point in time, if the time when the data is collected is not trustable, the performance comparison study is meaningless. The time error in the system could be clock difference and data transmission latency issues, oversampled and undersampled messages, jitter, and irregular sampling problems. The above time synchronisation problem will be discussed in detail in Chapter 4.

3.5 CHAPTER SUMMARY

This chapter discusses the benchmarking data set in the area of four attitude filters and two sensor groups. It first analyses the input characteristics of four attitude estimation algorithms. Three types of input characteristics, which are measurement inputs, initial data inputs, and parameter inputs, are analysed based on the research literature. It then discusses the format and semantics of the collected sensor measurements. Finally, this chapter examines the issues identified as a need for sensor pre-processing. Although both the accuracy and the time synchronisation issue need to be addressed for a benchmarking data set, this thesis only focuses on the time synchronisation issue.

Chapter 4 Time synchronisation problem

This chapter analyses the time synchronisation issue for sensor data pre-processing. In Chapter 4.1, background information of time synchronisation and definition of two clock systems being used in this research project are provided. A research literature in the time synchronisation area is given in Chapter 4.2. The time synchronisation issue related to the clock problems and the sampling problems are further discussed in Chapter 4.3 and Chapter 4.4, respectively.

4.1 INTRODUCTION

The benchmarking data sets contain two main measurement groups. They are ground truth information (a quaternion q provided by the VICON system) as well as angular velocity and vector directions provided by IMU measurements. The above two measurement groups may record measurement events with time series, i.e. values collected at successive discrete times. The cycle counter is typically used for time stamping each event before recording it in a per CPU buffer. In other words, the time stamps in the above time series are the recorded times when the Analogue-to-Digital (A/D in short) Converter sends the measurements back to the CPU. The measurement process involves two separate clocks. We will refer to the timestamps generated by the A/D converter in the Microcontroller ATmega644/V as the IMU clock. Similarly, we will refer to the timestamps generated by the Host PC in the Vicon MX architecture as the VICON clock. A third clock to consider in conjunction with the IMU clock and the VICON clock is a hypothetical real-time clock, that may or may not correspond to an actual clock and keeps track of the current physical time as a time baseline.

The comparison between IMU measurements and ground truth values given by the VICON system measurements is only meaningful when the IMU clock and the VICON clock are properly aligned. This could for example be achieved by synchronising both time clocks with the real-time clock; however, this approach is usually not feasible in small scale robotic systems. There was not any evidence to suggest that there would be performance issues with existing controllers associated with the above clocks being unsynchronised. A time synchronisation problem should be addressed, however, since this research project needs to provide a benchmarking data set for attitude and pose estimators in UAV applications. This is because poorly synchronised data could (dis)advantage particular filter algorithms[1]

4.2 EXISTING LITERATURE

To understand the time synchronisation problem, existing literature has been explored on aspects of clock difference and data transmission latency issues, oversampled and undersampled messages, jitter, and irregular sampling problems. The following list of papers is by no means exhaustive, the purpose of this review was to learn about time synchronisation and to identify potential issues, rather than to provide an overview of existing results. In terms of solving the clock difference and data transmission latency problems, an advantages and disadvantages analysis of different time synchronisation scenarios and existing solutions, as well as an accuracy calculation method was explored for integrating GPS and an INS system in (Ding, W., Wang, J., Li, Y.,Mumford, P., and Rizos, C., 2008). Regarding irregularities of sampling messages, a theoretical explaination of oversampled and undersampled messages, and jitter on clocks between the flight computer and IMU clocks was provided in (O'Shaughnessy, D. J., Vaughan, R. M., Haley, D.R., and Shapiro, H. S., 2006). The authors then explained the reasons for clock drift in detail with a possible solution to calculate and eliminate the effect of clock drift. Irregular sampling problems were further explored by Eng, F. (2007). The author observes that the sampling times are most often taken to be equally spaced in the statistical signal processing. The author then provided tools to deal with errors induced by non-uniform sampling.

Generally, time synchronisation problems can be divided into two categories, namely clock problems and sampling problems. The former problem type refers to the case where the reading the clock gave cannot be trusted, and the solution is to find the hypothetical real time, while the latter problem type refers to the case where the clock is reliable but the sampling frequency is irregular. Both problem types can occur simultaneously.

The possible sources of time synchronisation problems are as follows. We stipulate that each subsection below contains the discussion of one single problem instead of a mixture of

[1] Poorly synchronised data could disadvantage an attitude filter if the filter has a good outcome when the data is well synchronised and a bad outcome when the data is poorly synchronised. Whether poorly synchronised data would disadvantage and/or advantage an attitude filter depends on the nature of the algorithm. We can identify this issue if we have a well synchronised data set and compare the performance of filter algorithms with that of the poorly synchronised data.

problems. In other words, each discussion section assumes that the time series are perfect except for the issue being discussed.

4.3 CLOCK PROBLEMS

The most important clock problem is clock misalignment, i.e. the IMU clock differs from the VICON clock. As illustrated above, these two clocks are generated by two different systems providing separate series of timestamps. Whether these two clocks are well aligned is unknown even though both systems operate at the same time with timestamps being recorded. The possible reasons for clock misalignment are as follows.

4.3.1 Starting Time Difference between Two Clocks

The IMU clock and VICON clock may not power up and start counting exactly at the same time, resulting in a time difference between these two. In other words, a few IMU timestamps might be recorded before/after the collection of VICON timestamps. A visual representation of the above is shown in Figure 4.1. We will use the notation $\triangle$t to represent the starting time difference between the IMU clock and the VICON clock. A mathematical representation is:

$t_{IMU} - t_{VICON} = \triangle t$, where $\triangle$t could be positive, negative or zero.

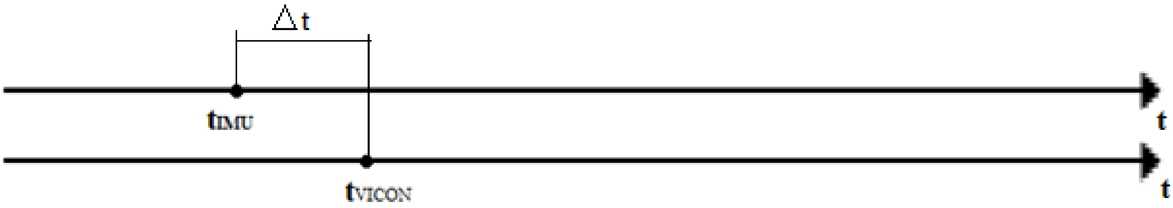


Figure 4.1: Start time misalignment.

4.3.2 Scale Difference

The IMU clock and the VICON clock may have scale differences. This has to do with choice of time unit, e.g. seconds vs. milliseconds, as designed by the designer. Thus, the same ordinal on different clocks may not indicate the same read, thus these two clocks are not aligned. A visual representation is shown in Figure 4.2. The black nodes on the IMU clock and the VICON clock indicate the recorded timestamps. As can be seen, the unit of the IMU clock is second while that of the VICON clock is millisecond. Thus, the tenth node of these two clocks indicates 10 second and 0.01 second, separately. If we use notation *i* and *j* to represent IMU and VICON clock scaling factors over 1 second, and notation N_{IMU} and N_{VICON} to represent the number of recorded timestamps for the two clocks, a mathematical description about the time difference between these two clocks can be written below:

$$\text{Time difference } \triangle t = i \times N_{IMU} - j \times N_{VICON}$$

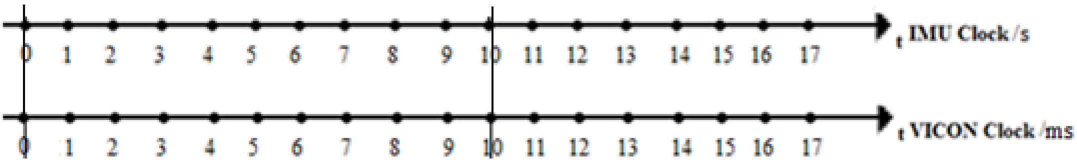


Figure 4.2: Clock scaling problem.

4.3.3 Clock Drifts

Clock drifts, also known as clock bias, is the offset between the clock and a very precise reference clock per unit of time (Marouani and Dagenais, 2008). Clock drifts will accumulate over time. The IMU and VICON clocks have drifts, meaning that the IMU clock (or the VICON clock, or both) does not run at the exact right speed compared to the real-time clock. The relative drift rate between clocks is small and is often expressed as a ratio in parts per million (Values of 1 ppm or 10^{-6} are typical for computer grade crystals). The major reason for clock drift appears to be thermally driven (Marouani and Dagenais, 2008). This is because the circuit temperature directly affects the crystal frequency and thus the drift. The IMU clock and the VICON clock are misaligned when at least one clock drifts away from the real-time clock. The pattern in Figure 4.3 explains the clock drift resulting in misalignment between the IMU clock and the VICON clock. For a small period of time, it is difficult to identify the clock drift. By looking at the accumulated time for a long interval (a thousand seconds), however, the IMU clock drifts obviously with respect to the VICON clock. If the drift rate is 50 ppm, the IMU clock loses 10 ms on the hypothetical real clock in 200 s, resulting in clock misalignment.

