UNSW SYDNEY

Aircraft Attitude Estimation Using Panoramic Images

Author:
Habibi Tehrani, Mohsen

Publication Date:
2014

DOI:
https://doi.org/10.26190/unsworks/2657

License:
https://creativecommons.org/licenses/by-nc-nd/3.0/au/
Link to license to see what you are allowed to do with this resource.

Downloaded from http://hdl.handle.net/1959.4/54009 in https://unsworks.unsw.edu.au on 2024-04-28

Aircraft Attitude Estimation Using Panoramic Images

Mohsen Habibi Tehrani

A thesis submitted in fulfilment
of the requirements for the degree of
Doctor of Philosophy

School of Engineering and Information Technology
University of New South Wales, Canberra, Australia

March 2014

THE UNIVERSITY OF NEW SOUTH WALES
Thesis/Dissertation Sheet

Surname or Family name: Habibi Tehrani
First name: Mohsen Other name/s:
Abbreviation for degree as given in the University calendar: PhD
School: School of Engineering and Information Technology Faculty: UNSW Canberra
Title: AIRCRAFT ATTITUDE ESTIMATION USING PANORAMIC IMAGES

Abstract (350 words maximum)

This thesis investigates the problem of reliably estimating attitude from panoramic imagery in cluttered environments. Accurate attitude is an essential input to the stabilisation systems of autonomous aerial vehicles. A new camera system which combines a CCD camera, UltraViolet (UV) filters and a panoramic mirror-lens is designed. Drawing on biological inspiration from the Ocelli organ possessed by certain insects, UV filtered images are used to enhance the contrast between the sky and ground and mitigate the effect of the sun. A novel method for real–time horizon-based attitude estimation using panoramic image that is capable of estimating an aircraft pitch and roll at a low altitude in the presence of sun, clouds and occluding features such as tree, building, is developed. Also, a new method for panoramic sky/ground thresholding, consisting of a horizon– and a sun–tracking system which works effectively even when the horizon line is difficult to detect by normal thresholding methods due to flares and other effects from the presence of the sun in the image, is proposed. An algorithm for estimating the attitude from three–dimensional mapping of the horizon projected onto a 3D plane is developed. The use of optic flow to determine pitch and roll rates is investigated using the panoramic image and image interpolation algorithm (I^2A). Two methods which employ sensor fusion techniques, Extended Kalman Filter (EKF) and Artificial Neural Networks (ANNs), are used to fuse unfiltered measurements from inertial sensors and the vision system. The EKF estimates gyroscope biases and also the attitude. The ANN fuses the optic flow and horizon–based attitude to provide smooth attitude estimations. The results obtained from different parts of the research are tested and validated through simulations and real flight tests.

Declaration relating to disposition of project thesis/dissertation

I hereby grant to the University of New South Wales or its agents the right to archive and to make available my thesis or dissertation in whole or in part in the University libraries in all forms of media, now or here after known, subject to the provisions of the Copyright Act 1968. I retain all property rights, such as patent rights. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

I also authorise University Microfilms to use the 350 word abstract of my thesis in Dissertation Abstracts International (this is applicable to doctoral theses only).

....
Signature Witness Date

The University recognises that there may be exceptional circumstances requiring restrictions on copying or conditions on use. Requests for restriction for a period of up to 2 years must be made in writing. Requests for a longer period of restriction may be considered in exceptional circumstances and require the approval of the Dean of Graduate Research.

FOR OFFICE USE ONLY Date of completion of requirements for Award:

Copyright Statement

I hereby grant The University of New South Wales or its agents the right to archive and to make available my thesis or dissertation in whole or part in the University libraries in all forms of media, now or hereafter known, subject to the provisions of the Copyright Act 1968. I retain all proprietary rights, such as patent rights. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

I also authorise University Microfilms to use the abstract of my thesis in Dissertation Abstract International (this is applicable to doctoral thesis only).

I have either used no substantial portions of copyright material in my thesis or I have obtained permission to use copyright material; where permission has not been granted I have applied/will apply for a partial restriction of the digital copy of my thesis or dissertation.

Signed ....

Date ....

Authenticity Statement

I certify that the Library deposit digital copy is a direct equivalent of the final officially approved version of my thesis. No emendation of content has occurred and if there are any minor variations in formatting, they are the result of the conversion to digital format.

Signed ....

Date ....

Originality Statement

I hereby declare that this submission is my own work and to the best of my knowledge it contains no material previously published or written by another person, or substantial portions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institute, except where due acknowledgment is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.

Signed ....

Date ....

Abstract

This thesis investigates the problem of reliably estimating attitude from panoramic imagery in cluttered environments. Accurate attitude is an essential input to the stabilisation systems of autonomous aerial vehicles. A new camera system which combines a CCD camera, UltraViolet (UV) filters and a panoramic mirror-lens is designed. Drawing on biological inspiration from the Ocelli organ possessed by certain insects, UV filtered images are used to enhance the contrast between the sky and ground and mitigate the effect of the sun. A novel method for real–time horizon-based attitude estimation using panoramic image that is capable of estimating an aircraft pitch and roll at a low altitude in the presence of sun, clouds and occluding features such as tree, building, is developed. Also, a new method for panoramic sky/ground thresholding, consisting of a horizon– and a sun–tracking system which works effectively even when the horizon line is difficult to detect by normal thresholding methods due to flares and other effects from the presence of the sun in the image, is proposed. An algorithm for estimating the attitude from three–dimensional mapping of the horizon projected onto a 3D plane is developed. The use of optic flow to determine pitch and roll rates is investigated using the panoramic image and image interpolation algorithm (I^2A). Two methods which employ sensor fusion techniques, Extended Kalman Filter (EKF) and Artificial Neural Networks (ANNs), are used to fuse unfiltered measurements from inertial sensors and the vision system. The EKF estimates gyroscope biases and also the attitude. The ANN fuses the optic flow and horizon–based attitude to provide smooth attitude estimations. The results obtained from different parts of the research are tested and validated through simulations and real flight tests.

Acknowledgments

Foremost, I would like to express my deepest gratitude to my supervisors, Dr. Matthew Garratt and Dr Sreenatha Anavatti, for their continuous support since the first day of my PhD study. I will be forever thankful for their priceless advice, patience and immense knowledge as without their guidance, it would have been impossible for me to complete this work. In particular, I would like to thank Dr. Matthew Garratt for his invaluable encouragement and insightful discussions that helped me improve my knowledge of image processing and sensor fusion that were the main parts of my thesis.

In addition, I also want to thank Prof. Jochen Zeil and Dr. Wolfgang Stuerzl for providing the panoramic lens which was one of the key requirement for implementing this work. A special thanks to Mrs Denise Russell for proofreading my thesis.

Most importantly, I would like to thank my parents, Shahsavar and Marzieh, and my brothers, Amin and Younes, and my sister, Narges, for their invaluable love and support throughout my education.

I want to express my deepest gratitude to Mr Wilson Saed for his support and spiritual inspiration throughout my life.

I would like to thank Mr Mohammad Esmaeil Zadeh and his family for their help during my PhD.

On a personal note, i want to especially thank my closest friends in Australia for helping me and treating me as a part of their family throughout my way: Ms Bahar and Victoria Kakavand, Ms Munes Saadati, Mr Alireza, Ms Atousa and Ms Anousha Hassanpur, Ms Mojgan Kakavand and Mr Mahyar Rami nejad.

List of Publications

M. H. Tehrani, M. A. Garratt, and S. Anavatti. Horizon-Based Attitude Estimation from a Panoramic Vision Sensor. In Embedded Guidance, Navigation and Control in Aerospace, EGNCA, http://www.ifac-egnca.org/, 2012. Bangalore, India.

M. H. Tehrani, M. A. Garratt, and S. Anavatti. Gyroscope offset estimation using panoramic vision-based attitude estimation and Extended Kalman Filter. In 2nd International Conference on Communications, Computing and Control Applications (CCCA), 1-5. IEEE, 2012.

M. H Tehrani, M. A. Garratt, and S Anavatti. An accurate attitude estimation using panoramic vision and an Extended Kalman Filter. Global Journal on Technology, 3, 2013.

Contents

List of Figures

List of Tables

Nomenclature

a	Accelerations
A	Horizon points matrix
A	3D horizon plane normal vector coefficient in x axis
$\mathbf{b}$	Magnetic field (Tesla)
b	Neuron's bias
B	3×3 rotation matrix
B	3D horizon plane's normal vector coefficient in y axis
C	3D horizon plane's normal vector coefficient in x axis
d	Distance of an object from the horizon
$f_1 - f_6$	Six reference images
g	Acceleration due to gravity (scalar)
g	Gravity vector
h	Height above ground
H	EKF observation matrix
K	EKF gain matrix
P	Actual gyroscopic measurement in along aircraft x axis (roll)
$\mathbf{P}$	EKF covariance matrix
q	Vector of quaternion parameters $[q_0\, q_1\, q_2\, q_3]^T$
q_0, q_1, q_2, q_3	Quaternion parameters
Q	Actual gyroscopic measurement in along aircraft y axis (pitch)
$\mathbf{Q}$	EKF process noise matrix
r	Radius in Spherical coordinate system
R	Actual gyroscopic measurement in along aircraft z axis (yaw)
R	Radius of the Earth
$\mathbf{R}$	EKF measurement noise matrix
R_x	Rotation around x-axis in a reference coordinate system
R_y	Rotation around y-axis in a reference coordinate system
R_z	Rotation around z-axis in a reference coordinate system
U	Image pixel coordinate in X axis
V	Image pixel coordinate in Y axis
X	X position in Earth centred coordinates

X_G	Cartesian coordinate of a horizon point in x axis respect to the Earth
w_k	Neuron's weight
Y	Y position in Earth centred coordinates
Y_G	Cartesian coordinate of a horizon point in y axis respect to the Earth
Z	Z position in Earth centred coordinates
Z_G	Cartesian coordinate of a horizon point in z axis respect to the Earth
α	Lens angular gain
α	Elevation angle in spherical coordinate
β	Azimuth angle in spherical coordinate
δ_p	Offset error of roll–rate gyroscope
δ_q	Offset error of pitch–rate gyroscope
δ_r	Offset error of yaw–rate gyroscope
δ_x	Offset error of x–body axis accelerometer
δ_y	Offset error of y–body axis accelerometer
δ_z	Offset error of z–body axis accelerometer
ΔT	Time Step
Δx	Image translation along image's x axis
Δx_{ref}	Image shift reference along image's x axis
Δy	Image translation along image's y axis
Δy_{ref}	Image shift reference along image's y axis
$\Delta\theta$	Image rotation along image's x and y axis
$\Delta\theta_{ref}$	Image rotation reference along image's x and y axis
$\widehat{\Delta x}$	Interpolated image displacement along image's x axis
$\widehat{\Delta y}$	Interpolated image displacement along image's y axis
θ	Aircraft pitch angle
θ_v	Aircraft vision–based pitch angle
θ	Elevation in Spherical coordinate system
μ	Mean value
φ	Azimuth is Spherical coordinate system
$\varphi(v)$	Sigmoidal function
ϕ	Aircraft's roll angle
ϕ_v	Aircraft's vision–based roll angle
Φ	EKF fundamental matrix
Φ	Non–linear model of the system
ψ	Aircraft's yaw angle
Ψ	I^2A patch function
Ω	Angular velocity tensor

Abbreviations

AHRS	Attitude and Heading Reference System
AND	Logical AND operation
ANN	Artificial Neural Network
$\sin^{-1}$	Inverse sine
$\tan^{-1}$	Inverse tangent
bps	Bits per second
CCD	Charge-Coupled Device
CNC machine	Computer Numerical Control machine
cos	Cosine
cm	Centimetre
DCM	Direction Cosine Matrix
ECEF	Earth–Centred Earth-Fixed Frame
EKF	Extended Kalman Filter
g	Gram
GPS	Global Positioning System
KF	Kalman Filter
kg	Kilograms
km	Kilometer
Gbyte	Gigabyte
gyro	Gyroscope
h	Hour
HZ	Unit of frequency
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
IR	Infra–Red
I^2A	Image Interpolation Algorithm
LM	Levenberg-Marquardt Algorithm
I/O	Input–Output
LOG	Laplace Of Guass
LRF	Laser Range Finder

LSE	Least Squared Estimation
Lux	Luminous emittance
m	Meters
MAh	MilliAmps hour
MAV	Micro Air Vehicle
Max	Maximum
Mb/s	Megabit per second
Mbyte	Megabyte
MFI	Micromechanical Flying Insect
Min	Minimum
mm	Millimetres
ms	Millisecond
NED	North, East, Down coordinates
NEU	North, East, Up coordinates
nm	Nano meters
NTSC	National Telvision System Committee
PAL	Phase Alteration Line
FFANN	FeedForward Artificial Neural Network
fps	Frames per second
RGB	Red, Green, Blue
RMS	Root Mean Squared
RUAV	Rotorcraft UAV
RNN	Recurrent Neural Network
R2D	Radian to Degree
s	Seconds
SCSI	Small Computer System Interface
SD	Spatial Disorientation
SECAM	Sequential Couleur a Memoire or Sequential Color with Memory
sin	Sine
std	Standard deviation
tan	Tangent
TVL	Television Lines
UAV	Uninhabited Airborne Vehicle
UAV	Unmanned Aerial Vehicle
USB	Universal Serial Bus
UV	Ultra–Violet
V	Volt
Var	Variance

VGA	Video Graphics Array
VOR	Vestibulo–Ocular Reflex
W	Watts
WGS	World Geodetic System 1984 standard
2D	Two-Dimensional space
3D	Three-Dimensional space
μs	Microsecond

Chapter 1

Introduction

1.1 Aim

The objective of this thesis is to explore the feasibility and advantage of using panoramic vision to determine the attitude of a flying vehicle (e.g. fixed–wing or rotary–wing aircraft) in a cluttered outdoor environment. Whilst a number of studies have shown that visual sensing of the horizon line separating the ground and sky can be achieved, researchers have not looked at th problem for this to be achieved when a vehicle is close to the ground where some parts of the horizon may be obscured by buildings, trees and other occlusions. This work is relevant to small unmanned helicopters and Micro Air Vehicles (MAVs) that might be required to take–off, land, hover and navigate in cluttered environments to fulfil their mission [1].