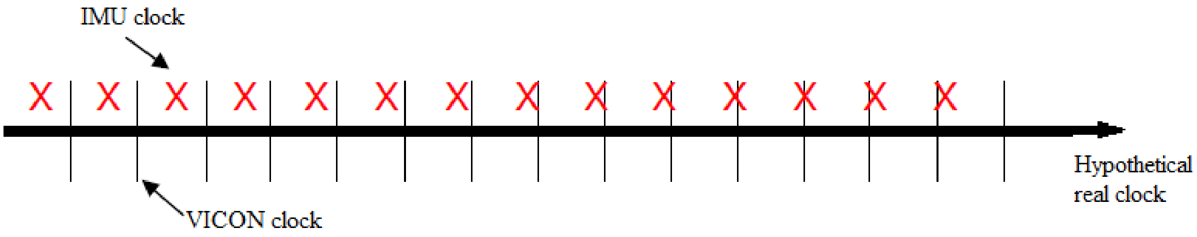


Figure 4.3: Misalignment between the IMU clock and the VICON clock.

4.3.4 Clock Jitter

Clock jitter, also known as clock noise and non-uniformity, causes small increases and decreases in the spacing between clock intervals, i.e. the spacing between consecutive IMU clock nodes (or VICON clock nodes, or both) is not constant. Figure 4.4 is a visual representation of clock jitter for the IMU clock. The dots on the upper time-line represent the IMU timestamps having clock jitter while that on the lower time-line represent the IMU timestamps without clock jitter. Since both clock jitter and clock drift effect the spacing between clock intervals, we have to observe a long period of time to differentiate between the two problems. Unlike clock drift, clock jitter will not accumulate over time. As a consequence, clock jitter appears to occur at random over a short period of time instead of the whole experiment.

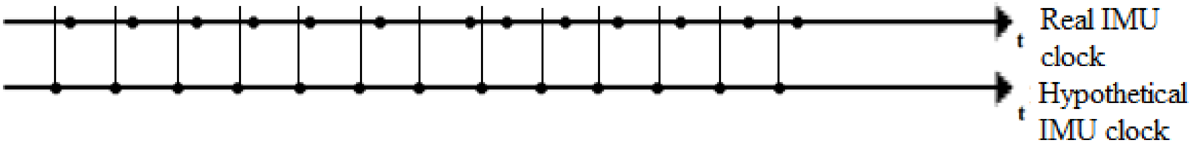


Figure 4.4: IMU clock drift.

4.4 SAMPLING PROBLEMS

Sampling problems occur during the process of recording the values of a signal at given points in time. It is expected that the points in time where the IMU and the VICON system record data are aligned. This includes the time between two adjacent points being equidistant for each set of data points. In reality, especially for the IMU sampling, however, the sampling frequency keeps changing around an average frequency to a small degree. The reasons for sampling problems are as follows.

4.4.1 Different Sampling Frequencies

The frequencies of the IMU recorded points (with the notation f_{IMU}) and the VICON recorded points (with the notation f_{VICON}) are very likely different. As a consequence, after aligning the time-line of the IMU clock and the VICON clock, an IMU timestamp would have no corresponding VICON timestamp. Figure 4.5 below visualises the above different sampling frequency issue. Here, the average IMU frequency is 64.04 Hz while that of the VICON system is 200 Hz. As can be seen, the second triggered node of the VICON clock has no aligned IMU node. And the second triggered node of the IMU clock has no direct relation with the second

VICON node. Thus the sampling information would be meaningless if we compare the above two sets of sampling without using nodes where they are aligned. Possible solutions to overcome this problem are hardware triggering and interpolation.

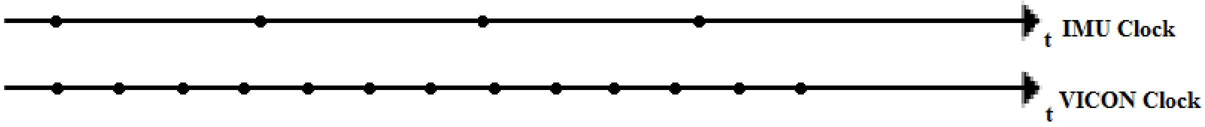


Figure 4.5: Clock frequency differences.

4.4.2 Sampling Irregularity Characteristic

Irregular sampling may occur due to the A/D converters. Figure 4.6 shows the transmission behaviour between a CPU and an A/D converter with respect to timing parameters on a time-line. The first node indicates the time when the CPU sends a command to the A/D converter requiring it to start work. This time is the reference reading of the CPU clock in a programming sense rather than the hypothetical real time. The second and the third nodes represent the times when the A/D converter starts and finishes executing the command, respectively. The characteristic of this time depends on the A/D converter where the completion time of each conversion may vary (Atmel, 2012). The forth and the fifth nodes represent the times when the CPU reads and records the time along with the measurements, respectively. Similar to the first node, these two times depend on programming and would normally be small time periods. Actually Figure 4.6 is a theoretical breakdown of the complete data transmission process. In reality, only the last node which is the time when the CPU writes values to memory can be recorded, while we cannot get access to the time of the other nodes. The sample frequency changes when any loop of data transmission processes varies with time. The biggest effect is the convertion completion times of the A/D converter.

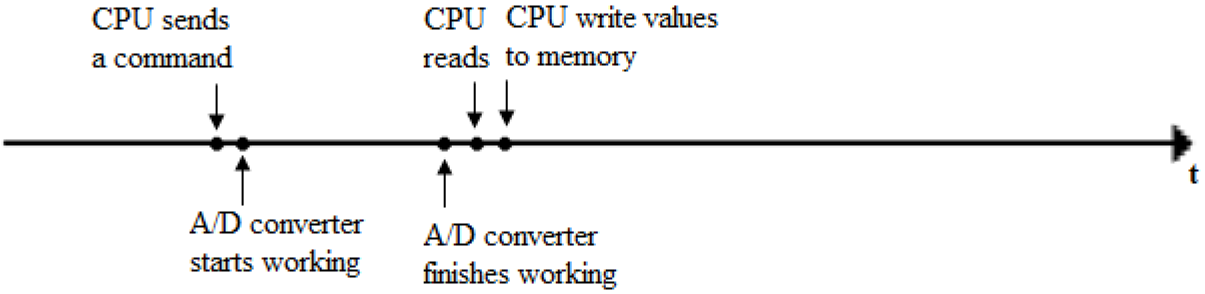


Figure 4.6: A/D converter sampling.

4.4.3 Sampling jitter

Sampling jitter is the deviation of the significant instances of a signal from their ideal location in time, i.e. jitter is how far a signal transition is from its reference, when it should transition. Jitter is caused by noise on the clock signal. One possible reason for jitter in our case is mismatch errors in time interleaved A/D converters (Elbornsson, 2003). For A/D converters, sampling is normally assumed to be periodic with a fixed period. In reality, however, a time error brings a deviation from the nominal sampling instance. This time error is insignificant when a single A/D converter is used. The above time error would cause irregular sampling intervals when interleaved A/D converters are used. This means there are small sampling increases and/or decreases in the consecutive IMU clock (or VICON clock, or both). This is because all the A/D converters are not identical due to the manufacturing process. A visualised representation is shown in Figure 4.7. Although this figure looks the same as before, it is a new time synchronisation issue. We will refer to the VICON clock being the ground truth. The IMU sampling time differs from its ideal location when sampling jitter occurs.

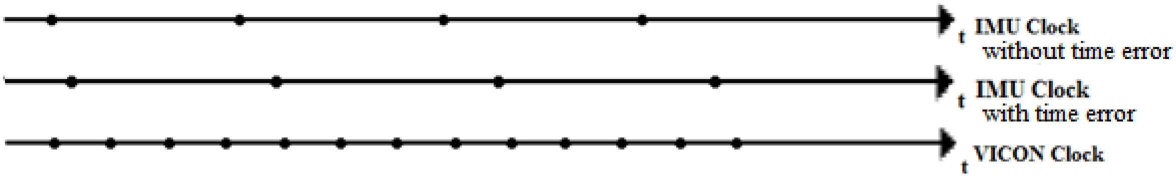


Figure 4.7: Clock frequency differences for individual A/D converter.

4.4.4 CPU Load

The overall system load has an effect on the sampling frequencies. Generally speaking, a CPU has priority in mission execution in order to ensure that important tasks, like quadrotor flying, are to be completed. A loaded system then spends more time in code sections where interrupts are temporarily disabled (Marouani and Dagenais, 2008). Therefore, high system load increases the interrupt latency, resulting in low accuracy of the measurement frequency.

4.5 CHAPTER SUMMARY

This chapter provides a general analysis of time synchronisation problem on two categories. In terms of clock problem, starting time difference between two clocks, scale difference, clock drift and clock jitter have been investigated. Regarding sampling problem, different sampling frequencies, sampling irregularity characteristic, sampling jitter and CPU load have been analysed. After understanding the above time synchronisation problem, we are able to solve the time synchronisation issue, which is presented in the next chapter.

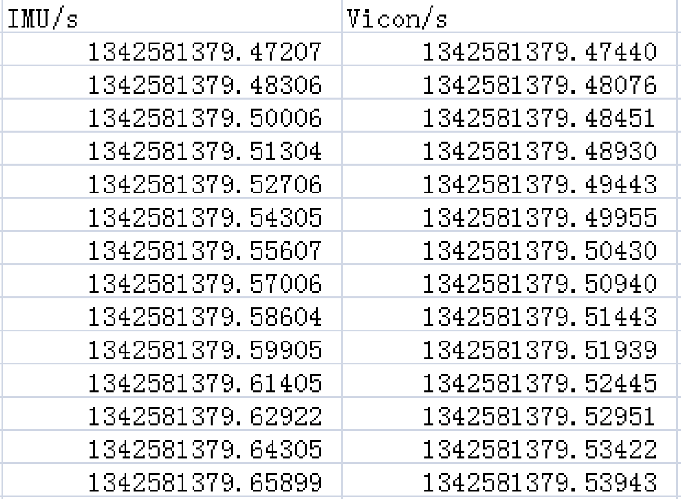
Chapter 5 Possible solutions to the time synchronisation problem

Chapter 4 discusses the general sources of time synchronisation problem with features for each source. This chapter first analyses which sources of time synchronisation problem need to be addressed for the Mikrokopter-based quadrotor. This chapter then provides general information about two ways, which are hardware and software approaches, to solve the problem. Finally, this chapter exams existing solutions and how it different with this research project and provides a possible solution to the time synchronisation issues.

5.1 Time Synchronisation Problem on Mikrokopter-based quadrotor

The sensor data is collected within the ANU Vision Robotics group. As mentioned above, the IMU time stamps are generated by the A/D converter in the Microcontroller ATmega644/V while the VICON time stamps are generated by the Host PC in the Vicon MX architecture. The time stamps of these two sensor groups are extracted from the original sensor data set which is shown in text format in Table 5.1 and in one dimensional zoomed visualisation in Figure 5.1. In Figure 5.1, the blue cross represents the IMU time stamps and the red dot represents the VICON time stamps. As can be seen, the starting time of these two clocks are different.

Table 5.1: The IMU and VICON time stamps from the original sensor data set



IMU/s	Vicon/s
1342581379.47207	1342581379.47440
1342581379.48306	1342581379.48076
1342581379.50006	1342581379.48451
1342581379.51304	1342581379.48930
1342581379.52706	1342581379.49443
1342581379.54305	1342581379.49955
1342581379.55607	1342581379.50430
1342581379.57006	1342581379.50940
1342581379.58604	1342581379.51443
1342581379.59905	1342581379.51939
1342581379.61405	1342581379.52445
1342581379.62922	1342581379.52951
1342581379.64305	1342581379.53422
1342581379.65899	1342581379.53943

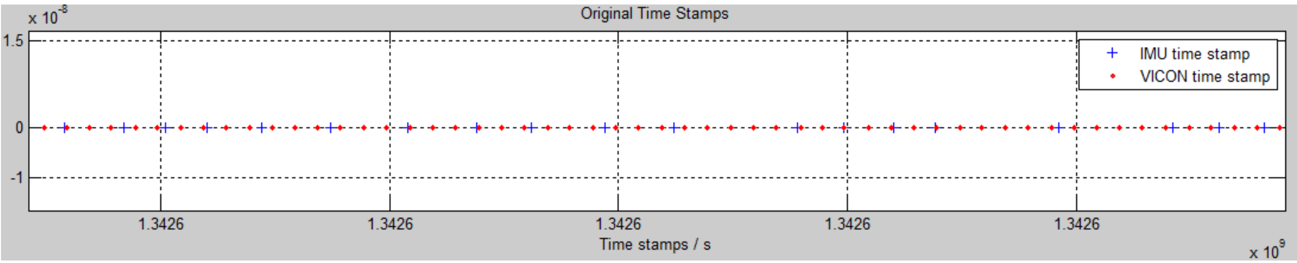


Figure 5.1: Original Time Stamps.

These two groups of time stamps are processed for analysis. Firstly, the average IMU frequency is 64.04 Hz while that of the VICON system is 200 Hz. These are calculated through the following formula.

$$\text{Sampling frequency} = \frac{\text{Total Time Elapse}}{\text{Number of Samples}}$$

As can be seen, the sampling frequencies of these two sensor groups are different. Figure 3 visualises the time stamp interval of the two sensor groups over the first 2916 samples. The blue line and the red line represent the IMU intervals and the VICON intervals, respectively. The two green lines represent the average time stamp intervals of the IMU (0.01562s) and the VICON (0.005s), respectively. As can be seen, the time stamp intervals of IMU are very unstable while that of the VICON are relatively stable.

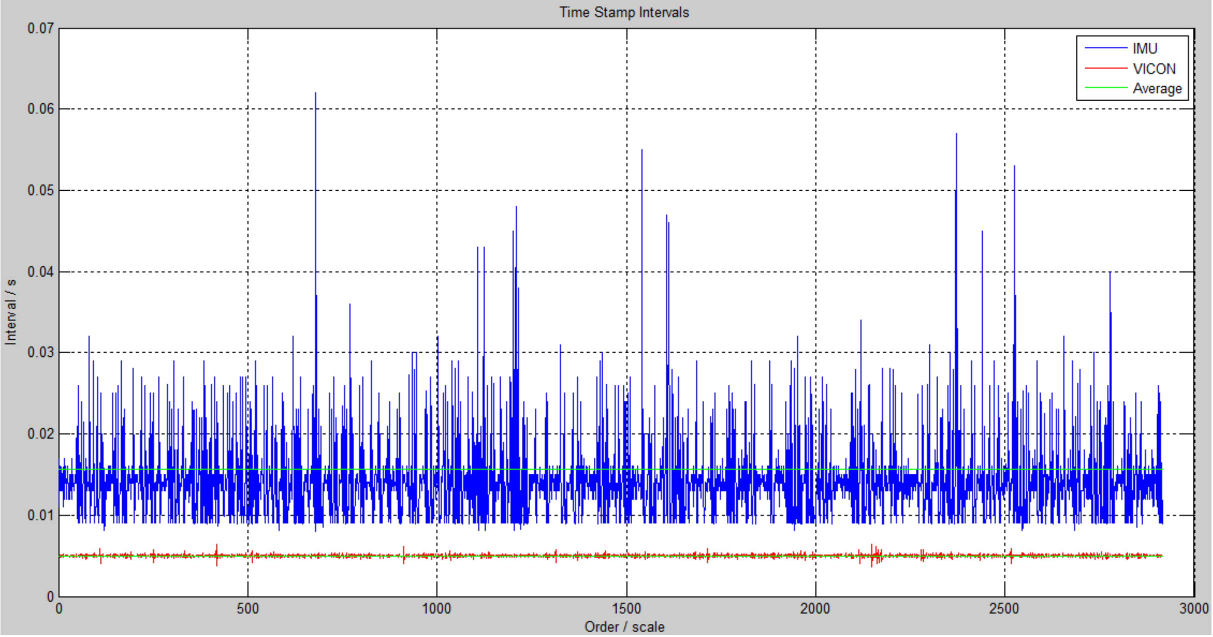


Figure 5.2: Time stamp intervals.

Since the IMU time sample intervals do not have a constant pattern, the issue of clock drifts is relatively small in this research scenario. Then we consider the above time stamps over a relatively long period of time to check if the IMU time stamps have clock jitter problem.

Theoretically, the number of IMU time stamps over 1 second is 64 times of a single sample, while that over 2 seconds is 128 times of a single sample. The number of IMU samples over 1 second and 2 seconds are shown in Figure 5.3 and Figure 5.4, respectively.

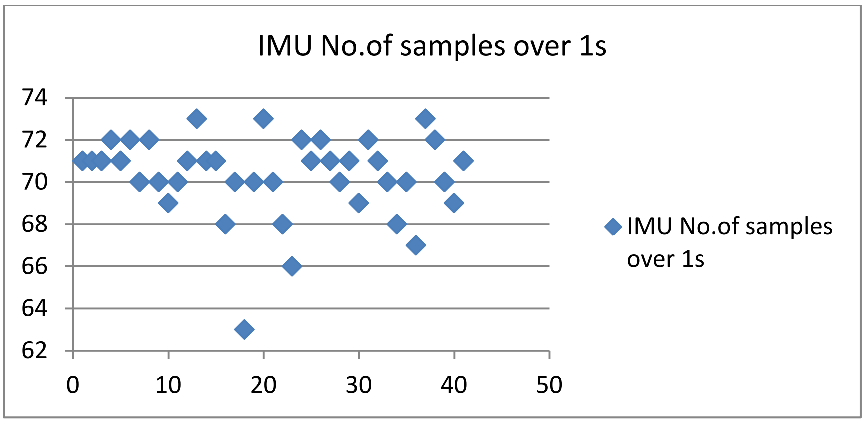


Figure 5.3: Number of IMU samples over 1s.

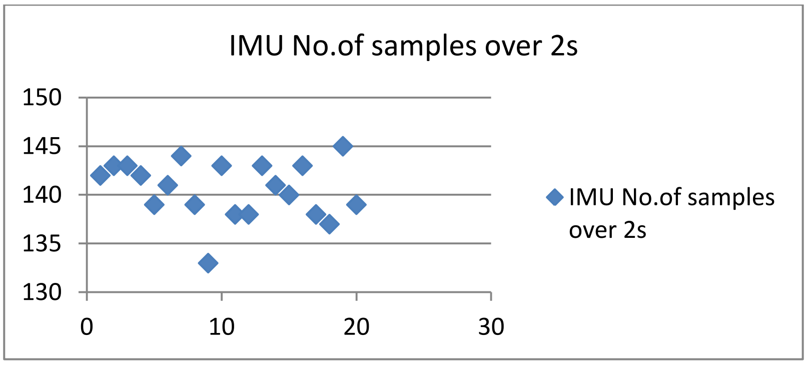


Figure 5.4: Number of IMU samples over 2s.

As can be seen, both the number of IMU samples over 1s and 2s are not regular over the whole experiment period. Thus, the effect of clock jitter is relatively small compared with other issues.