Sensing attitude is a vital part of any control loop system for autonomous controlling and attitude stabilising the attitude of a flight vehicle [2]. This is particularly the case for a small rotorcraft (e.g. helicopters) in hover as where small changes in its reference attitude can cause rapid changes in fits light trajectory. Traditionally, this sensing performed using inertial sensors that fuse accelerometer and gyroscope measurements, either mechanically or electronically, to provide a stable attitude reference. However, the low–end versions of these devices are highly susceptible to errors introduced by vibration and vehicle dynamics and can become unusable in certain situations or on certain platforms [3]. The use of vision to replace or augment inertial sensing provides a potential solution to these problems. In addition, vision sensors are also considered as passive sensing devices (no energy emittance for the sensing [4]), light–weight and can have low–power consumption which makes them suitable for implementation on MAVs.

The aim of this thesis is to develop a technique for performing visual attitude estimation so that the problems of inertial sensors can be overcome. This work includes the image processing aspects of horizon detection and two different sensor fusion techniques for combining sensory data. Both simulations and flight tests are conducted to confirm the accuracy of the results.

1.2 Motivation

1.2.1 Applications

Aircraft can be categorised into two groups according to two their different structures: fixed–wing or rotary–wing. Due to its unique design, a rotary–wing aircraft can perform different flight modes from those a fixed–wing aircraft is able to achieve. The remarkable capabilities of a Rotorcraft UAVs (RUAVs) are more highlighted when there is a limited area for takeoff and landing, especially when to be used in urban area and fly at low altitude. In addition, the unique features of RUAV , including the ability to fly in any directions, hover with lateral movement, be stationary in the air and take off and land on a small area with no runway requirement, make them useful in limited areas such as urban areas or landing on a ship deck. Military and commercial applications of Unmanned Aerial Vehicles (UAVs, commonly called drone) have experienced unprecedented levels of growth in recent years. UAVs have advantages over manned aircraft, such as:

- they are smaller in size which enables them to take off and land in small areas and be kept in a small hangar, and

- they have the capability to fly and operate either by remote control or autonomously which makes them suitable for undertaking operations dangerous for humans.

In order to operate in a very constrained environment (e.g. indoors), Micro Air Vehicles (MAVs) are proposed which are miniature form of UAV. Due to the reduced MAV structure and size, they are able to operate in areas where normal aircraft or UAVs are unable to operate such such as the confined spaces between buildings [5]. To be remotely or autonomously controlled, the MAV's avionics should be able to provide reliable and accurate information about the aircraft.

1.2.2 Biological Inspiration

The inspiration behind this work comes from biology, in particular, the analysis of how insects can fly so precisely and find their path despite having a very small brain. Different aspects of insects behaviour have been researched for many years, including determining what sensory mechanisms they posses which enable them to control their flight paths. Research studies have shown that they rely more on their vision system than other senses for navigation (e.g. they control their attitude using both visual and mechanical stimulus [6]). Based on experimental results, as a lack of visual information can severely affect an insect's flight capabilities. Visual motion is clearly a key element in insect sensing for flight control.

Visual cues are considered as the natural primary senses for determining orientation. When flying, human pilots rely heavily on a visual attitude reference, either a visible horizon that artificially–projected by an instrument. During visual flight in a clear sky, a pilot controls the plane's attitude with respect to the natural horizon while, during instrument flight in clouds or at night, the artificial–horizon (computed by data sensed from instruments) is displayed on an indicator [7]. Pilots are trained to fly by visual reference to the horizon (whether natural or artificially projected), to control the wing's bank angle and aircraft's pitch attitude. When not using instruments for flying, a pilot is able to maintain a field of view outside of the cockpit, which aids in avoiding collisions. This also allows for more stable flight, as otherwise, pilots tend to rely on the information displayed on other instruments (e.g. airspeed). Throughout history, a pilot being distracted from focusing on the attitude reference (e.g. by suffering Spatial Disorientation [8]). Spacial disorientation (SD) is the person's inability to determine his attitude and orientation in space respect to the Earth or a reference. Due to being in an acceleratory environment, an aircraft pilot can misinterpret the aircraft's real orientation during a bank turn due to centrifugal force, called spacial disorientation, which causes him misinterpret the gravity direction [9].

It has been shown through scientific experiments that changes in the visual patterns presented to insects (image motions or optic flows) dramatically affect the flight control and stability of winged insects (wing–flapping speed [6], flight speed [10]).

An insect's compound eyes comprise many photoreceptors, with a mix of spectral sensitivities [11]. In addition, most winged insects carry receptors called Ocelli which are sensitive to the UV wavelength and are believed to assist with horizon detection [12]. A a UV image provides an enhanced contrast between the sky and ground ([13], see Figure 3.8) that makes horizon detection easier. This motivates the use of UV imagery in this thesis.

Sensor fusion is required to integrate data from different sources such as horizon detection and optic flow to obtain the advantages of different systems in order to overcome the shortcomings of sensor. Insects must fuse different sensory information to gain better understanding of their environment and how to use it for attitude estimation [[14], [6]].

The aim of this thesis is to develop and investigate a novel method for aircraft attitude estimation using a biologically–inspired vision system that mimics the insect vision system. It is implemented and tested on a small rotary–wing aircraft flying at a low altitude which has a significant amount of vibration due to its engine and spinning rotor–blades. The attitude of this aircraft is estimated in two different expressions, attitude angles (roll and pitch) and angular velocities, which are fused using two sensor fusion techniques. The aim is to demonstrate not only the feasibility of this method but its advantages over the existing methods and how it overcomes the existing drawbacks of cheap Inertial Measurement Units (IMUs).

1.3 Contributions of Thesis

The aim of this thesis is to design a new panoramic vision system capable of independently estimating an aircraft's attitude without any assistance from an IMU. By using panoramic images, it is able to estimate its attitude from the horizon and calculate its angular body rates from image motions. Several contributions are provided in this work, with the main one relating to how to process a panoramic image to estimate the attitude of a rotary–wing aircraft in a cluttered environment, which can be summarised as follows:

1. The developing of an imaging system for capturing UV–filtered panoramic images to enhance the sky/ground contrast for horizon–based attitude estimation;
2. The introduction of two novel image thresholding methods for using:
   - sky/ground segmentation for panoramic images, and
   - regional–based thresholding to improve horizon extraction in regions around the sun, affected by the sun luminosity;
3. The creation of a novel method for real–time horizon–based attitude estimation using of panoramic mapping of the horizon when flying:
   - in a cluttered environment,
   - at low altitude, and
   - on rotary–wing aircraft with vibrating platforms.

4. The Calculation of an aircraft's body angular rates from image motions using optic flow to improve attitude estimation through sensor fusion, and

5. The integration of sensory data from the IMU and vision system to improve the attitude estimation.

1.4 Thesis Layout

This thesis contains seven chapters. Chapter 2 provides a review of related work in the literature, covering the existing methods used for visual–based attitude estimation, including horizon–based attitude estimation, optic flow and sensor fusion techniques. Chapter 3 represents an overview of the hardware equipment and software programs used to conduct this work. Chapter 4 contains one of the main parts of this research, image processing approach used to estimate attitude from the horizon. Chapter 5, an optic flow method is introduced as a biological approach for image motion detection for use as a complementary sensory information to improve attitude estimation. Implementations of two sensor fusion techniques are discussed in Chapter 6, with the beneficial aspect of integrating sensors is elaborated. In Chapter 7, the conclusion drawn from achievements of this work are highlighted, and recommendations for future work proposed.

Chapter 2

Literature Review

2.1 Introduction

This chapter summarises literature in the fields related to the objectives of the work. First it explains the attitude presentation. Then it explores different methods of attitude presentation in three–dimensional space and the mathematical background of the attitude representation. In section 2.2, the navigation systems that are used commonly in aerial applications are outlined. In this chapter, we also focus on the application of the inertial–based navigation sensors and global positioning system and explain limitations and then mostly focus on the vision–based attitude determination methods. The application of vision-based attitude estimation is highlighted in section 2.3. Different methods of horizon detection are presented in section 2.4. Omnidirectional vision-based systems are reviewed in section 2.5. Biological approaches for attitude stabilization and biologically-inspired motion detection are explained in the context of insect egomotion determination for navigation. The beneficial aspects of applying biologically inspired vision systems over the conventional method will be elaborated to support the motivation behind this work. Different optic flow estimation techniques are explained in 2.8. The application of signal filtering and sensor fusion techniques are also reviewed.

2.1.1 Attitude Representation

Attitude refers to the position and the orientation of an object in threedimensional space, in relation to a reference frame. Attitude is vital for aviation and particularly important for flight control. The purpose of this thesis is to find a way to explore the feasibility and advantage of using panoramic vision to determine attitude. In order to describe the attitude of a body, it is necessary to describe how the axes that move with the body (body frame) have been rotated from a reference frame that is fixed to the earth. There are a number of standard forms for representing rotation of a rigid body, such as Direction Cosine Matrix (DCM), Euler angles and Quaternion. In this work we will make use of Euler angles, Quaternion and rotation matrix representations which will be briefly discussed in the following.

For this work, the earth fixed reference frame we will be using is the North, East, Down (NED) coordinate system. The NED is a right-handed frame that rotates with the Earth and is used in naval and aerial applications for navigation. The NED Cartesian axes comprise an Xaxis which points to the North pole, a Y axis which points to the East and a Zaxis which points towards the center of the Earth (see Figure 2.1). Other representations are described in [[15], [16]]. The standard aircraft body reference frame we will use is a set of right-handed axes fixed to the aircraft such that the X-axis points towards the nose, the Y-axis points towards the right wing, and the Z-axis points downwards when the aircraft is level.

2.1.2 Euler angles

Euler angles are the most intuitive form of representing the body rotations. They consist of three consecutive rotations that transform a reference frame to a given frame. The transformation matrix from Navigation frame to Body frame is conducted by rotational Euler angles ψ, θ and ϕ about the z, y$'$ and x$''$ axes as shown in Figure 2.2.

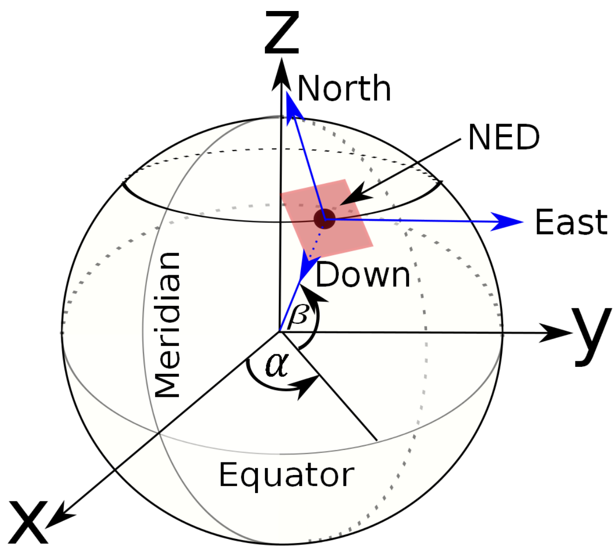


Figure 2.1: Earth–related coordinate systems

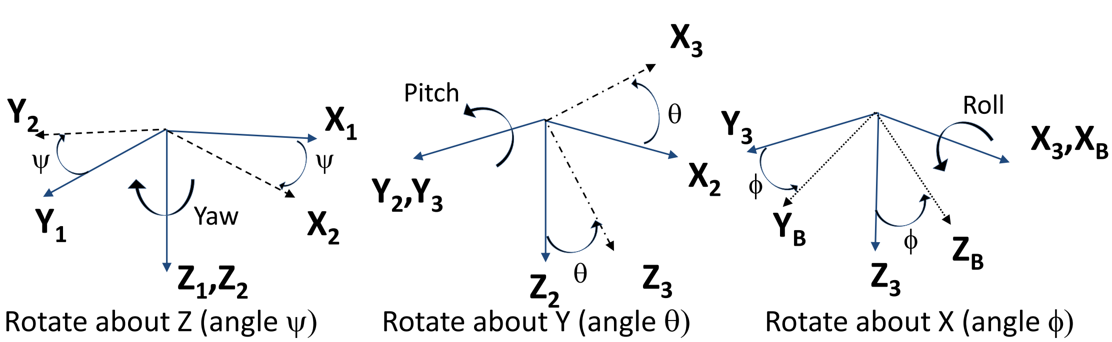


Figure 2.2: Euler Angles

2.1.3 Quaternion

Quaternion is another form of representing rotations. Compared to Euler angles, they are less intuitive but do not exhibit numerical singularity. Quaternion formulation avoids singularity (which is a drawback of Euler formulation of angles) and the differential form of quaternion formulation is numerically preferable for a time–varying coordinate transformation [17]. Quaternions consist of four components (Eq 2.1) which are yielded from the sine and cosine calculation of the three Euler angles (Eq 2.2).