Next the effect of A/D converters will be examined. The characteristics of the A/D converter in the Microcontroller ATmega644/V are analysed below. Based on the design features, the minimum input clock frequency of successive approximation circuitry is 50 kHz, which requires the maximum response time. The maximum response time for a conversion takes 25 ADC clock cycles in order to initialize the analogue circuitry. Thus, the maximum response time for an analogue-digital conversion is:

$$\frac{1}{50\text{kHz}} \times 25 \text{ ADC clock cycles} = 0.5\text{ms}$$

Since the average IMU frequency is 64.04 Hz, the maximum tolerance for one conversion is:

$$\frac{0.5\text{ms}}{1/60.04\text{Hz}} \approx 3\%$$

Recall the information shown in Figure 4.6 and the discussion in Chapter 4.4.2. Because only the last node which is the time when the CPU writes values to memory can be recorded, thus, the above tolerance the A/D converter has in the Mikrokopter platform has an impact on the time synchronisation issue.

Based on the above analysis, the time synchronisation problem for the Mikrokopter-based quadrotor is the mixture of the above sources. It includes starting time difference between two clocks, different sampling frequencies, sampling irregularity characteristic, sampling jitter, and CPU load.

5.2 GENERAL APPROACHES TO SOLVE THE PROBLEM

There are two different approaches to solve the time synchronisation problem: the hardware approach and the software approach. The hardware approach is through a triggering device where the input change generates an instant output change. The hardware approach is able to avoid the priority issue where a software signal passes the CPU. In terms of the software approach, a single conversion is started by writing a logical one to the A/D converter indicating the start of the conversion. If a different data channel is selected while a conversion is in progress, the ADC will finish the current conversion before performing the channel change.

5.3 INTERPOLATION

Different approaches to the time synchronisation problem have been used in the research literature. MAV Datasets (Lee, 2010) used the cubic interpolation for the synchronisation issue. The figures below are linear and cubic interpolation on gyroscope and accelerometer sensor data to compensate the sampling frequency difference, and are simulated in MATLAB.

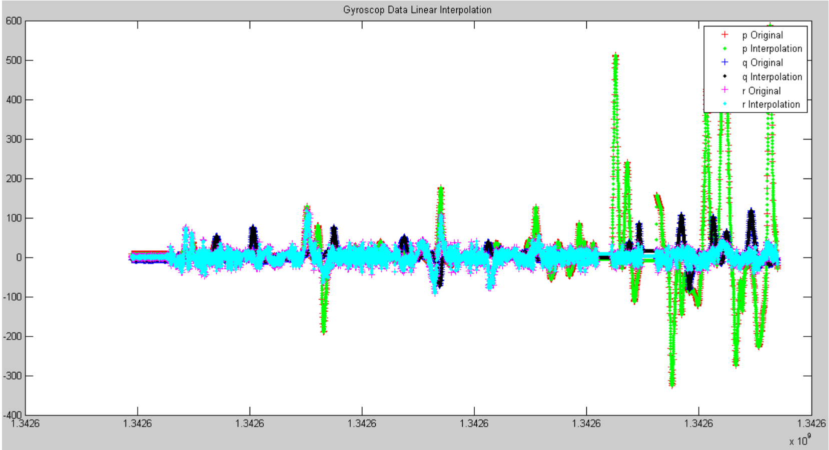


Figure 5.5: Linear interpolation on gyroscope data.

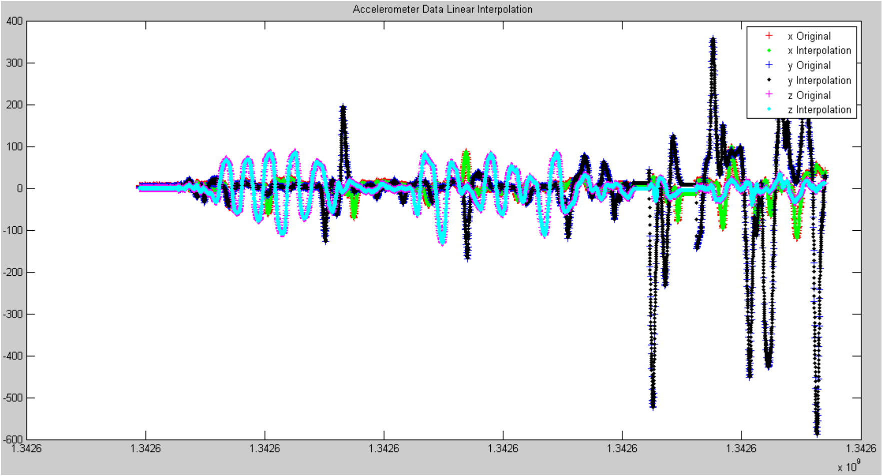


Figure 5.6: Linear interpolation on accelerometer data.

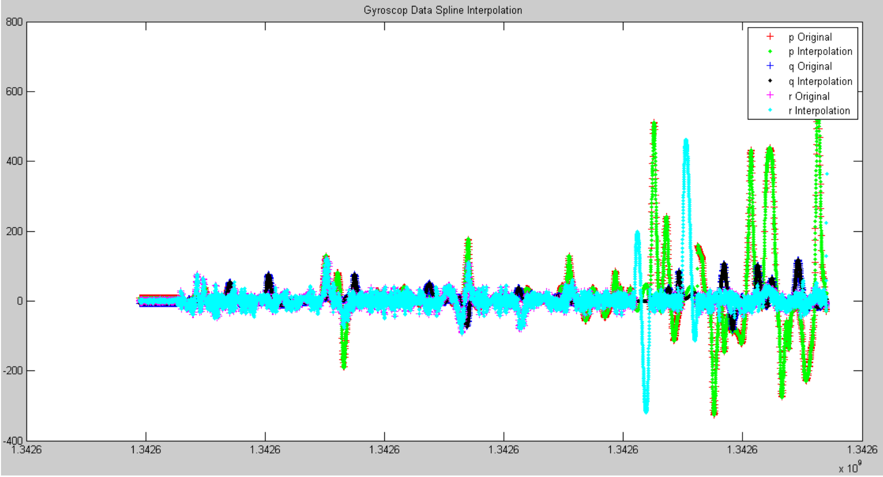


Figure 5.7: Cubic interpolation on gyroscope data.

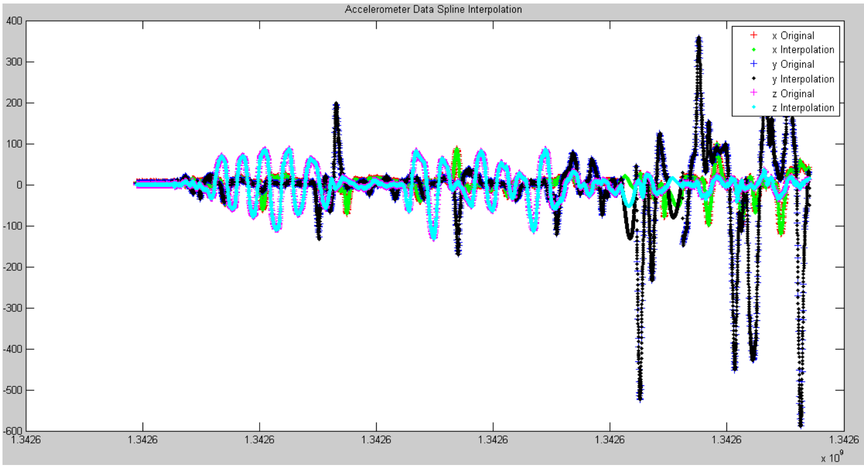


Figure 5.8: Cubic interpolation on accelerometer data.

If we zoom the MATLAB simulation image, it would have the following effect.

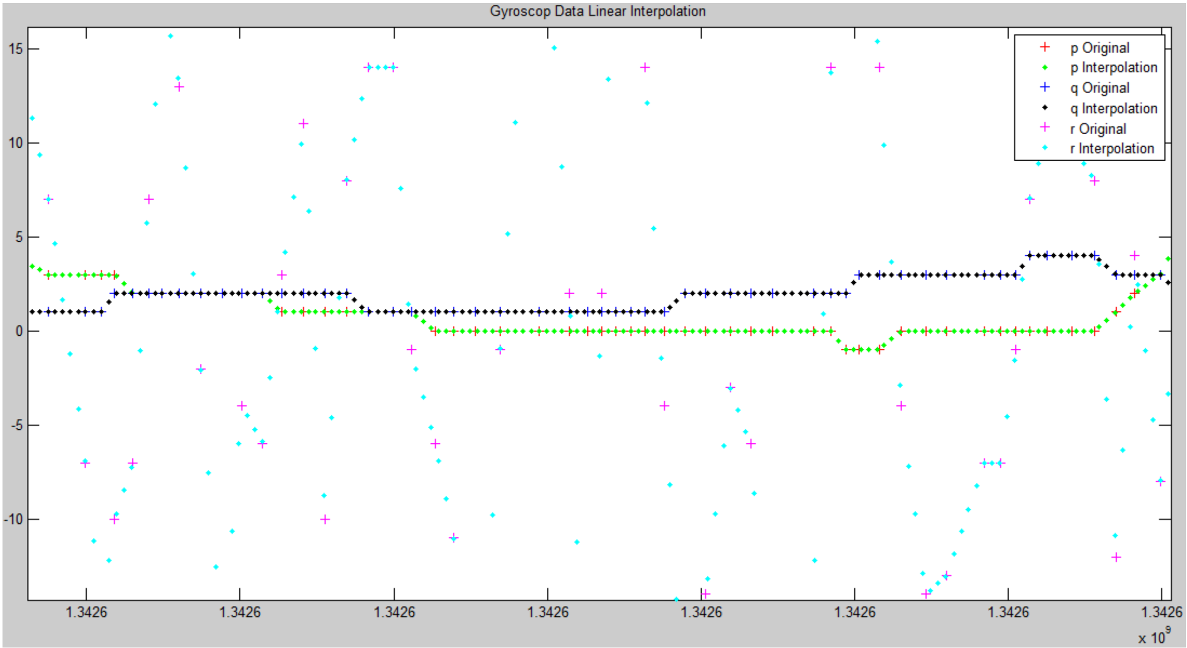


Figure 5.9: Linear interpolation on gyroscope data.

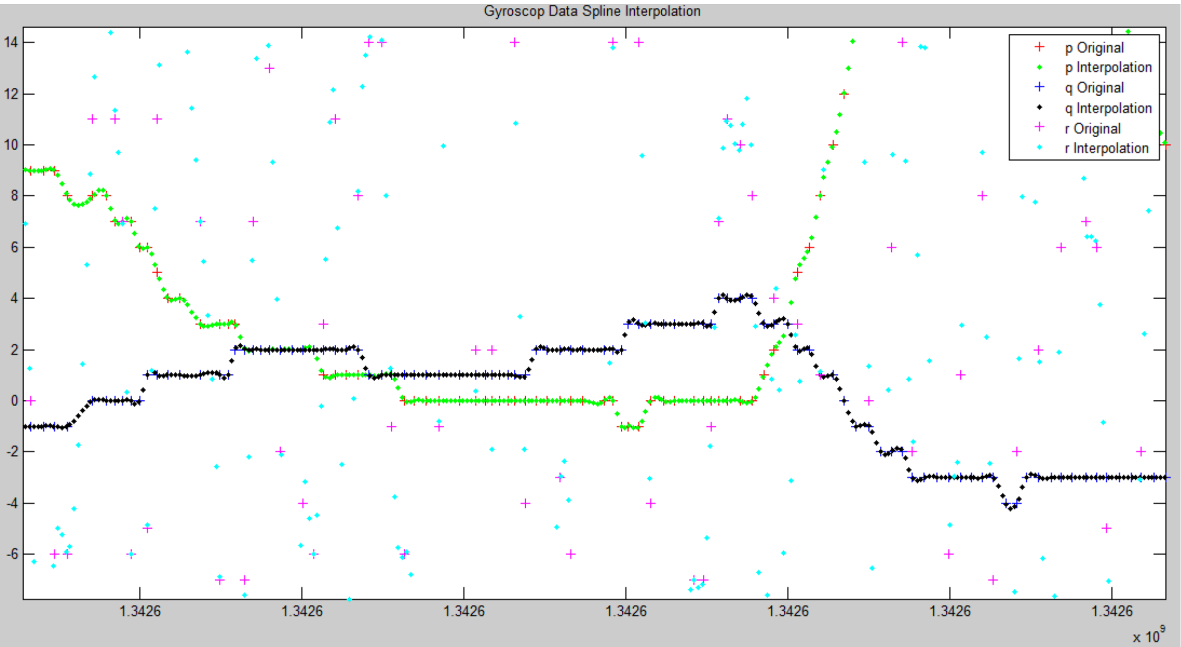


Figure 5.10: Cubic interpolation on gyroscope data.

Generally speaking, the cubic interpolation has a better result compared with the linear interpolation. The interpolation methodology is able to solve the time synchronisation issue on sampling frequency difference.

5.4 CHAPTER SUMMARY

This chapter first analyses which sources of time synchronisation problem need to be addressed for the Mikrokopter-based quadrotor. This chapter then provides general information about hardware and software approaches for A/D conversion. Finally, this chapter exams existing solutions and how it different with this research project and provides a possible solution to the time synchronisation issues.

Chapter 6 Conclusions and Future Work

6.1 CONCLUSIONS

The aim of this thesis is to design and collect a benchmarking data set used for four different attitude estimation algorithms in the Mikrokopter-based quadrotor platform. This thesis first provides necessary background on the Mikrokopter platform, three orientation representations, performance comparison and sampling and A/D converter. It then analyses the input characteristics of four attitude estimation algorithms, the format and semantics of the sensor measurements, and the issues identified as a need for sensor pre-processing. A detailed analysis of the time synchronisation problem on the clock and the sampling issues are provided later in the thesis. Finally, the time synchronisation issue is analysed for the Mikrokopter platform, existing solutions to the time synchronisation problem and how they are different from this research project are examined, and a possible solution to the sensor pre-processing on the time synchronisation in this research scenario is proved.

6.2 FUTURE WORK

The outcome of this thesis is not a complete version of the benchmarking data set for attitude and pose estimators in UAV applications. Possible future directions of this research include but are not restricted to the following points.

- Apply other auxiliary sensors, camera sensor for example, to provide additional information for sensor data pre-processing.
- Investigate the time synchronisation issue on the Mikrokopter platform to achieve higher time synchronisation accuracy.
- Pre-pressing sensor data to calibrate the sensor accuracy issues for the Mikrokopter platform.
- Scenario design where more synchronized datasets with different flying scenarios on Mikrokopter platform are designed and collected.
- Apply the above sensor data pre-processing techniques to other UAV application platforms.

Bibliography

Aerospace Control Systems. (n.d.). What is the effect of thermal stability on attitude determination performance? Retrieved 23th October, 2013, from the Aerospace Control Systems LLC (ACS) website: http://www.acsinnovations.com/Literature/FAQ/What-is-the-effect-of-thermal-stability-on-attitude-determination-performance/

Analogue Devices. (2010). ADXRS610 datasheet. Retrieved 17th October, 2013, from the website: http://www.analogue.com/static/imported-files/data_sheets/ADXRS610.pdf

Anderson, B. D. O., and Moore, J. (1979). *Optimal filtering.* Prentice Hall.

Artese, G., and Trecroci, A. (2008). Calibration of a low cost MEMS INS sensor for an integrated navigation system. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. XXXVII (B5),* 877-882.

Atmel. (2012). ATmega644 datasheet. Retrieved 17th October, 2013, from the website: http://www.atmel.com/Images/2593s.pdf

Craig, J. (2005). *Introduction to Robotics: Mechanics and Control* (3rd ed.). Prentice Hall.

Crassidis, J. L., Markley, F. L., and Cheng, Y. (2007). Survey of nonlinear attitude estimation methods. *Journal of Guidance, Control, and Dynamics, 30(3),* 12 -28.

Crassidis, J. and Markley, F. (2003). Unscented filtering for spacecraft attitude estimation. *Journal of Guidance Control and Dynamics, 26(4),* 536–542.

Davis, E. (2013). Development of a low cost quadrotor platform for swarm experiments. *Proceedings of the 32nd Chinese Control Conference.* 7072 - 7077.

Diebel, J. (2006). Representing attitude: Euler angles, unit quaternions, and rotation vectors. *Stanford University Technical Report, Stanford University, Stanford, CA.*

Ding, W., Wang, J., Li, Y., Mumford, P., and Rizos, C. (2008). Time synchronization error and calibration in integrated GPS/INS systems. *ETRI, 30(1),* 59-67.

Elbornsson, J. (2003). Analysis, estimation and compensation of mismatch effects in A/D converters. PhD thesis, Linköpings universitet, Sweden.

Eng, F. (2007). Non-uniform sampling in statistical signal processing. PhD thesis, Linköpings universitet, Sweden.

Haby, J. (n.d.). What is the difference between accuracy and precision? Retrieved 25th October, 2013, from the ultimate weather education website: http://www.theweatherprediction.com/habyhints/246/

HiSystems GmbH. (2008). Mikrokopter. Retrieved 15th April, 2013, from the website: http://gallery.mikrokopter.de/main.php/v/tech/CIMG8456_FC.JPG.html?g2_imageViewsIndex=1

Hou, X. (2013). Energy supervision based obstacle avoidance for haptic teleoperation of UAVs - version 1. Retrieved 25th September, 2013, from the website: https://www.youtube.com/watch?v=0_GxM-gT8xw

Julier, S. J., Uhlmann, J. K., and Durrant-Whyte, H. F. (1995). A new approach for filtering nonlinear systems. *Proceedings of the American Control Conference, 3*, 1628–1632.

Lee, G. H., Achtelik, M., Fraundorfer, F., Pollefeys, M., and Siegwart, R. (2010). A benchmarking tool for MAV visual pose estimation. *ICARCV. 6(10)*, 1541-1546.

Lefferts, E. J., Markley, F. L., and Shuster, M. D. (1982). Kalman filtering for spacecraft attitude estimation. *Journal of Guidance, Control, and Dynamics, 5(5)*, 417–429.

Li, Y., Niu, X., Zhang, Q., Zhang H. and Shi, C. (2012). An in situ hand calibration method using a pseudo-observation scheme for low-end inertial measurement units. *Measurement Science and Technology. 23 (10)*, 1-11.

Lim, H., Park, J., Lee, D., and Kim, H.J. (2012). Build your own quadrotor. *IEEE Robotics & Automation Magazine, 19(3)*, 33-45.

Mahony, R. (2013). *Spatial coordinates*, lecture notes distributed in Robotics at The Australian National University.

Markley, F., (2003). Attitude error representations for Kalman filtering. *Journal of guidance, control, and dynamics, 26(2)*, 311–317.

Marouani, H. and Dagenais, M. R. (2008). Internal clock drift estimation in computer clusters. *Journal of Computer Systems, Networks, and Communications, 2008(583162)*, 1-7.

Mortensen, R. E. (1968). Maximum-likelihood recursive nonlinear filtering. *Journal of Optimization Theory and Applications, 2(6)*, 368-394.

O'Shaughnessy, D. J., Vaughan, R. M., Haley, D.R., and Shapiro, H. S. (2006). MESSENGER IMU interface timing issues and in-flight calibration results. *Guidance and Control, 125*, 485-505.

Pascoal, A., Kaminer, I., and Oliveira, P. (2000). Navigation system design using time-varying complementary filters. *IEEE Trans. Aerosp. Electron. Syst., 36(4)*, 1099–1114.