$$q = [q_0 \quad q_1 \quad q_2 \quad q_3]^T \tag{2.1}$$

By combining the quaternion representations of the Euler rotations we get:

$$q = \begin{bmatrix} \cos(\phi/2)\cos(\theta/2)\cos(\psi/2) + \sin(\phi/2)\sin(\theta/2)\sin(\psi/2) \\ \sin(\phi/2)\cos(\theta/2)\cos(\psi/2) - \cos(\phi/2)\sin(\theta/2)\sin(\psi/2) \\ \cos(\phi/2)\cos(\theta/2)\sin(\psi/2) + \sin(\phi/2)\cos(\theta/2)\sin(\psi/2) \\ \cos(\phi/2)\cos(\theta/2)\sin(\psi/2) - \sin(\phi/2)\sin(\theta/2)\cos(\psi/2) \end{bmatrix} \tag{2.2}$$

2.1.4 Rotation Matrix

Three dimensional rotations can be written as the product of three individual rotations about each axis: ϕ about X, θ about Y and ψ about Z axes, as follows.

$$R_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\ \phi & \cos\phi \end{pmatrix} \tag{2.3}$$

$$R_y = \begin{pmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{pmatrix} \tag{2.4}$$

$$R_z = \begin{pmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2.5}$$

So rotations in the three dimensions are:

$$B = \begin{pmatrix} \cos\theta\cos\psi & -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi \\ \cos\theta\sin\psi & \cos\phi\cos\psi + \sin\theta\sin\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{pmatrix} \tag{2.6}$$

For a given rotation represented in ϕ, θ and ψ, a point $[x_1, y_1, z_1]$ in one coordinate systems can be transformed to a point $[x_2, y_2, z_2]$ in another coordinate system by multiplying the rotation matrix B as follows:

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = B \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \tag{2.7}$$

The rotation matrix can be constructed in quaternion form [17]:

$$B = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_2 + q_0q_3) \\ 2(q_1q_2 - q_0q_3) & q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_1q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{2.8}$$

2.2 Navigation Systems

Navigation systems are used to provide the position and direction of an object with respect to a known reference frame. The position of each location on the earth as a point on a sphere can be identified by two numbers, latitude and longitude. Two common devices which are used for navigation are Global Positioning Systems (GPS) and Inertial Measurement Units (IMU).

2.2.1 Inertial Navigation System (INS)

Inertial sensors have been used in many applications where the information about the position and direction of an object is needed, for example providing attitude information in aerospace systems [18]. Inertial Navigation Systems (INS) are used to measure the movement of an object. They are valuable because they are self–contained sensors and do not depend on any external signal such as that from navigation satellites. To calculate the attitude from an INS internal sensors, highly accurate measurements of acceleration and magnetic field are required. The basic principle of inertial navigation is in the Equation 2.9 [19]. The inertial system measures three orthogonal accelerations and estimates velocity and position by single and double integration of sensed acceleration respectively.

$$\Delta_{position} = x = \int \dot{x} dt = \int\int \ddot{x} dt = \int\int a\, dt \tag{2.9}$$

The position of a vehicle with respect to a coordinate frame can be determined by knowing the starting point and the vehicle accelerations and the angle in which the vehicle makes with the reference frame. To calculate the body position from the acceleration vector, the Earth's gravity $\mathbf{g}$ must be subtracted from the sensed acceleration before integration [19]. If the vehicle is in state of unaccelerated motion (i.e. constant velocity), knowing the gravity vector is sufficient to determine the roll and pitch angles.

Accurate calculation of acceleration is a complex problem due to factors such as the rotating Earth (Coriolis and centrifugal forces), that transform Newton's laws to a uniformly rotating frame of reference [15] and the translatory accelerations due to the aircraft's pitching and rolling. Accelerometers detect the combination of the gravitational force and also translational motion due to the vehicle movement. The separation of the true vehicle acceleration from the gravitational force is calculated on–board using the vehicle's known attitude. In such cases, the quantity of non–

gravitational force is exerted on the instrument [20]. Two methods are used to tackle the acceleration calculation complexity down in the earth's horizon plane, stable platform and strap–down platform [21] or sometimes called platform systems [22].

Inertial Navigation Systems (INS) are very complex systems and are affected by possible errors due to the sensors (accelerometers and gyroscopes) or the processing unit that are classified as bounded and unbounded errors. The bounded errors are not time–varying errors but unbounded errors are. For instance, any false bounded acceleration causes an unbounded error in the vehicle's speed and position, due to signal integration. Some of the common errors in INS are detailed from as follows [23]:

- Bias (constant bias in gyroscope and accelerometer): when integrated, the gyro bias causes angular error to grow linearly with time;

- White noise: Uncorrelated zero–mean random noise in the sensor output due to thermo–mechanical noise that perturbed the sensor output and introduces a standard deviation in the sensor outputs in both gyro and accelerometer; and

- Calibration: Any misalignment or errors in scaling factor (in gyro or accelerometer) causes drift in orientation and position calculation.

2.2.2 Global Positioning System (GPS)

INS sensors provide accurate sensory measurement for a short period of time and their measurements drift due to the sensory errors as they are integrated by the attitude equation and cause unbounded error. In this case, a long term correction is necessary to overcome this problem. In contrast to the INS sensors, the Global Positioning System (GPS) provides accurate position for a long time and is used as a complementary device to correct the INS errors and increase the attitude accuracy [22]. The GPS accuracy varies but generally the achievable accuracy is around -/+10 meters [22]. A GPS device can suffer from errors from the satellite and errors from the signal reception including:

- The position and the clock of the satellite in orbit may drift from its predefined position over time, when this happens the satellite corrects its position and clock in reference to the ground station. These types of corrections are done on a daily basis [22],

- Low position update frequency (usually 1 Hz),

- GPS relies on the external satellite signal and needs sky visibility. Any satellite signal blockage interrupts normal GPS operation, and

- The GPS accuracy highly depends on the number of satellites visible to the receiver.

2.3 Vision–Based Attitude Estimation

Accurate attitude determination is critical for an autopilot system to control an aircraft autonomously. Generally, inertial navigation systems (INS) are used for attitude determination, often with complimentary correction from a Global Position System (GPS). However despite being widely used, INS/GPS suffer from a number of errors [24]. A GPS suffers from signal blockage, low update frequency, and sensitivity to satellite orbits, while an INS suffers from white noise, random walk, temperature effects, calibration errors, and drifts in gyroscope and bias [23]. Accelerometer bias and gyroscope drift are the main causes of INS drift in position and velocity [16]. Any bias error in the inertial sensors will cause the attitude to drift over the time due to the integration of the error. To detect and mitigate errors in the attitude estimation, integrated systems are used to fuse attitude information from different sources to get the advantage of several systems. In aerial vehicles, having a backup measurement is critical when a sensor fails to provide reliable measurements. Due to major problems in GPS and INS systems (especially cheap ones), an alternative method of attitude determination is necessary.

As the horizon is such a valuable attitude reference to a human pilot, it is suggested that an estimate of pitch and roll could be obtained through machine vision and processing of the horizon. Under instrument flight rules, where the horizon is not available, an artificial horizon is used as a substitute reference [25]. Human pilots are also trained to control the attitude of the aircraft in relation to the horizon (when clear visibility exists) rather than relying on instrument. Controlling the aircraft during instrument flight (using instrument for flight control) fairly requires more precise techniques than visual flight. Additionally similarity between the visual and instrument flights is that, in visual flights, the pilot controls the attitude with respect to the horizon and in instrument flight, the artificial horizon (computed by sensed data from instruments) is displayed on the indicator [7].

Visual information offers valuable information about the aircraft movement that can lead to estimates in aircraft attitude. Visually detected attitude is not affected by vehicle acceleration during manoeuvring nor vibration, unlike inertial data. The navigational application of vision based systems has been investigated by several authors for use in aerial applications such as sky/ground segmentation and object detection [[26], [27]], parametric features tracking [28], MAV autonomy using horizon extraction [29], angle determination using horizon line for a small UAV [30], autonomous flight control using vision–based system [31] and integrated vision–based attitude estimation[32].

The horizon appears as a straight line that separates the ground from the sky and is the reference point for the horizon tracking algorithm [33]. Using a single camera lens with limited view angles causes vulnerability to horizon visibility, especially when the aircraft is heading up or down and the horizon is out of view. For example Ettinger et al [34] designed a vision system using a forward–looking camera, offering flight stability for a MAV using the horizon. Another vision system was designed by Dusha [32] that was able to estimate the attitude. To have the visible horizon, Mondragon et al [35] used a catadioptric imaging system. The extracted horizon points from the panoramic image were reconstructed on a unitary sphere. Then, the roll and pitch angles were calculated from the norm vector of a plane which was calculated from the horizon points.

The main reason behind using a vision–based systems with conventional measuring systems is not only to integrate different measurement to outperform the estimation, but also, in most cases, vision–based measurements are used to serve as a reference standpoint or backup measurement.

2.4 Horizon Detection

The basic meaning of horizon detection is dividing image into two regions, sky and ground, and the line that separates these regions is the horizon. To determine the attitude in a vision system, typically a wide–angle camera captures a complete view of the environment including the horizon. In this context the quality of the visual horizon is based on effective separation of sky from ground.

Automated image segmentation for horizon determination and extraction can be computationally expensive and challenging to implement in real–time applications because of a wide–range of inherent variations in image architecture (e.g. image contrast and pixel intensity). Some studies have been conducted to enhance the quality of horizon detection to tackle this problem. For instance, McGee et al [36] used a Support Vector Machine [SVM, [37][38]] that needed 120ms to process a 320×240 image resolution using a PC104 with 700MHz Pentium 3 processor and Hough transform [39] that required 200–600ms for sky/ground line extraction. Dusha [32] used morphological smoothing [40] on each channel (red, green, and blue, see Figure 2.3), taken from a forward–looking camera, that required around 1.74s in MATLAB and 66ms using OpenCV for a 352×288 image resolution using a Pentium–3GHz (125ms computation time when implemented using C/OpenCv [41]) and Sobel operator for horizon extraction that takes 190ms. In Dushas work, two fixed–wing aircraft was used at high altitude and his method showed standard deviation of 0.71○ and 0.42○ (in clear condition) and 1.75○ and 1.79○ (in obscured condition) for roll and pitch respectively.. Todorovic [42] used multiscale linear discriminant analysis (MLDA) that reguired 33ms to process a 128×128 image resolution using 2.4GHz x86 processor. Cornel [43] used a formula

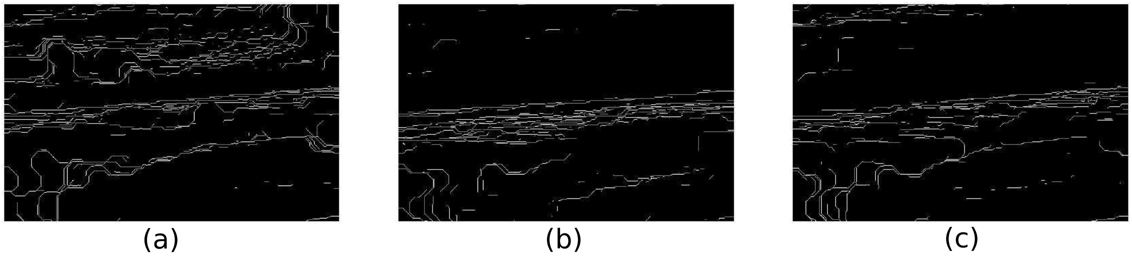


Figure 2.3: (a): Edge detection on Red channel, (b): Edge detection on Green channel, (c): Edge detection on Blue channel.

$(3B^2/(R+G+B))$ to convert the values of red, green and blue channels (RGB) of each pixel into one value to discriminate the sky and the ground with low computation complexity. Otsu [44] was then used for finding the threshold value [44]. Bao et al [29] introduced a method of horizon extraction using Laplacian of Gaussian filter (LOG). LOG is a derivative filter that was used to enhance rapid intensity changes so the edges (as a sign of rapid intensity changes) are more highlighted in the image. A global threshold method was used to convert the image to binary image for the edge detection. By studying 124 representative images, Thurrowgood et al [45] developed Equation 2.10 to transform the Red, Green and Blue (RGB) channels into a single paramater. The image threshold was calculated using Gaussian distribution. Then, the horizon points were mapped onto a unit sphere and the attitude. The attitude was estimated from a LSE–estimated plane using the horizon points with standard deviation of 1.39° and 1.69° for roll and pitch angles respectively.

$$C = -1.16R + 0.363G + 1.43B - 82.3 \tag{2.10}$$

One of the main problem in real–time implementation of sky/ground segmentation is the high amount of computation that is required for the image or edge enhancement. An inefficient sky/ground segmentation method may result in the classification and retention of false edges making the selection of true horizon edges computationally more difficult. In such cases, another method is needed to get the most probable horizon edges, such as using color information [41] or tracking the location of the horizon using optic flow [32].

The attitude can be measured by sensors which are pointed to different locations in space, each one with a particular wavelength response. This method can be considered as a basic reference but cannot give a reliable and accurate measurement because of lack of any other extracted image information. Chahl et al presented a bio–inspired horizon sensor as an attitude reference system using ultra violet and green light for Mars exploration [46]. The system mimicked the Ocelli analysing the environment in UV–green spectrum and was designed for environments like Mars, where the gravity is weak and another source of information is needed to determine the attitude.

2.5 Omnidirectional Vision–Based systems

Standard lenses have limited field of view that sometimes causes a useful part of the image to be only partially visible or completely missing. To overcome this limitation:

- the camera should be actuated simultaneously to the designated area, or
- using several cameras pointing in different directions [47], or
- using a camera with a panoramic lens.

The application of a single camera for vision–based control of UAVs has been studied for many years. Vision systems with a single camera have been used for autonomous landing of UAVs [[48], [49], [50]]. To increase the view angle, Saul Thurrowgood et al [51] demonstrated a vision system for visual horizon detection using a dual–fish eye camera system to produce 360° images.

The first hand–cranked panoramic camera was invented by Putchburger in Australia in 1843 that was able to capture 150° images [52]. A panoramic capturing vision system was introduced and patented in 1970 by Rees [53], consisting of a camera and a hyperboloid mirror. Usually omnidirectional vision consists of a convex mirror with a projective camera that has the capability of capturing 360° images [54]. In contrast to using several cameras, an omnidirectional lens is able to capture a 360 degree semi–spherical field of view with one camera [55], enabling the vision system to track features without actuating the camera. The advantage of using omnidirectional lens for acquiring a panoramic image with a single shot makes it suitable for tracking the movement of a UAV without actuating the camera.

The omnidirectional lenses have been used in a number of tasks to provide a complete view. Ability to provide a full view of surrounding is a key point of omnidirectional lenses cause all features can be capture on one image and each feature can be used independently for any purposes. To use this capability, Hrabar and Sukhatme [56] designed an omnidirectional vision system for an autonomous helicopter to be used for several target tracking. Ground based robots have also used omnidirectional visual sensors for target tracking [57] and path following [58].

Using omnidirectional vision for UAV attitude estimation has previously been investigated on platforms such as airplane [59], fixed–wing aircraft [41] and rotorcraft [35]. Panoramic vision has a number of unique advantages over using a single camera when the horizon is considered as a reference for UAV attitude estimation, including:

- Complete view of surrounding with 360 degree visibility;
- Complete or partial horizon presentation in the image regardless of the UAV's attitude (less horizon visibility when it is located on the lens blind spot); and
- Independent roll and pitch angle estimation due to capturing a complete view of the horizon.

Roll and pitch angles can be calculated directly from a panoramic image because the corresponding horizon line related to these angles is in the image [59]. With a single camera, only a portion of the horizon line is visible and the horizon can also be out of view depending on the attitude and the viewing angle of of the camera lens. Most panoramic imaging devices have major problems, including:

- Blockage to a part of the projected image by the camera or lens holding arrangements,

- Possible mechanical damage to the reflective surface (should be physically protected), and

- Possible image distortion due to enclosing the panoramic reflective surface in a glass or Perspex cylinder (burdening image processing).

Sturzl et al [60] described a novel catadioptric imaging system, a mirror–lens combination that avoids the above problems in addition to being rugged, consists of an almost spherical solid Perspex block with parametric curved shape. Because of the curved shape of the reflective surface, the projected image suffers from different deformations such as stretching, bending. The distortion depends on the size of the object, viewing angle and elevational changes. An object that is projected by a panoramic lens suffers from distortion and has different size and shape due tot the lens distortion. The image distortion can be corrected using image filtering. For example, Chahl and Srinivasan [61], presented a technique for homogeneous spatial filtering of panoramic images with non–elevational gain to reduce stretching and bending which is presented on the image.

2.6 Biological approaches for attitude stabilization

Insects rely mostly on their advanced visual system. Insects use visual cues in the UV–Green spectrum for navigation [62]. In addition to having compound eyes, most winged insects carry receptors called Ocelli which are associated to horizon detection (suggested by German zoologist Rudolf Hesse) and are sensitive to UV wavelengths [12]. The Ocelli are located on the top of the head with forwarded looking median Ocellus and two side–ways looking lateral Ocelli. Insect eyes are equipped with highly sensitive UV receptors [63] to exploit the difference in the UV intensities to separate sky from background. Ocelli are shown to be critical to flight stability in [13] as rapid changes in the horizon can be sensed by them (see Figure 2.4). Examination of the physical structure and functionality of Locust Ocelli by Martin Wilson in 1978 [13] revealed that the Ocelli was sensitive to UV wavelengths. Bee compound eyes are equipped with

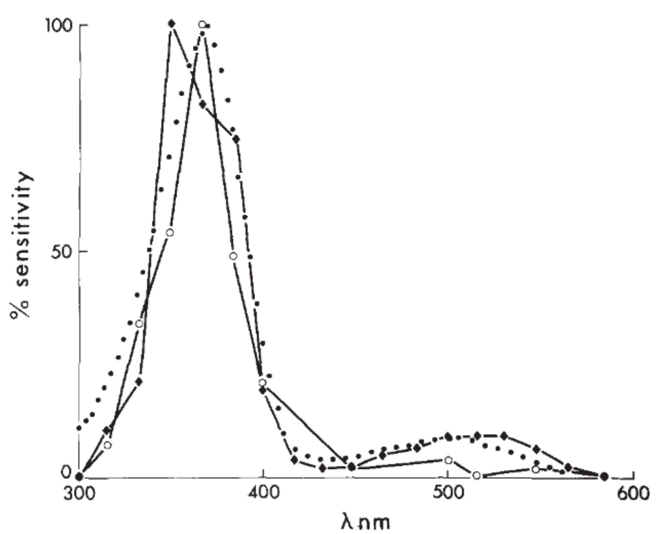


Figure 2.4: L neurons spectral sensitivity in UV and green wavelengths.

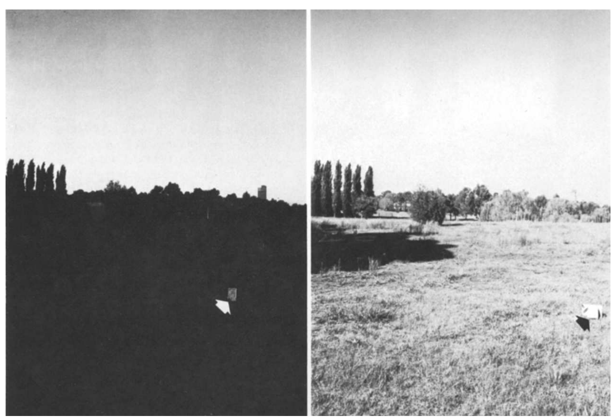


Figure 2.5: Left image: image in UV wavelength, right: image in green wavelength.

receptors with sensitivity in UV, blue and green [63] to separate different landmark properties with respect to different wavelengths. These receptors have sensitive peaks to 350 nm (may be as low 340 nm and as high 380nm) for UV, 440nm for blue and 540nm for green [64]. In UV wavelengths, most objects and surfaces on the ground are quite dark, while the sky is bright, enhancing horizon detection (see Figures 2.4 [13] and 2.5). Figure 2.5 shows two images taken from ground level of a natural scene which contain ground and sky. One image shows the intensity in UV and the other shows the intensity in green light. Although the image quality of the Figure 2.5 is degraded and the image texture is poor, Figure 2.5 shows much easier it would be to separate the ground and sky from the UV image.

2.7 Biologically–Inspired Motion Detection

Insects have relatively small brains (honeybee 960,000 neurons [65]) compared to human brains (170.68±13.86 billion cells (aged 50 to 70) [66]), yet have effective flight control capabilities, and exhibit robust obstacle avoidance behaviour, even in complex environments. Recent biological studies on insects have revealed that the apparent image motions of visual features (optic flow) is one mechanism that is used to achieve this [[10], [67]].

Douglass and Strausfeld studied the anatomical organisation of the fly optic lobes on motion detection [68]. They found that visual movements are sensed by the fluctuation of light intensity received by various photoreceptors. Medulla and lobula cells are arrays of sensitive neurons to visual motions [[69],[68]]. The lobula plate tangential cells (LPTCs) have different motion sensitivity which provide different information to the insect, including horizontal (HS), centrifugal horizontal (CH), and vertical (VS) cells [[69],[70]].

Baird et al [10] investigated how night–active insects use optic flow in dim light conditions compared to day–active insects. They chose two different flying insects to examine the effect of visual motions on the insect's flight in an experimental perspex tunnel (14cm width, 14.5cm height, 50cm length). To examine different visual scenarios, they used different visual patterns (check and stripe) with different light intensities on the tunnel's wall. Their research showed day–active and night–active insects react differently to different tunnel's visual patterns by keeping different distances to the wall and also the fact that day–active insects rely on vision cues for flight control more than night–active insects as their ground speed reduce in dim light. To investigate the effect of optic flow on insect flight, Baird et al in [67] investigated the effect of different visual patterns on Bumblebees' groundspeed in tunnels with different widths (15cm and 30cm). To prove that the bee primarily relies on optic flow to regulate the groundspeed, the tunnel pattern changed from chequerboard pattern to axial stripe. The experiment revealed that the bees reduce groundspeed when the pattern is changed from chequerboard pattern to axial stripe. Their research shows how insects interpret optical flow of different visual patterns for the flight control.

Recent investigations also show that birds use visual information for flying. Partha S. Bhagavatula et al, [71] investigated how birds use optic flow to fly through narrow passages without collision. They looked at how a bird (Melopsittacus undulatus) can fly through a narrow passage marked with horizontal and vertical strips. The investigation showed that the birds rely mainly on optic flow changes from visual cues for flight control and for flying through narrow passages without collisions, by balancing the speed of image changes from both eyes.

The extraordinary flight capabilities of insects motivated scientists to mimic insects' vision processing system to solve engineering tasks and to reduce the image processing load. Dusha [32] used the optic flow to calculate the body rates and also to estimate the position of the best horizon line for an integrated horizon–based attitude estimation. Attitude stabilization mechanism based on biological sensors has been proposed such as Piezo–actuated biomimetic angular rate sensors [72], biomimetic Ocelli [73], Biomimetic Visual Sensing [74] and fly–Ocelli inspired attitude controller by Schenato et al [75].

In addition to optic flow, flying insects integrate sensory information from non-visual sources to improve their motion estimation. Some insects have a pair of organs called Halteres (evolutionarily modified wings with small knobbed structures) with functionality like gyroscopes. They inform insects about rotary movements during flight. Halteres react to sudden changes in the flight direction and send a signal to the control system to stabilize the flight. Huston et al [76] investigated the effect of visual information and Halteres stimulation on a blowfly's neck movements. Their study using extracellular and intracellular recordings which revealed that neck motor neurons (NMNs) do not show a potential response to some visual stimuli when Halteres are not stimulated. NMNs are potentially affected by visual inputs when Halteres are also active. The rule of Halteres during a flight has been a challenge and different studies have shown different behaviours. Hengstenberg [14] studied how a fly fuses different sources of sensory information during gaze movement changes when walking or flying. His quantitative observation demonstrated that, although body motion is perceived by Halters when walking, the insect controls the head's posture by pattern motion, dorsal light and the gravity direction. Halteres give pure information about body rotation even if the surrounding visual information is changing. A fly fuses different sensory information to have a better understanding about its body or flight condition. Rosner [77] studied the effect of a blowfly's Halter removal on the fly's head movements (such as pulling them out or covering them with bees wax). They observed that the fly moves its head in relation to the visual input. But after Haltere removing, the head fluctuated more and the strength of the head's jitter was reduced.

Sherman and Dickinson [6] showed the complimentary relationship between the compound eyes and the Halteres which enhances the aerodynamic performance. Due to different mechanisms of transduction (photo compound eye) and mechano (Halteres transduction), Halteres respond to angular velocity faster than visual systems. For the experiment on 80 fruit flies, a flight simulator was used to create visual rotations alongside mechanical pitching and roll movement. Their study showed that the wing-beat amplitude and frequency is higher when visual and mechanical movements were applied. When visual and mechanical stimuli are present, the fly strikes its wings harder (in terms of amplitude and frequency in the opposite way of rotation) so that its body gets back to the stable–level condition. The study also showed how a fly integrates different sensory information from different organs for flight control. It was shown by biological researchers that the vision system and the Halteres are complimentary in these insects with Halteres, as they increase the frequency bandwidth coverage and and provide better environmental understanding.

2.8 Optic Flow Estimation Techniques

Many methods have been proposed for estimating visual motions. The aim of this section is not to provide an extensive review of all existing optical flow techniques, but rather to highlight key properties of general optic flow techniques. A quantitative comparison of some existing optic flow methods was studied by Barron in [78] which is summarised below. Optic flow techniques can be categorised into four classes:

- Gradient–based: Compute the image velocity from the derivative of image intensity or filtered images (e.g. Nagel [79], Horn and Schunck [80], Lucas and Kanade [[81], [82]]),

- Region–based: Compute the image velocity as a shifted displacement in the way two images are matched in different times (e.g. [83], Anadan [[84], [85]] and singh [[86]),

- Energy–based: Compute the displacement based on the output energy of a filter (e.g. Heeger [87], [88]), least–sequared fit of energy using gober–energy filter), and

- Phase–based: Compute the image velocity based on the phase behaviour of band–pass filter(e.g. Waxman et al [89], Fleet and Jepson [90]).

In Barron's evaluation, the phase–based method produced results with greatest accuracy, but was very computationally expensive (large number of filters), sensitive to temporal aliasing (because of frequency tuning of filters) and had issues regarding confidence measures (removing unwanted poor velocity estimates).

The first–order local differential method (Lucas and Kanade) also showed reliable results. Noticeable differences were observed from the first–order and the higher order pixel differencing. The spatiotemporal smoothing had a positive effect on the velocity estimation improvement by removing temporal aliasing. Barron found local methods to be more efficient, as well as more robust and accurate to the noise than the global smoothness method.