Sa, I. (2011). MikroKopter Documentation. Cyphy Laboratory. Retrieved 20th July, 2013, from the website: https://www.google.com.au/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CC0QFjAA&url=https%3A%2F%2Fwiki.qut.edu.au%2Fdownload%2Fattachments%2F145528940%2FMK_Doc.pdf%3Fversion%3D1%26modificationDate%3D1339475905000&ei=B7NxUqKwAs2ekgW6kYCQDw&usg=AFQjCNFPyjyzNTiE5SZyPvanuYfRZ4QmZw&sig2=tTC-wZNKLtbCHiRRB3rO7A&bvm=bv.55819444,d.dGI&cad=rja

Stmicroelectronics. (2008). LIS344ALH datasheet. Retrieved 17th October, 2013, from the website:http://www.st.com/web/en/resource/technical/document/datasheet/CD00182781.pdf

Syed, Z. F., Aggarwal, P., Goodall, C., Niu, X. and El-Sheimy, N. (2007). A new multi-position calibration method for MEMS inertial navigation systems. *Measurement Science and Technology. 18 (7),* 1897-1907.

The Johns Hopkins University. (2013). Vicon motion capturing system. Retrieved 17th October, 2013, from the website: http://krieger.jhu.edu/mbi/hsiaolab/facilities/Vicon.html

Titterton, D. H., and Weston, J. L. (1997). *Strapdown Inertial Navigation Technology*. London: Institution of Electrical Engineers.

Twiss, R. J., and Moores, E. M. (1992). *The orientation of structures. Structural geology* (2nd ed.). Macmillan.

Verth, J. V. (2012). *Understanding rotations*. Retrieved 15th April, 2013, from the website: https://www.google.com.au/search?q=Understanding+Rotations&ie=utf-8&oe=utf-8&aq=t&rls=org.mozilla:zh-CN:official&client=firefox-a&channel=fflb&gfe_rd=cr&ei=_7lxUsWtEMrC8geSnIGICw

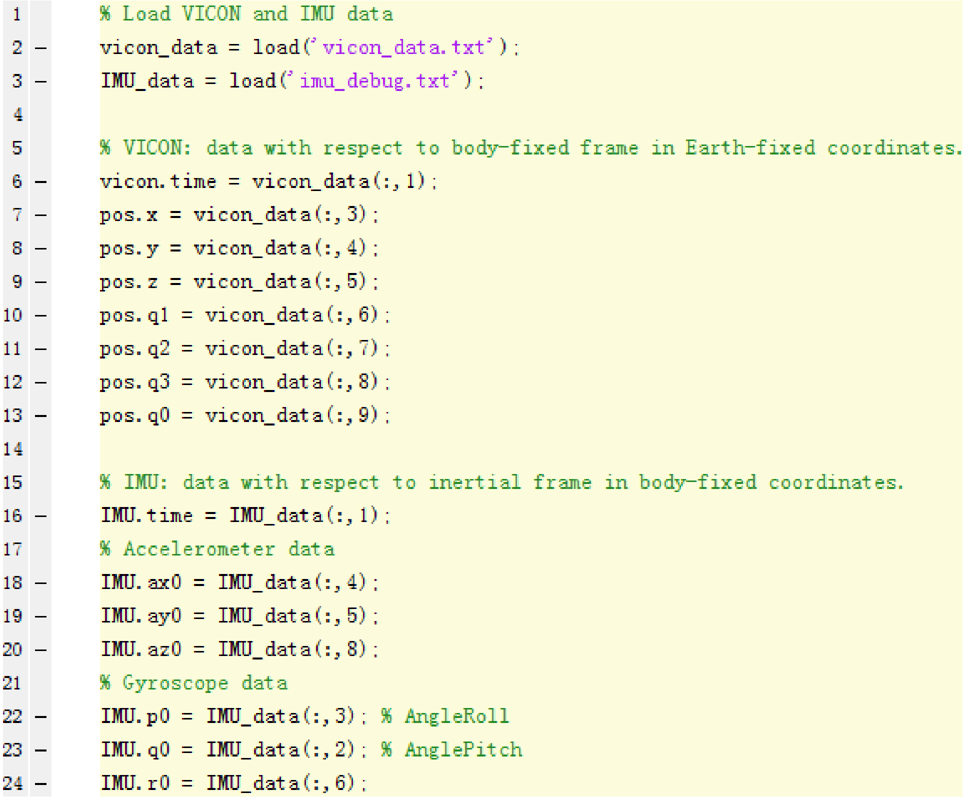
White, D. R., and Clare, J.F., (1992). Noise in measurements obtained by sampling. *Meas. Sci. Technol. 3(1),* 1-16.

Zamani, M. (2013). Deterministic attitude and pose filtering, an embedded lie groups approach. PhD thesis, The Australian National University, Australia.

Appendix A MATLAB Code

Presented here are the MATLAB code for the simulation and analysis of this research project.

Listing A.1: Read sensor data in MATLAB.

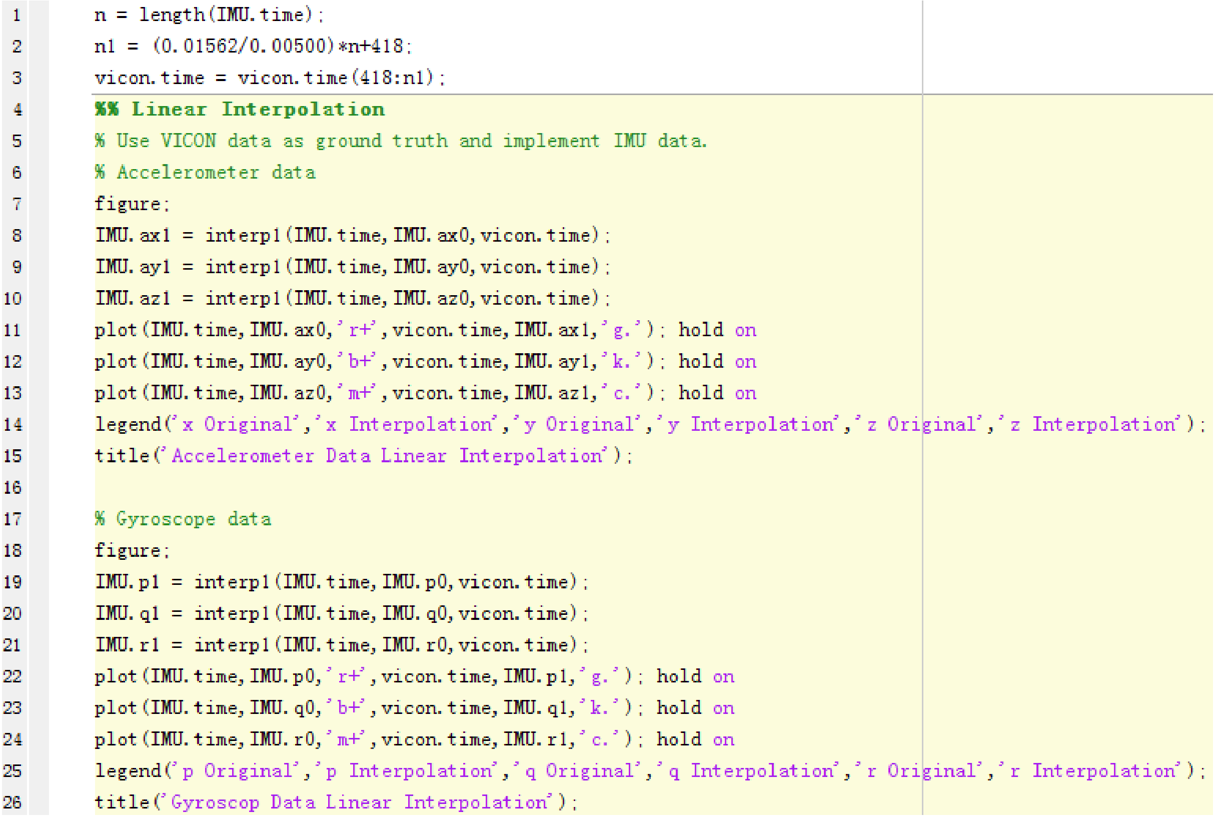


```
% Load VICON and IMU data
vicon_data = load('vicon_data.txt');
IMU_data = load('imu_debug.txt');

% VICON: data with respect to body-fixed frame in Earth-fixed coordinates.
vicon.time = vicon_data(:,1);
pos.x = vicon_data(:,3);
pos.y = vicon_data(:,4);
pos.z = vicon_data(:,5);
pos.q1 = vicon_data(:,6);
pos.q2 = vicon_data(:,7);
pos.q3 = vicon_data(:,8);
pos.q0 = vicon_data(:,9);

% IMU: data with respect to inertial frame in body-fixed coordinates.
IMU.time = IMU_data(:,1);
% Accelerometer data
IMU.ax0 = IMU_data(:,4);
IMU.ay0 = IMU_data(:,5);
IMU.az0 = IMU_data(:,8);
% Gyroscope data
IMU.p0 = IMU_data(:,3); % AngleRoll
IMU.q0 = IMU_data(:,2); % AnglePitch
IMU.r0 = IMU_data(:,6);
```