Although region–based matching produced poorer results than differential method, it has the ability to estimate sub–pixel displacement with high–speed image translations and may perform better if aliasing exists in the image acquisition process. It is based on finding the best match of image displacement of two images with a shift reference (shifting window) and calculating the sum of squared differences (SSD) to maximise similarity.

Energy–based method did not show as reliable results as other techniques. Energy–based and phase–based methods require intensive computations and this property makes them unsuitable for robotic applications.

This thesis uses the Image Interpolation Algorithm (I2A) that was developed by Srinivasan [91]. This technique is a non–iterative process that computes image movements (translation and rotation) throughout the interpolation of a set of reference images by comparing the intensity of a patch with six shifted patches. The image movement estimation is based on the fact that during a displacement, the deformation of two images is linear and continuous. The image displacement in the 2D dimension is calculated from six reference frames that were shifted by a reference shift. The estimation was robust in calculating displacement using noisy images and dependent on the shift value and estimation deviates from correct values when the displacement exceeded the shift value. So the reference displacement should be defined according to the expected image displacement.

Optic flow estimation has also been applied to different engineering applications. One of the challenging parts in using aerial vehicles is landing. While landing a manned aircraft, the pilot mainly relies on vision, alongside supplementary sensory information such as the aircraft's attitude, altitude and sometime radar. The landing becomes more difficult when the airborne vehicle is unmanned and needs some sort of autonomy. Chahl et al, [92], tested the landing strategy used by honeybees and applied it to uninhabited airborne vehicles (UAVs). Chahl described how the trajectories of landing show that horizontal speed is proportional to the height (from Srinivasan et al [93]). When the bees approach the touchdown point, they keep a s shallow angle with constant angular velocity. Following a constant angular velocity from the image (captured from eyes) ensures that as the height is reducing, the speed is proportionally decreasing. This strategy was implemented on a fixed–wing aircraft with optic flow estimation implemented in real time using Image Interpolation Algorithm (I^2A, [91]). The control system controls the pitch rate to control the height respect to the optic flow rate. The experiment showed that a constant optic flow rate causes the aircraft altitude to reduce linearly, and demonstrated that an appropriate choice of optic flow rate is needed to have enough speed at touchdown point.

2.9 Integrated Systems

Inertial measurement systems are a common approach for calculating aircraft attitude based on integrating angular rates to obtain changes in angles. In some conditions, inertial sensors are unable to find accurate angles due to the error integration and lack of a reliable attitude reference. Sensor fusion techniques are used in navigation systems to integrate data from different sources, such as INS, GPS and vision, to get the advantages of different systems to overcome the sensor shortcomings. There are different techniques to fuse data from different sources. Among them, Kalman Filter and Artificial Neural Networks are widely used in prediction of non–linear systems due to their flexibility and simplicity in modeling.

2.9.1 Artificial Neural Networks

An Artificial Neural Network is a computational structure which is inspired by the central nervous system of animals. It consists of interconnected neurons with an inside activation function which enables the network to learn. The cognition capability of nervous systems has enabled animals to learn their reaction to different stimuli during their evolutionary life. Artificial Neural Networks (ANN) can be implemented in hardware and software as a computational structure to mimic a task. The first people to implement an electronic neural network was McCulloch and Pitts in 1943. A good overview of the ANNs are from the book written by Mandic and Chambers [94] which is the basis of the summary of this section.

The structure of a typical ANN is shown in Figure 2.7. An ANN consists of an input layer, a hidden layer (or more hidden layers) and an output layer. Three parameters that generally define an ANN are:

1. A paradigm for neurons interconnection between different layers,
2. The learning algorithm for updating interconnection weights, and
3. The neuron's activation function for neuron's adaptation to mapping its input to its output.

A non–linear system can be modelled with a structure that consists of functions to represent the non–linear part of the system. Activation function has a key influence of the performance of an ANN and combination of linear and non–linear activation functions can be used to model a system. A system can be considered as an unknown system to be approximated from a set of data. Non–linear Sigmoidal functions are the most commonly used activation functions. However, the ANN based on the sigmoidal transfer function preferable to others, there is still not a strong justification behind it [94]. A sigmoidal function is an s–shape function with a different saturation value. A logistic sigmoidal function (unipolar function) is defined as:

$$\varphi(v) = \frac{1}{1 + e^{(-ax)}} \tag{2.11}$$

A typical transfer function is shown in Figure 2.6. The saturation value can be modified, for example for Figure 2.6, the values of saturation are 0 and 1. It is important that the data falls within the saturation range of the activation function. To have the data in

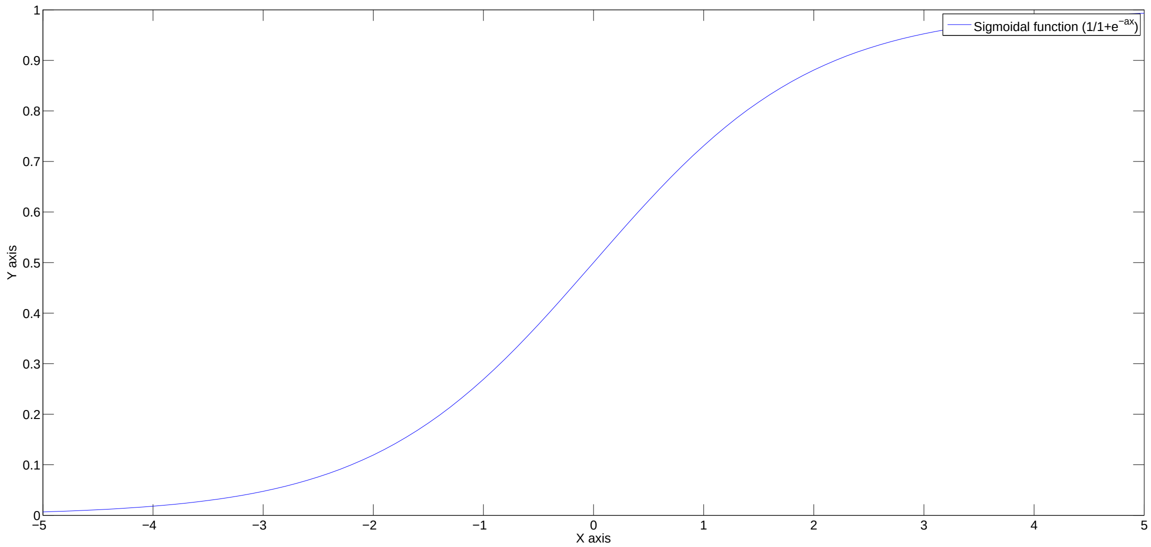


Figure 2.6: Sigmoidal function

the saturation range, data can be normalized using mean value and standard deviation as shown in Equations 2.12 and 2.13 for normalization to μ=0 and std=1.

$$\mu = \frac{\sum_{i=1}^{N} x_i}{N} \tag{2.12}$$

$$std = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \mu)^2}{N}} \tag{2.13}$$

In the real world, most signals are generated or measured by a non–linear system and it is crucial to have a non–linear model of the system to achieve an acceptable performance from a system estimator. Additionally, the parameters of the system's model should be identified and validated in an appropriate way to work as a precise system predictor. ANNs have the ability to map a non–linear system providing a set of input and output data. Two main types of ANNs are shown in Figure 2.7, feedforward network (2.7a) and recurrent Neural Network (2.7c). A multilayer feedfoward network prediction equation for the network in figure 2.7a can be represented as:

$$y(k) = \Phi(X_1, ...X_n) \tag{2.14}$$

In the Figure, Φ represents a non–linear model of the system. The model of a neuron is shown in Figure 2.7b, [95]. The mathematical model of the neuron is defined in Equations 2.15 and 2.16 (equation representation is adopted from [96]):

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \tag{2.15}$$

$$y_k = \varphi(u_k + b_k) \tag{2.16}$$

A multilayer feedfoward network is able to model any measurable function in a satisfactory way throughout learning and can be used an universal approximator [97]. It was inferred in [97] that any lack of accuracies in mapping a function arise from inadequate learning data, or lack of decisive relationship between input and output or the number of units in the hidden layers. Generally, a recurrent network is a modified feedforward network with feedbacks (local, global or both). A feedfoward network captures the dynamics of a system using training data, but for a more complex system, a recurrent network has better performance when the system depends not only the input, but also on the previous history of the signal. A recurrent network can have local and global feedback (Figure 2.7c). The feedback connects a delayed version of output back into the input and introduces a state memory to the network to make it appropriate for prediction. A local feedback is a delayed feedback from the output of neurons back to its input. A global feedback is achieved when the output of the network connects to its input. The mathematical model of Recurrent Neural Net (RNN) as a predictor with global feedback can be represented in Equation 2.17.

$$y_k = \Phi(X_1, ..., X_n, \hat{e}(k-1), ..., \hat{e}(k-p)) \tag{2.17}$$

Where Φ is representing non–linear model of the system and $\hat{e}$ is defined as:

$$\hat{e}(k-p) = y(k-p) - \widehat{y}(k-p),\ j = 1, ..., p \tag{2.18}$$

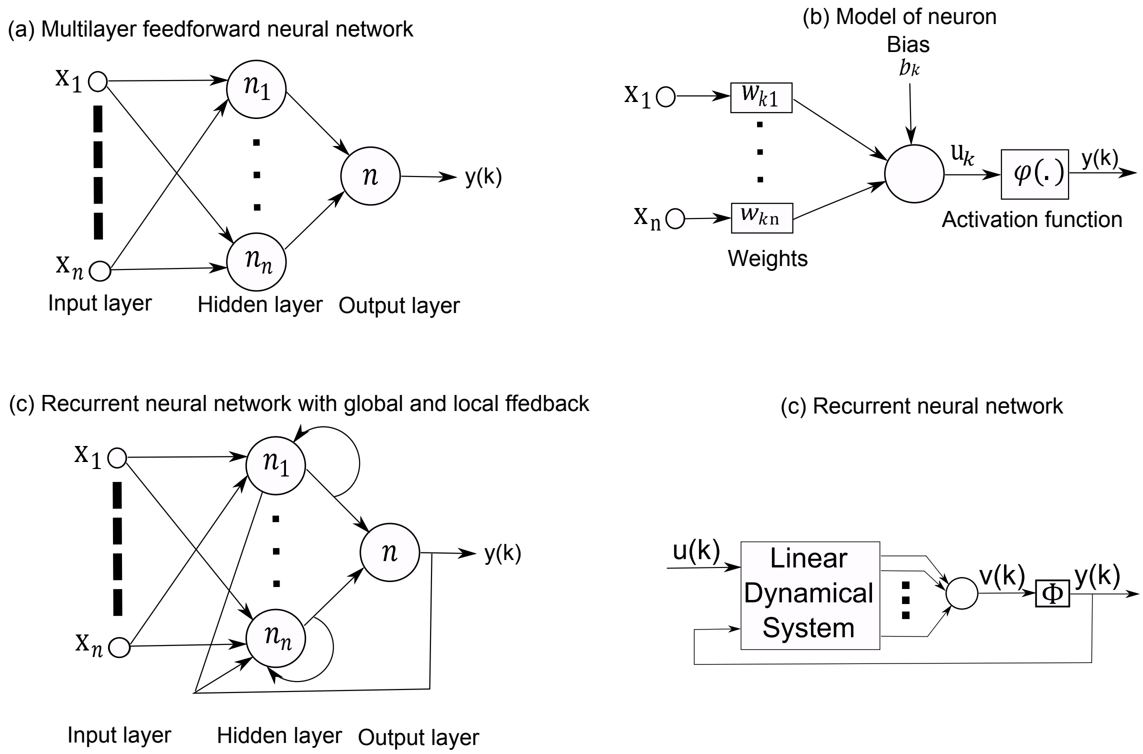


Figure 2.7: Neural Networks

A non–linear system can be represented as an ANN that is considered as a black box that is represents (mimic) a non–linear system behaviour with unknown dynamics whose functionality is defined by the training data from the underlying system. A recurrent neural network architecture with output feedback is shown in Figure 2.7d.

An ANN can be trained in two different ways, supervised or unsupervised. In supervised, input and output data (target data) is used. The aim is to find the system function f:X→Y where (x,y), x∈X and y∈Y are the data fed into the network. The mean–squared error is commonly used as the cost in the training algorithm which calculates the cost. Backpropagation is one of the supervised methods which uses the least–squared error to optimise the system estimation, by minimising the error of the ANN output and the targeted data. It first computes the estimation in the forward direction to the output, then calculates the error and propagates it from the output layer to the input layer to update the weights. It stops the training when the error reaches a sufficiently small value. However reaching this stage might be time–consuming for converging the value of the error of a complex systems to the defined value. The Levenberg–Marquardt training algorithm was developed by Levenberg [98] and Marquardt [99], and is a solution for minimising non–linear functions. The Levenberg–Marquardt (LM) algorithm is used for training Neural Networks using gradient descent and GaussNewton methods. The combination of two algorithms makes LM one of the most efficient training algorithms [100] and is claimed to be the fastest method to train an ANN with a large amount of training data [101]. LM is used as a supervised technique to train all the networks in this work.