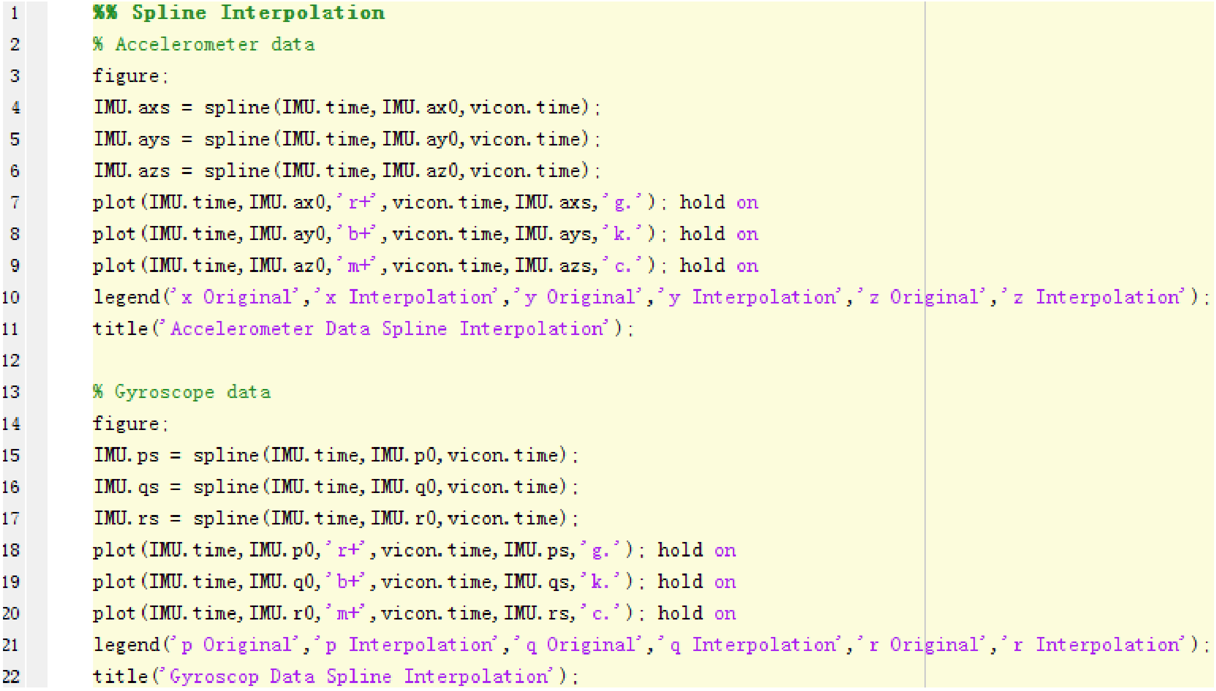
Listing A.2: Linear Interpolation.



```
n = length(IMU.time);
n1 = (0.01562/0.00500)*n+418;
vicon.time = vicon.time(418:n1);
%% Linear Interpolation
% Use VICON data as ground truth and implement IMU data.
% Accelerometer data
figure;
IMU.ax1 = interp1(IMU.time,IMU.ax0,vicon.time);
IMU.ay1 = interp1(IMU.time,IMU.ay0,vicon.time);
IMU.az1 = interp1(IMU.time,IMU.az0,vicon.time);
plot(IMU.time,IMU.ax0,'r+',vicon.time,IMU.ax1,'g.'); hold on
plot(IMU.time,IMU.ay0,'b+',vicon.time,IMU.ay1,'k.'); hold on
plot(IMU.time,IMU.az0,'m+',vicon.time,IMU.az1,'c.'); hold on
legend('x Original','x Interpolation','y Original','y Interpolation','z Original','z Interpolation');
title('Accelerometer Data Linear Interpolation');

% Gyroscope data
figure;
IMU.p1 = interp1(IMU.time,IMU.p0,vicon.time);
IMU.q1 = interp1(IMU.time,IMU.q0,vicon.time);
IMU.r1 = interp1(IMU.time,IMU.r0,vicon.time);
plot(IMU.time,IMU.p0,'r+',vicon.time,IMU.p1,'g.'); hold on
plot(IMU.time,IMU.q0,'b+',vicon.time,IMU.q1,'k.'); hold on
plot(IMU.time,IMU.r0,'m+',vicon.time,IMU.r1,'c.'); hold on
legend('p Original','p Interpolation','q Original','q Interpolation','r Original','r Interpolation');
title('Gyroscop Data Linear Interpolation');
```

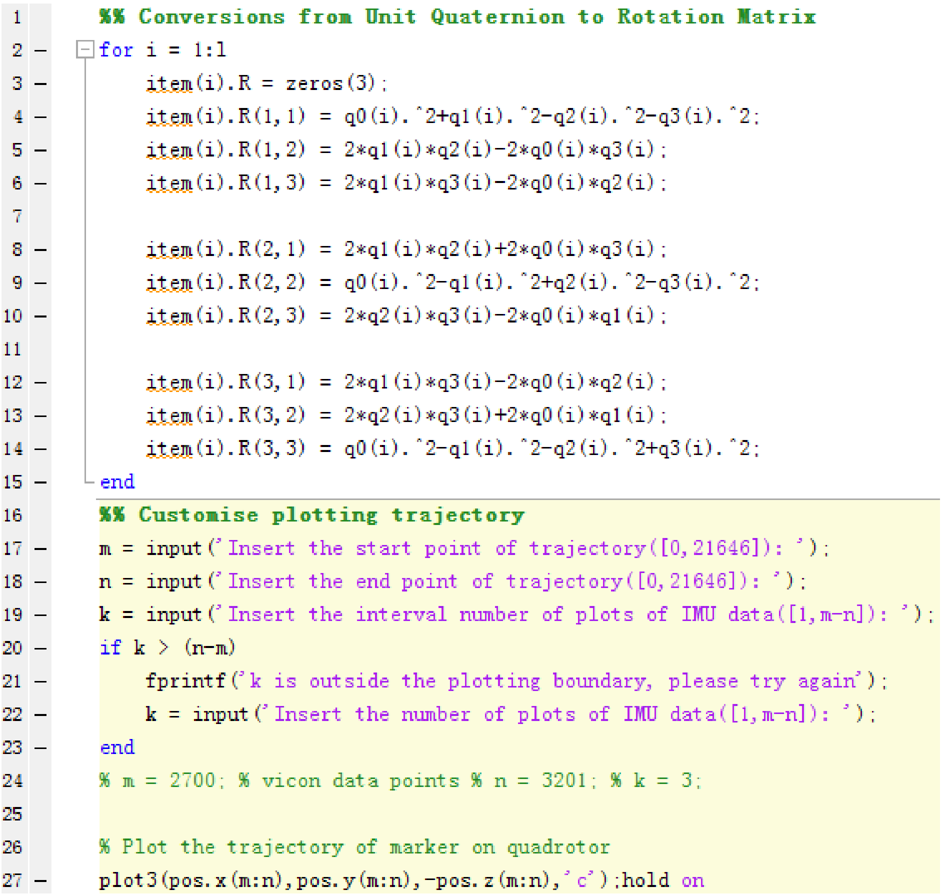
Listing A.3: Spline Interpolation.



```
%% Spline Interpolation
% Accelerometer data
figure;
IMU.axs = spline(IMU.time,IMU.ax0,vicon.time);
IMU.ays = spline(IMU.time,IMU.ay0,vicon.time);
IMU.azs = spline(IMU.time,IMU.az0,vicon.time);
plot(IMU.time,IMU.ax0,'r+',vicon.time,IMU.axs,'g.'); hold on
plot(IMU.time,IMU.ay0,'b+',vicon.time,IMU.ays,'k.'); hold on
plot(IMU.time,IMU.az0,'m+',vicon.time,IMU.azs,'c.'); hold on
legend('x Original','x Interpolation','y Original','y Interpolation','z Original','z Interpolation');
title('Accelerometer Data Spline Interpolation');

% Gyroscope data
figure;
IMU.ps = spline(IMU.time,IMU.p0,vicon.time);
IMU.qs = spline(IMU.time,IMU.q0,vicon.time);
IMU.rs = spline(IMU.time,IMU.r0,vicon.time);
plot(IMU.time,IMU.p0,'r+',vicon.time,IMU.ps,'g.'); hold on
plot(IMU.time,IMU.q0,'b+',vicon.time,IMU.qs,'k.'); hold on
plot(IMU.time,IMU.r0,'m+',vicon.time,IMU.rs,'c.'); hold on
legend('p Original','p Interpolation','q Original','q Interpolation','r Original','r Interpolation');
title('Gyroscop Data Spline Interpolation');
```

Listing A.4: Conversions from unit quaternion to rotation matrix.