Neural Networks have been studied for many years and many structure and learning methods have been proposed and tested. Artificial Neural Networks have also been tested empirically and applied to sensor fusion. The ANN's unique feature for modelling and prediction of a non–linear system makes it suitable when a mathematical or stochastic model of a system is highly non–linear, unknown or not available. A mathematical model of a system can be accurate enough for prediction when there is no noise or uncertainties incorporated in the system. ANNs can be a solution for modelling a system with high–order non–linearity when other methods are incapable of accurately moderating a complex system. Artificial Neural Networks are inspired by animal nervous systems that are capable of modelling a system throughout input/output mapping. The Artificial Neural Networks have been applied for data fusion to improve the estimation when a non–linear system is complicated and can not be represented by a stochastic model. Charles C. Chang and Kai–Tai Song [102] designed an ANN with feed–forward network for data integration using a pair of ultrasonic sensors for environment perception. The ANN is used due to it better estimation of ultrasonic images. Sebastian Thrun [103] used neural networks for grid–based map learning using integrated sonar measurements (16 or 24 sensors) for indoor mobile navigation. There has not been much work done on application of Artificial Neural Networks on sensor fusion for navigational systems. The Kalman Filter has been used extensively for estimation and filtering, but it is unable to accurately estimate the states because of the possibility of inaccurate process model. Applying advanced data fusion technique to such process estimation can solve the existing filtering problems. Sharaf et al did INS/GPS integration using radial basis function ANNs [104]. The ANN is trained to estimate the accurate position and the INS error. The ANN's input is the INS position data and the output is the position error. The ANN is trained with GPS data. The trained ANN is able to estimate the INS error during the GPS outages with error up to 0.7m during simulated signal outage. Naser El–Sheimy et al [105] proposed a method of INS/GPS integration for vehicular navigation using feed–forward ANNs to fuse uncompensated INS and GPS measurements. In this method, the position and velocity are estimated by trained ANN parameters during GPS signal blockage.

2.9.2 Kalman Filter

The Kalman Filter, introduced by Rudolf Kalman in 1960 [106], comprises a recursive filter based on least–squares error. Kalman Filters are Bayesian filter [107] that estimates a system's states provided measurement with Gaussian noise. The Kalman Filter can provide an optimal estimate of the states from noisy data when linear equations of a linear dynamical system are expressed in state–space form. Extended Kalman Filter (EKF) was invented for use with a system with non–linear dynamics when the measurement may not be a linear function of the state equation, [108]. The (EKF) was proposed by Schmidt in 1970 [109]. The state update equations of EKF and relationship between the measurements and the states are non–linear. In an EKF, the states are linearized by fundamental matrix to be used in discrete sampling time. Fundamental matrix (or transition matrix) is a state matrix to propagate the time–inaviant states from the time t_0 to time t [108](will be explained in chapter 6). Two ways of calculating the fundamental matrix are Laplace transform and Taylor–series expansion (see [108] for more). An EKF is a linear solution for a non–linear system with Gaussian noise characteristics.

Kalman Filter has been widely applied to navigation systems for many years to exploit the advantages of fusing navigation systems to overcome their errors. Several errors affect inertial navigation accuracy such as error in initial state conditions, installation error, instrument error. In such cases, inertial alignment is a key point to reduce drift integration, because three gyroscopes and accelerometers are never perfectly orthogonal. The error of alignment can be corrected using a Kalman Filter. And Fuquan et al showed that initial alignment of inertial navigation system can be done by Kalman Filter. Fuquan et al [110] presents a Kalman Filter to estimate the state of errors in a strap down system. The advantage of using a Kalman Filter is its ability to predict the next timestep based on the history of the previous timesteps. This occurs as it updates itself with every incoming observation, particularly when its signal is mixed with random white noise.

Some research has been done on signal integration of different devices such as vision–based and INS, and it has shown better results for attitude estimation because the position and attitude are observable. For example, the work presented by Dumien Dusha and Luis Mejies [111] involves a method for integrating GPS estimates with visual odometry position. Kalman Filter was used in the process of signal integration to estimate the dynamics process error. Visual horizon can be considered as a complimentary reference for UAV attitude control purposes. The attitude which is estimated from the vision can be deployed as an additional input to compute the precise attitude of the UAV and also to compare the performance of IMU and GPS. Damien Dusha [41] et al used an EKF for tracking the best possible horizon line for attitude estimation using optic flow method [[112], [113]] to calculate the body rate from the horizon. The main reason behind using the horizon as an attitude reference if that, the position and the distance of the horizon is not influenced by the aircraft horizontal movements and can also be used as a good reference for calculating the body rotation rates.

Allen D et al, [114], presented an Extended Kalman Filter (EKF) approach for the vision–aided inertial navigation system of a UAV with the ability to estimate the position and attitude using IMU, camera, and magnetometer.

Gupta and Brennan in [115] used a discrete extended Kalman Filter to estimate the vehicle kinematic orientation and to fuse vision and IMU measurements to correct inertial error integration and also to estimate inertial sensor biases.

2.9.3 Summary

The application of vision systems for aircraft attitude estimation has been investigated by many authors. The advantage of using vision systems in attitude estimation has been highlighted in the literature. Many authors have used a single camera with a normal lens that introduces some shortcomings due to the lens's limited angle of view. A normal lens has a limited visual coverage, and tracking of designated features therefore fails when the they are located out of view. For an aerial vehicle with six degree of freedom (DOF) rotations, panoramic vision is a better option than a single pointed camera. Some research has been done on the application of panoramic vision for aircraft attitude estimation as compared with normal vision systems. Although the panoramic vision provides a full environment visibility using a single camera, the image that is captured by a panoramic lens suffers from the possible lens distortion and reflective non–linearity. A main part of horizon–referenced attitude estimation is a precise sky/ground segmentation. Any lack of accuracy in sky/ground segmentation directly affects the horizon extraction and the attitude estimation. To calculate the attitude from the horizon, authors have considered the horizon as a line (when using a camera with normal lens) or as a circle on the 3D plane (when using a panoramic lens). Considering all 3D horizon points to estimate the attitude results in easier estimation as the 3D plane coefficients can be estimated by a simple regression method such as LSE. Sensor fusion is a method to integrate different measurements from different sources to overcome the shortcomings of each source. Two sensor fusion techniques are proposed, one with mathematical background and another with biological background. Both will be used and compared with each to show the advantage of using sensor fusion in minimising the shortcomings of each system and improving the accuracy of attitude estimation.

Chapter 3

System Overview

This chapter provides an overview of different parts of the test bed used to conduct this research with each part of the hardware and software described separately including:

- development of the imaging system and the effect of each of its parts on the final image,
- the rotary–wing aircraft (Eagle) with its avionics and communication devices, and
- A PC104 computer system with a video frame grabber.

As an aircraft's attitude is supposed to be estimated at a low altitude, a small electric–powered helicopter (Eagle) is used as the flying platform and because it flies close to the ground, it can operate in a cluttered environment. As it can take off with a limited amount of payload and electrical power supply, some design consideration are given to the size, weight and power consumption of each of its part.

3.1 Panoramic UV Camera

The panoramic imaging system, consists of a CCD camera with a CS lens, UltraViolet (UV) filter and a panoramic mirror–lens (Figure 3.1), captures the image to be processed

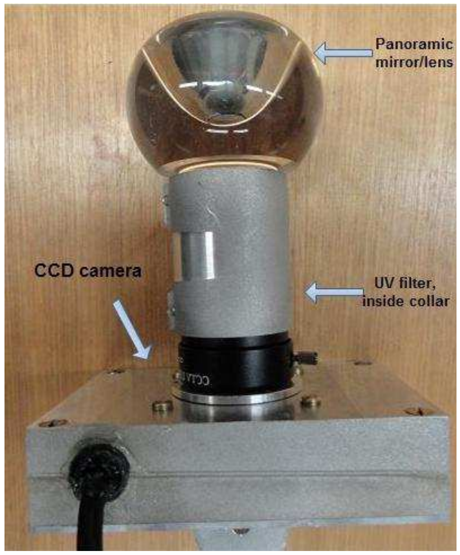


Figure 3.1: Panoramic imaging system

by the PC104 computer system. The CCD camera board is housed in a custom–designed aluminium housing which protects its electronics and provides a structure for its mounting on the helicopter.

3.1.1 Panoramic Mirror–Lens

Although one popular method for capturing a wide view of the surrounding area is to installing different cameras in different directions, this has several drawbacks. Firstly, it is cumbersome, secondly, it needs considerable processing time to reconstruct frames and finally, there is a chance of there being blind zones. Another method is to use an ultra–wide–angle lens but this provides only a 180° field of view. Moreover, these lenses are usually expensive and highly distort the image due to their refractive optics.

The panoramic mirror–lens used in this research was designed by Wolfgang Stuerzl [60]. It is made of a solid piece of Perspex lathed by a CNC machine and has 130° elevational and 360° azimuthal coverage. Figure 3.2a shows its schematic design [60] with its reflective curve and Figure 3.2b the actual manufactured panoramic lens.

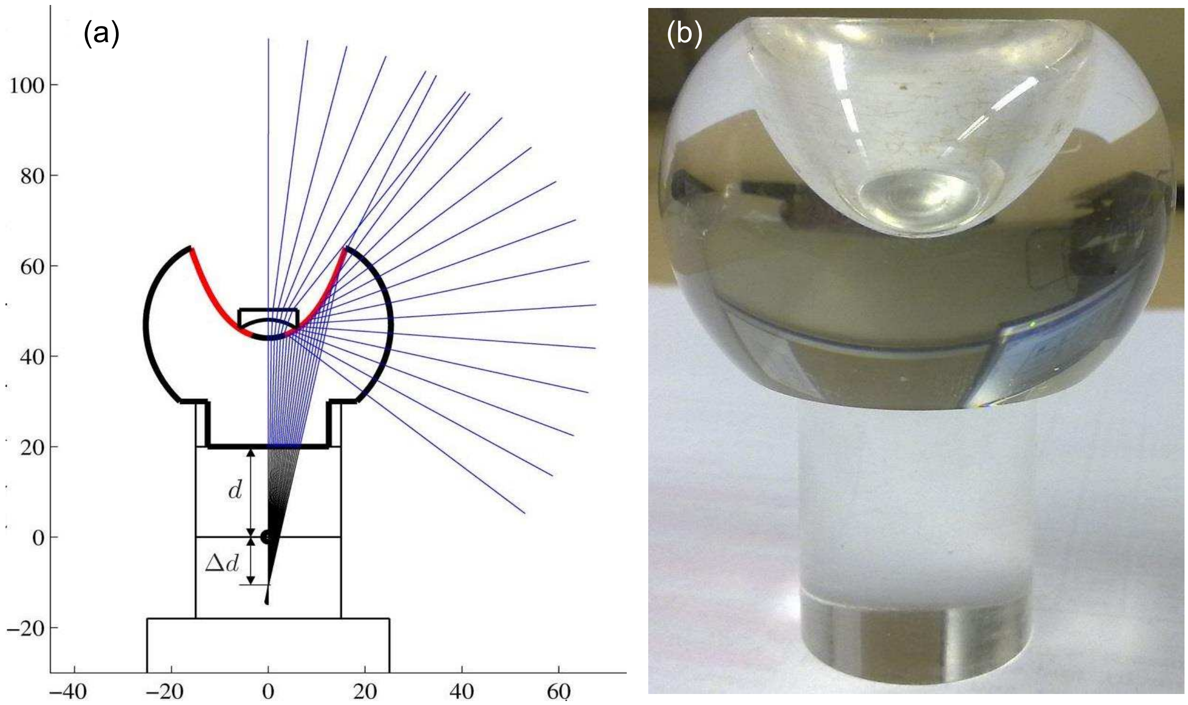


Figure 3.2: (a) Shape of air–Perspex interface, (b) Perspex lens

Figure 3.3 depicts a CCD camera placed under a reflective surface. The relationship between the elevation of an incoming ray ϕ and radial angle with respect to the optical axis is defined as α and formulated [116], as:

$$\alpha = \frac{\delta\phi}{\delta\theta} \tag{3.1}$$

With a constant α over the surface, a proportional relationship exists between the elevation angle in the environment (projected onto the surface) and the radius to the centre of the image. The lens designed by Sturzl [60] has an angular gain (due to its reflective surface and lens refraction factor) of between 6.8 and 8.2 for view points close its optical axis and a constant at 8.6 for its centeral and outer parts. This means there is a linear relationship between the incoming rays and rays projected onto the camera (linear elevation) which makes unwrapping it easy. One important property for

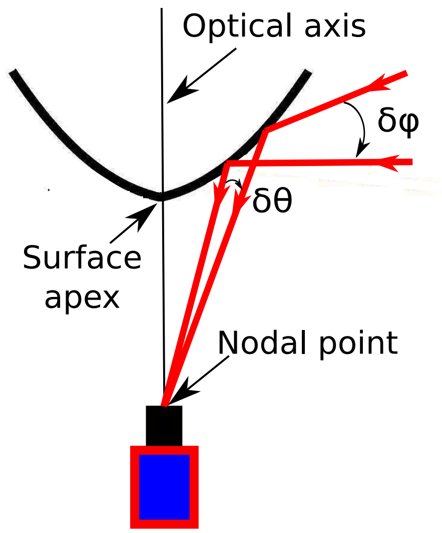


Figure 3.3: Angular gain definition of reflective surface.

satisfying a constant angular gain is γ which is mathematically expressed in Equation 3.2 and shown in Figure 3.4 [60].

$$\tan \gamma = \frac{r d\eta}{dr} \tag{3.2}$$

The distance of a point to the camera's nodal point is denoted by r and the angle of the reflected ray to the optical axis η. To formulate the proportional changes in the radian angle, the changes with respect to γ and η can be written as:

$$\frac{d}{d\eta}\left[\tan^{-1}\left(r\frac{d\theta}{dr}\right)\right] = k \tag{3.3}$$

With a constant value of the surface parameter (k), the relationship between the radial angle and angle of incidence is proportional. The angular gain α and k are related as:

$$k = \frac{2}{1+\alpha} \tag{3.4}$$

The parameters of the reflective surface of the lens are derived from [116]. The distance to the nodal point varies with respective to η as shown in Equation 3.5:

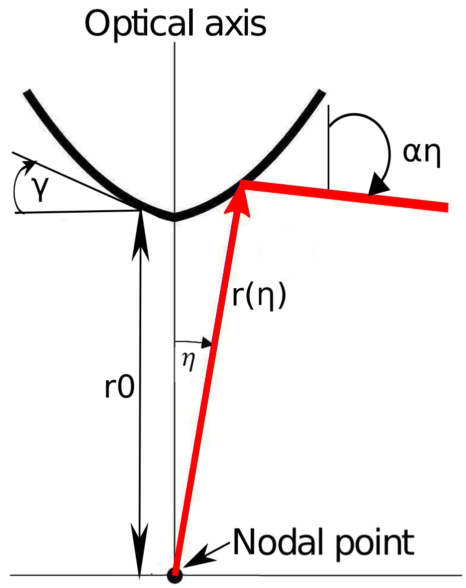


Figure 3.4: Reflective surface with constant angular gain.

$$r(\eta) = r_0 \frac{\cos^k \gamma}{\cos\left(\eta/k + \gamma\right)^k} \tag{3.5}$$

With a constant value of α=13, k is also constant which ensures that there is a proportional angular gain. In Equation 3.5, η is the angle between the incoming rays to the reflective surface and the optical axis. The parametric values for Figure 3.4 are defined as follows: maximum angle covered by the mirror surface is η_{max} that is η_{max}=12°, the angular gain is α with respect to the refractive factor of the Perspex (n_p=1.5), the overall gain is reduced by n_p^{-1}=2/3, α=8.6. r_0 is the distance from the apex of the reflective surface to the nodal point, i.e. r_0=65mm, and γ is the tangent angle of the mirror at the apex that is γ=0°.

The reflective surface depicted by the red line in Figure 3.2 is coated with a thin silver layer to provide good reflection while the central part is left uncoated. Due to the planar refraction, the practical viewpoint of the camera is at point Δd=10mm and d=20mm.

The existing mirror was on loan from the Australian National University but its silver layer was badly oxodised (see Figure 3.5) which made it unusable. To re–cover the surface, it was first thoroughly polished using bronze polish to remove the remaining

Figure 3.5: Oxidized reflective surface

silver residue and then coated with 99.9% pure silver using a vacuum–coating machine. Each coating took fifteen minutes to complete, each time a thin layer of silver (less than one micron thickness) resided on the surface. The process was repeated twenty times to produce a good reflective surface. Then the reflective surface was spray–painted with orange paint to prevent future oxidisation.

A typical panoramic image from the mirror–lens is shown in Figure 3.6 both before and after unwrapping into a spherical image coordinate system. The panoramic image is unwrapped using spherical coordinate system and shown in figure 3.6.

Note that the vertical angular gain, α, and the refraction index, $n_p \approx 1.5$, are not included into the unwrapping process, we expect image expansion and contraction in some parts of image. It is important to include an explicit value of α when calculating the location of each pixel in order to obtain a uniform unwrapped image. In addition to the optical properties of the panoramic lens, the CS lens of the CCD camera produces barrel distortion that makes real pixel mapping difficult. Rather than correcting the effects of all the optical distortions (from the panoramic and CS lenses) individually on the results, an all–encompassing calibration equation is used to relate the final vision–based attitude estimation to the actual attitude from the XSENS sensor which can

Figure 3.6: Unwrapped image using spherical coordinate system

largely correct the combined effect of optical non–linearities and misalignments. The following calibration equations are calculated using the MATLAB quadratic data–fitting function to transform the vision–based attitude estimation to match the IMU results as closely as possible.

$$\phi^\circ = 0.0025413\phi_v^{\circ 2} + 1.5385\phi_v^\circ + 2.2506 \tag{3.6}$$

$$\theta^\circ = 0.00352265\theta_v{}^2 + 1.55\theta_v^\circ - 0.37101 \tag{3.7}$$

The angles (roll and pitch) are presented in degrees and ϕ_v and θ_v are the attitudes from the vision system and ϕ and θ the calibrated angles. As can be seen in the equations, as the first terms have a minor effect on scaling, so the relationship between the vision and IMU results is nearly linear. Each equation has an offset term (constant value) to correct misalignment of the optics and is calculated for rotations up to $\pm 40^\circ$ which exceeds the values to be expected during low–speed flight close to the ground but are not applicable for higher angles due to the horizon being obscured and the image severely distorted.

3.1.2 UV filtering

Although the naked human eye is unable to see images in the UV spectrum, insects can as their Ocelli are sensitive to UV wavelengths to a peak of around 340nm. To mimic insect vision in such a wavelength, a combination of two UV filters is used to provide a pure UV band–pass filter with a peak of 340nm and side bands of 40nm (Figure 3.7). Filter (a) starts passing from 250nm to around 400nm with the peak at 340nm and has a small pass in the infrared region at around 710nm. Using only filter (a), some objects on the ground are visible as the ground is not totally dark. Then, to enhance the filtering performance, filter (b) is used in combination with filter (a). Filter (b) starts passing from 300nm to 650nm and has a flat response (no passing) to 1300nm, an ideal filtering which mimics an insect's Ocellus and enables us to cover almost the

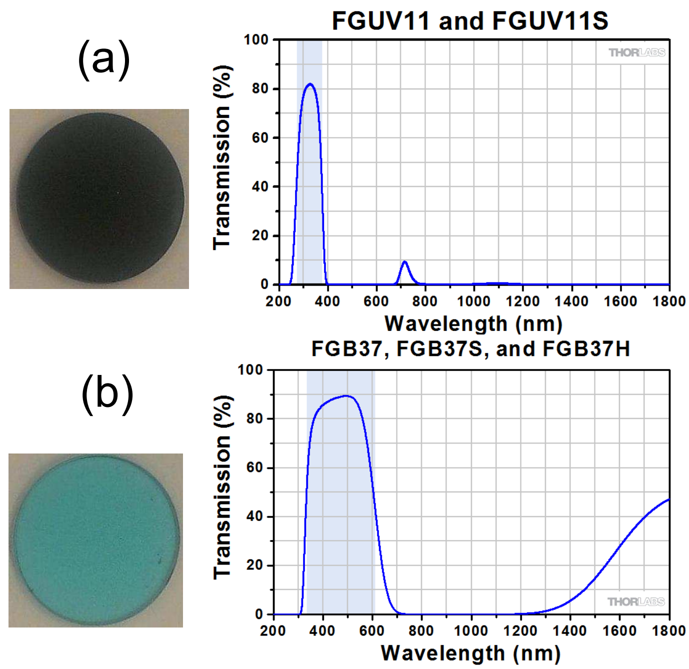


Figure 3.7: Responces of combination of two UV filters (graphs from thorlabs.com)

same UV wavelengths as sensed by insect receptors. In these wavelengths, the contrast between the sky and ground is enhanced, the earth is darker than the sky and the sun's luminosity is reduced. By improving the contrast between the sky and ground in such a way, segmentation for extracting the horizon line is simplified and does not require a highly efficient algorithm with a long processing time.

A cylindrical collar is constructed from aluminum to assemble the lenses together and house the UV filters. Its internal part is sandblasted and painted with matte paint to increase its opacity and decrease its internal light reflection. When looking at a typical panoramic image of a scene captured by this newly designed camera, most objects and surfaces on the ground are quite dark while the sky is bright but the effect of the sun is significantly reduced (Figure 3.8).

3.1.3 CCD Camera Board

The camera is a high–definition colour 600TVL video camera with a SONY CCD camera board (see Table 3.1) and CS lens mount. A CS lens with a 4–8mm adjustable focal point is used to focus the image on the panoramic reflective surface close to the camera.

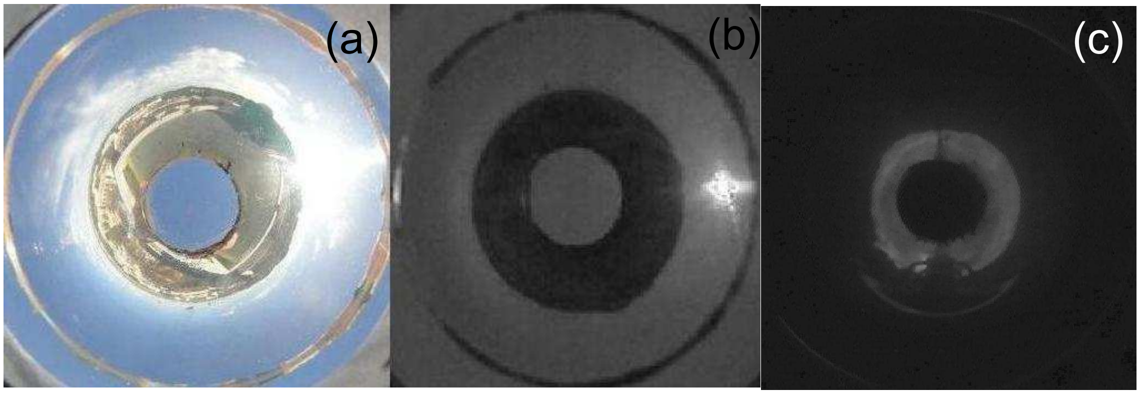


Figure 3.8: (a): normal panoramic Image, (b): UV–filtered panoramic image (upward camera), (c): UV–filtered Panoramic image (downward camera)

Usually commercial UV cameras are expensive and designed and equipped with a CCD sensor with maximum sensitivity to short wavelengths, starting from 330nm. However, their UV sensitivity depends on not only the CCD's sensor but also on its filter and can vary from one camera to another. Sensitivity to the UV spectrum (340nm) was the main criterion for choosing this camera and the one with the best quality and UV sensitivity was found through different trial and error tests. Some CCD filters are sensitive to infrared wavelengths (IR cut 700–900nm) and UV (less than 450nm). To obtain the maximum passing spectrum, the CCD filter was taken away by the company (from which it was ordered) and replaced by a UV–pass filter (to our specifications) with much improved results. The camera board has some unique specifications that make it suitable for our work, such as on–board processing including 3D Digital Noise Reduction (DNR) and a high signal to noise ratio, that enhance image quality even when capturing images on a vibrating platform such as a helicopter. Due to the payload limitation, the camera should also be as light as possible, the camera board is 100 grams and the total imaging system, including the camera, housing, CS lens, lens collar, panoramic mirror–lens and UV filters, 250gm. The camera board is show in Figure 3.9.

The camera has an analogue TV output that carries composite video signal output which is compatible with any monitor with an AV input. A composite video signal can be in a PAL (Phase Alteration Line), NTSC (National Television System Committee) or SECAM (Sequential Couleur a Memoire or Sequential Colour with Memory) system

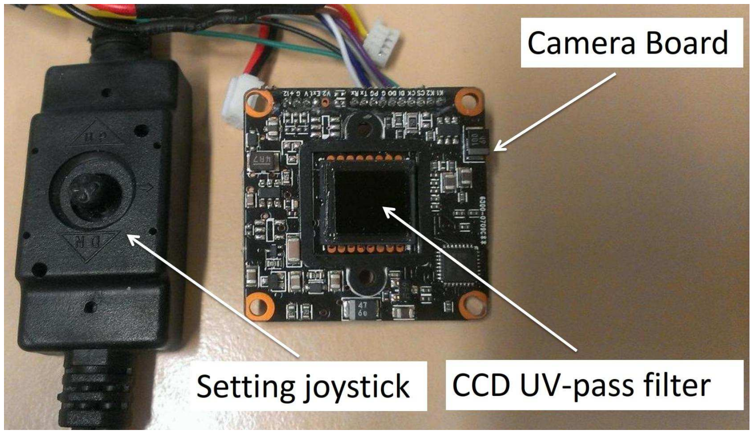


Figure 3.9: SONY CCD camera board

carried by a single cable, and it has three colour intensity components as well as vertical and horizontal synchronising signals. These three signal formats (PAL, NTSC and SECAM) are different in their encoding method, that is screen resolution, frames per second and format for transmitting the colours. The key specifications are summarised in Table 3.1, the camera's output format is set to PAL for this work.

Table 3.1: Specification of SONY camera board.