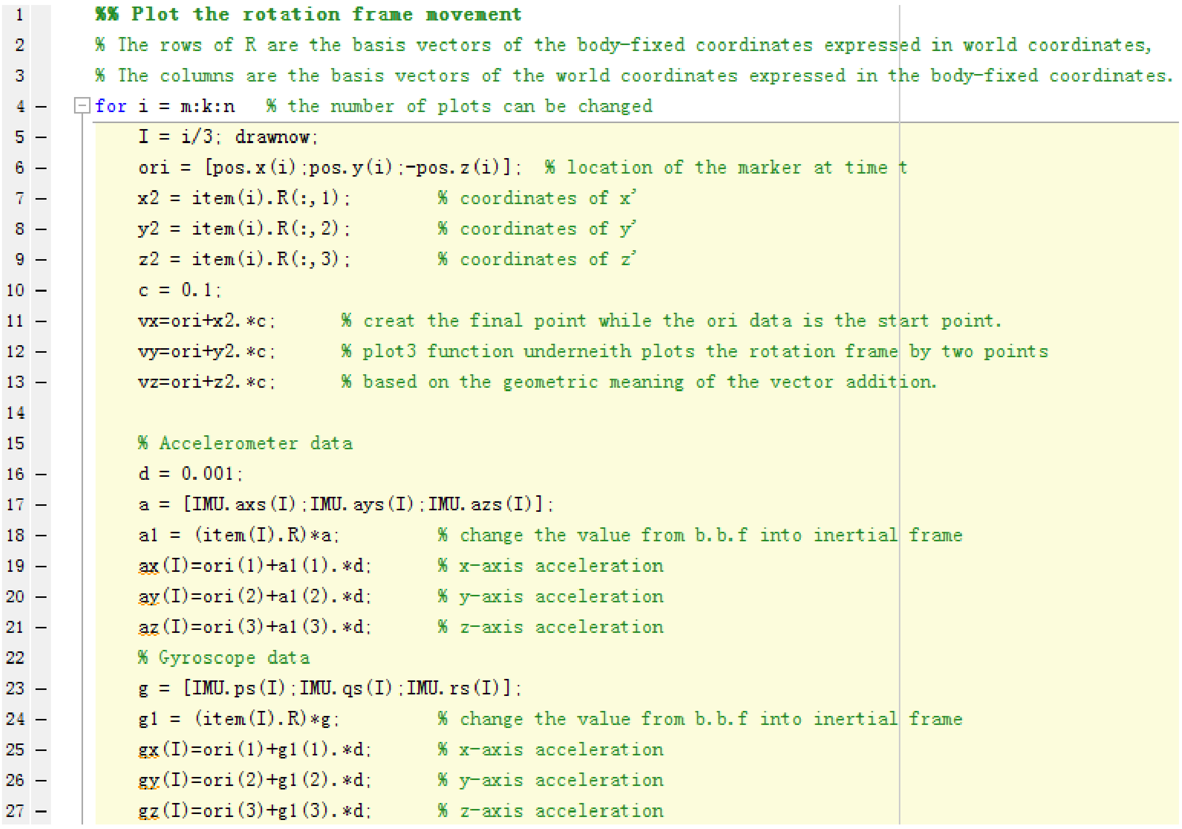
```
%% Conversions from Unit Quaternion to Rotation Matrix
for i = 1:1
    item(i).R = zeros(3);
    item(i).R(1,1) = q0(i).^2+q1(i).^2-q2(i).^2-q3(i).^2;
    item(i).R(1,2) = 2*q1(i)*q2(i)-2*q0(i)*q3(i);
    item(i).R(1,3) = 2*q1(i)*q3(i)-2*q0(i)*q2(i);

    item(i).R(2,1) = 2*q1(i)*q2(i)+2*q0(i)*q3(i);
    item(i).R(2,2) = q0(i).^2-q1(i).^2+q2(i).^2-q3(i).^2;
    item(i).R(2,3) = 2*q2(i)*q3(i)-2*q0(i)*q1(i);

    item(i).R(3,1) = 2*q1(i)*q3(i)-2*q0(i)*q2(i);
    item(i).R(3,2) = 2*q2(i)*q3(i)+2*q0(i)*q1(i);
    item(i).R(3,3) = q0(i).^2-q1(i).^2-q2(i).^2+q3(i).^2;
end
%% Customise plotting trajectory
m = input('Insert the start point of trajectory([0,21646]): ');
n = input('Insert the end point of trajectory([0,21646]): ');
k = input('Insert the interval number of plots of IMU data([1,m-n]): ');
if k > (n-m)
    fprintf('k is outside the plotting boundary, please try again');
    k = input('Insert the number of plots of IMU data([1,m-n]): ');
end
% m = 2700; % vicon data points % n = 3201; % k = 3;

% Plot the trajectory of marker on quadrotor
plot3(pos.x(m:n),pos.y(m:n),-pos.z(m:n),'c');hold on
```

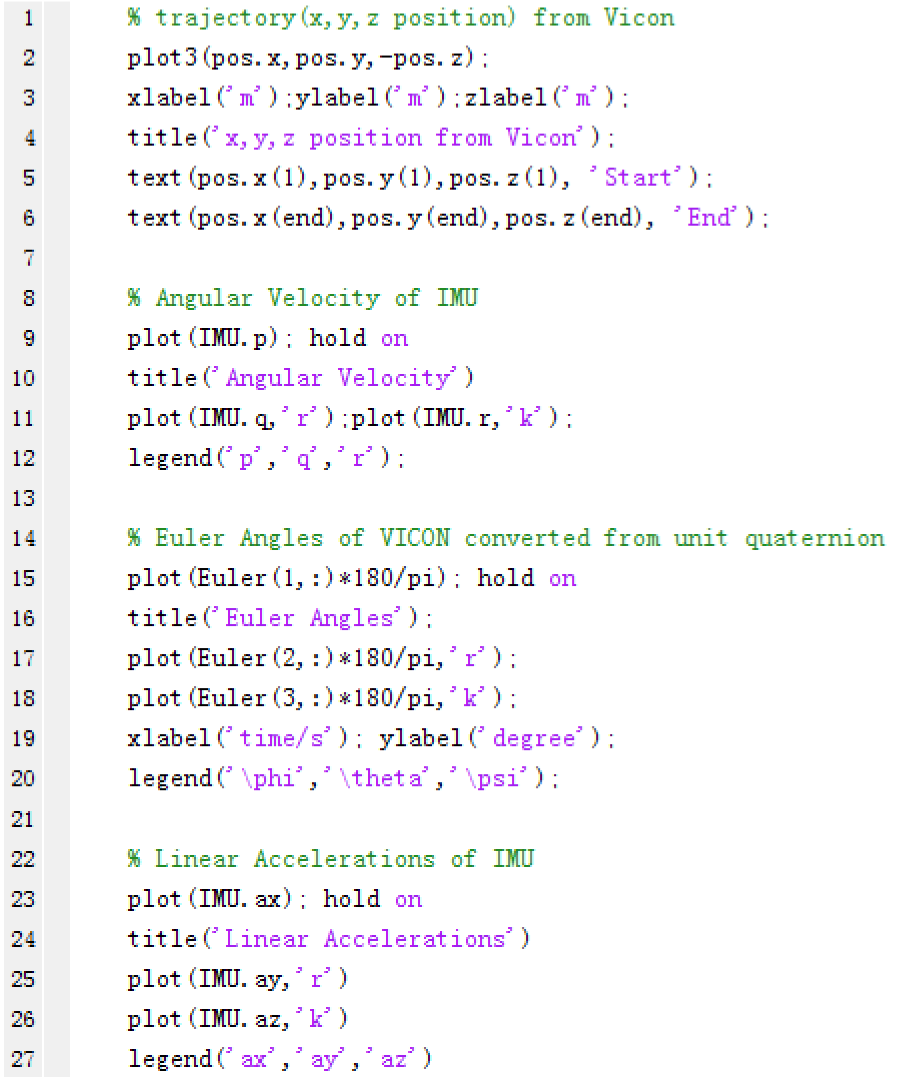
Listing A.5: Rotation frame movement.



```
%% Plot the rotation frame movement
% The rows of R are the basis vectors of the body-fixed coordinates expressed in world coordinates,
% The columns are the basis vectors of the world coordinates expressed in the body-fixed coordinates.
for i = m:k:n   % the number of plots can be changed
    I = i/3; drawnow;
    ori = [pos.x(i);pos.y(i);-pos.z(i)];  % location of the marker at time t
    x2 = item(i).R(:,1);         % coordinates of x'
    y2 = item(i).R(:,2);         % coordinates of y'
    z2 = item(i).R(:,3);         % coordinates of z'
    c = 0.1;
    vx=ori+x2.*c;      % creat the final point while the ori data is the start point.
    vy=ori+y2.*c;      % plot3 function underneith plots the rotation frame by two points
    vz=ori+z2.*c;      % based on the geometric meaning of the vector addition.

    % Accelerometer data
    d = 0.001;
    a = [IMU.axs(I);IMU.ays(I);IMU.azs(I)];
    a1 = (item(I).R)*a;          % change the value from b.b.f into inertial frame
    ax(I)=ori(1)+a1(1).*d;       % x-axis acceleration
    ay(I)=ori(2)+a1(2).*d;       % y-axis acceleration
    az(I)=ori(3)+a1(3).*d;       % z-axis acceleration
    % Gyroscope data
    g = [IMU.ps(I);IMU.qs(I);IMU.rs(I)];
    g1 = (item(I).R)*g;          % change the value from b.b.f into inertial frame
    gx(I)=ori(1)+g1(1).*d;       % x-axis acceleration
    gy(I)=ori(2)+g1(2).*d;       % y-axis acceleration
    gz(I)=ori(3)+g1(3).*d;       % z-axis acceleration
```

Listing A.6: Visualisations of sensor data.



```
% trajectory(x,y,z position) from Vicon
plot3(pos.x,pos.y,-pos.z);
xlabel('m');ylabel('m');zlabel('m');
title('x,y,z position from Vicon');
text(pos.x(1),pos.y(1),pos.z(1), 'Start');
text(pos.x(end),pos.y(end),pos.z(end), 'End');

% Angular Velocity of IMU
plot(IMU.p); hold on
title('Angular Velocity')
plot(IMU.q,'r');plot(IMU.r,'k');
legend('p','q','r');

% Euler Angles of VICON converted from unit quaternion
plot(Euler(1,:)*180/pi); hold on
title('Euler Angles');
plot(Euler(2,:)*180/pi,'r');
plot(Euler(3,:)*180/pi,'k');
xlabel('time/s'); ylabel('degree');
legend('\phi','\theta','\psi');

% Linear Accelerations of IMU
plot(IMU.ax); hold on
title('Linear Accelerations')
plot(IMU.ay,'r')
plot(IMU.az,'k')
legend('ax','ay','az')
```