Image Sensor	1/3 interline CCD
Video Output	1.0VP-P / 75 ohm Video 0.714V p-p Sync 0.286Vp-p
Effective Pixels	NTSC 768(H) X 494(V), PAL 752(H) X 582(V)
Luminance	S/N Ration 50Db Min (AGC off)
Sensitivity	0.1 Lux (color) / 0.03 Lux (B/W)
DNR	LOW,MIDDLE,HIGH,OFF
Supply Voltage	12V DC
Power Consumption	180mA / 2.2W (Max)
Size	38mm,38mm,15mm

3.2 Eagle Helicopter Platform

A Hirobo "Eagle" 60 size radio–controlled helicopter is used as the main test platform and its general specifications presented in Table 3.2. It was originally powered by a methanol internal combustion engine but was modified to use a brushless DC motor (Actro 32–4) which this reduces the vibration a little although it is still significant. The panoramic imaging system is mounted in front of the Eagle's undercarriage under

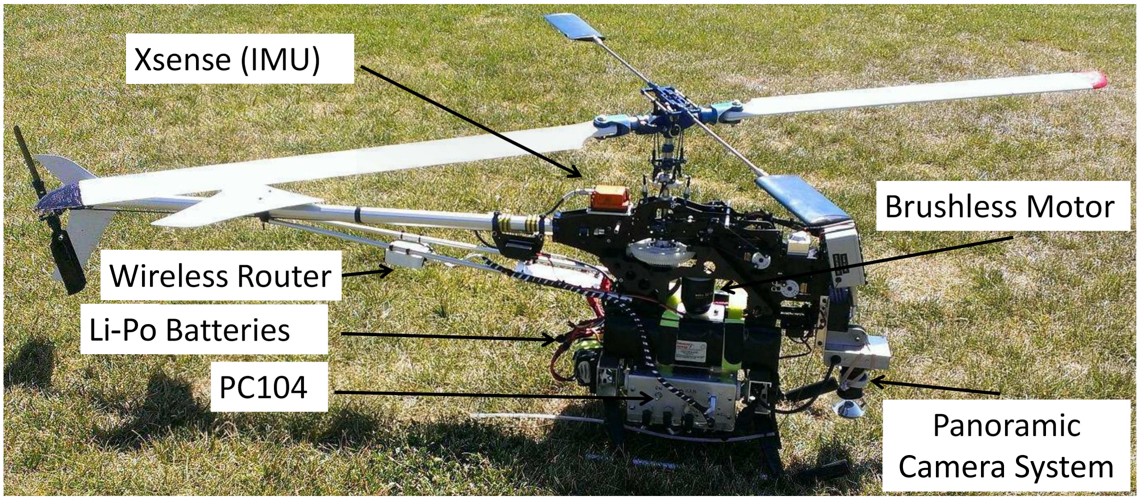


Figure 3.10: Eagle test platform

Table 3.2: Eagle platform specification

Parameter	Value
Main rotor diameter	1520mm
Tail rotor diameter	276mm
Number of blades	2
Height	470mm
Lenght	1260mm
Weight of the Eagle	4.5kg
Maximum Weight (including payload)	8.5kg
Motor type	Brushless Actro 32-4 DC motor
Rotor RPM	1600
Batteries	Two Lithium Polymer, 5 cells, 8000mAh
Available power	Around 1480 Watts (at 40 Amps, 37 V)
Flight time	15mins

its nose. The undercarriage, which contains the PC104 computer system and power systems, is attached to the Eagle's main frame by four shock absorbers to reduce the effect of body vibrations on its electronics and avionics. The helicopter's platform is equipped with the panoramic camera and PC104 computer systems, an IMU and a wireless router (Figure 3.10). The whole platform weighs 8.2kg, including two five–cell, 8000mAh and one two–cell, 5000mAh batteries to power up the Eagle and PC104 computer system respectively.

3.2.1 PC104 Computer System

A PC104–P3 with a pentium III CPU is used as the main computer system and, as it is light in weight and small in size, it is suitable for platforms with space and weight limitations, such as small aerial vehicles. An open–source Slackware Linux [117] with GNOME 1.4 environment is the operating system which is installed on 2Gbytes internal flash memory. For a Real Time Operating System (RTOS), the Linux kernel was modified and patched by Garratt [96] using RTLinux [118] version 3.1. A brief summary of the specifications of the operating system is provided in Table 3.3.

Table 3.3: Operating system specification

Name	Slackware Linux
Version	8.1
Release date	2001
Linux kernel version	2.4.4
Kernel supports	SCSI controllersUSB keyboards and mice Parallel-port IDE devices, IBM PS/2 machines
Compilers	gcc-2.95.3 (C, C++, Objective-C and Fortran-77)

The PC104's features are summarised in Table 3.4. Its programs and algorithms written in C are compiled using the GCC compiler [119], the RTOS boots up from the internal flash drive on start–up and the programs are loaded into the kernel by the user. The PC104 has a USB port that makes the transfer of data convenient for data inspection and post–processing, and a non–real–time program that is executed when the system is free. The real–time and non–real–time programs are run in different environments which is necessary because the execution of a real–time program (highly prioritised by the RTOS) cannot be blocked by that of a non–real–time program. The data from both these environments is accessible and can be shared by a shared memory structure for use in different programs. The shared data created and modified in the real–time programs is accessed by a non–real–time logger program and logged into a DAT file, which when saved, can be transferred to the PC using a USB memory stick. Communication between the real–time software and data logger program is achieved using FIFO (First–In–First–Out) based inter–process communication. The data logger is a non–real–time program that logs data from the real–time code, accesses the filesystem to record data and, due to the non-deterministic nature of the file system's input–output, must be separate from the real–time system.

Table 3.4: PC104 technical specification

CPU Speed	1.2Ghz Intel Tualatin Celeron Pentium III CPU
Main Memory	256Mbytes onboard memory
Flash Disk	2Gbytes Flash Disk
Number of serial ports	Two (RS232C, RS485/422)
Number of USB ports	2
Key Board and Mouse	Supports PS/2 KB & PS/2 Mouse
Network Interface	One 10/100 Base-T using Intel 82559ER
Embedded features	Digital I/O (4 inputs, 4 outputs)
Power supply	+5v

3.2.1.1 Video Grabber

The camera images are captured using a $FG104^{TM}$ PC/104–PLUS frame grabber. The $FG104^{TM}$'s video characteristics of which are summarised in Table 3.5. Although the video grabber has both PCI and ISA bus, the PCI bus is used to transfer signals and data to the PC/104–Plus due to its faster transfer rate. The CCD camera provides a composite video that is connected to the frame grabber's composite input and provides frames in PAL format (with an 8–bit grey level) with resolutions of 752(H) x 582(V). The frames are sub–sampled to 288(H) x 384(V) pixels by the frame grabber which is able to capture 25 full frames a second in PAL or SECAM video systems. In an interlaced video, a full frame is divided into two fields made up by composed of odd and even lines and for 25 fps, each field (odd or even) is displayed for $1/50^{th}$ of a second. The frame grabber captures odd and even lines of each frame each every 20ms and provides 50 half–frames consisting of odd and even lines a second.

Table 3.5: $FG104^{TM}$'s summarised video characteristics

Video inputs	PAL/NTSC/SECAM/S-Video
Output Resolutions	(PAL/SECAM): 768 x 576, 384 x 288 (NTSC): 640 x 480, 320 x 240, 720 x 480
Capture Rate	30 fps (NTSC), 25 fps (PAL/SECAM)

Table 3.6: The 3DM–GX1 general specification

Input voltage	+5.2 to +12.0 VDC
Typical power consumption	50 mA
Shock	500g
Data rate	angle mode up to 45 Hz; sensor mode up to 70 Hz
Baud rate	1200, 9600, 19200 bps
Communication Interfaces	RS-232
Weight	62g

3.2.2 Inertial Measurement Units

Two IMUs (a 3DM–GX1 from microstrain [120] and a XSENS MTi–300 [121]) provide a complete Attitude and Heading Reference System (AHRS). Some experiments use the 3DM–GX1 as the reference but this is replaced by the XSENS which can provide more accurate measurement on vibrating platforms. The 3DM–GX1's general specifications are given in Table 3.6.

The XSENS MTi–300, which is especially designed towork in high–vibration environment, is another IMU sensor. It is mounted on top of the Eagle's frame, as can be seen in Figure 3.10, and its data interface is via one of the PC104's RS232 serial ports. Although its internal sampling frequency is 100Hz, its data is read evry 20ms (50Hz) and it is also powered by one of the PC104's RS232 ports. The general specifications of the XSENS are in Table 3.7 and its technical specifications after testing and calibration in Table 3.8.

3.2.3 Communication

The Eagle and ground station communicate through a wireless connection using an Ethernet wireless router (TP–LINK 150 Mb/s), attached under the Eagle's tail (Figure

Table 3.7: The XSENS general specification

Input voltage	4.5-34V or 3V3
Typical power consumption	675-950 mW
Start-up time	2.5 sec
Shock	2000g
Sampling frequency	10 kHz/channel (60 kS/s)
Output frequency	Up to 2 kHz
Latency	<2 ms
Interfaces	RS232/422/UART/USB (no converters)
Built-in self test (BIT)	gyroscopes, accelerometers, magnetometer
Weight	55g

Table 3.8: The XSENS technical specification

	Accelerometer		Rate Gyro		Magnetometer	
Full Scale	50	[m/s2]	450	[$\circ$/s]	3.88	[a.u.]
Bandwidth[Hz]	375	[Hz]	415	[Hz]	n/a	

3.10) and connected to the PC104 LAN port while the ground station consists of a laptop with an onboard wireless capability. In Practice, the wireless connection is achievable up to 100 meters which is sufficient for proving the concept and validating the work. A free version of the MobaXterm home edition software [122] is used as an Xserver with an X11 server and terminal client to provide remote access to the PC104 through a wireless connection and enables the variables to be conveyed to the user during flight. Some graphical user interface (GUI) widgets are created using the EZwgl library [123] to allow the variables to be seen on the monitor using the direction PC104's VGA wireless connection. A widget created to monitor vision–based attitude data is shown in Figure 3.11.

3.2.4 Summary

In this chapter, the development of different parts of the experimental platform has been described. The camera with the combination of UV filters provided a good UV–filtered image despite the fact that it was not intended to operate in UV wavelengths. By careful consideration, the overall weight of the Eagle including the camera, PC104 and batteries, did not exceed the maximum allowable payload and the final weight was

Figure 3.11: The GUI widget for vision–based attitude data

8.2kg. The wireless connection between the ground station and Eagle was operable up to 100 meters which is enough for accessing the PC104 system and remotely monitoring different tasks.

Chapter 4

Vision–Based Attitude Estimation

4.1 Introduction

This chapter describes the main part of this research in witch a panoramic image is processed by different algorithms to provide an attitude estimation from the identified horizon. After capturing a panoramic image with enhanced contrast between the ground and sky, the aim is to extract the horizon where the ground is coincident with the sky and compute the attitude of the observer. Each algorithm is validated using MATLAB simulations with real and artificial panoramic images, including a continuous sequence of video frames recorded during flight tests on the Eagle helicopter. To test and examine the real–time performance of this technique before mounting it on the Eagle, the camera is fixed on a tripod (Figure 4.1), either facing towards the sky or hanging with the mirror pointing to the ground. Section 4.4 describes how the attitude is calculated from the 3D mapping of the horizon.

The chapter is organised as follows. In section 4.2, an explanation of how an ideal horizon can be determined is provided. Section 4.3 describes different parts of the image processing for horizon extraction, includtion of the optimum sky/ground threshold, edge detection, horizon extraction, Sun–Tracking thresholding and recovering the missing part of the horizon due to the sun luminosity. Section 4.4 describes the methods used to estimate the attitude from 3D mapping of the horizon. Simulations of the

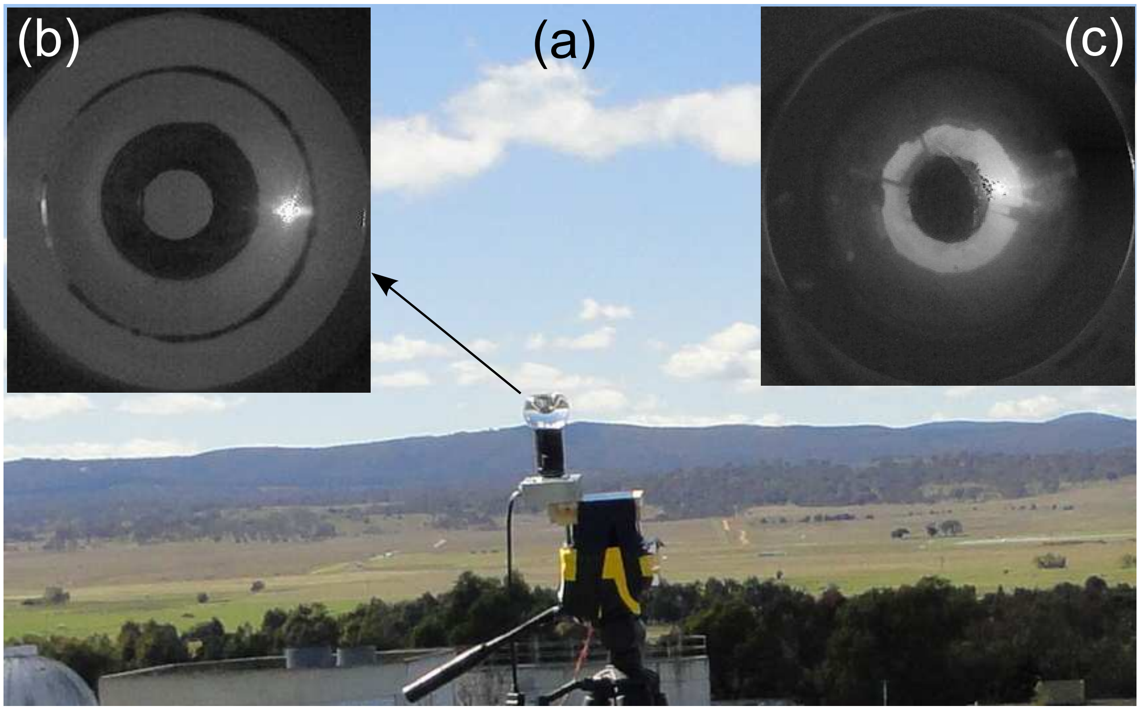


Figure 4.1: (a): Tripod–based platform, (b): panoramic projection from pointed down camera, (c): panoramic projection from pointed up camera.

panoramic horizon and attitude estimations were conducteds in MATLAB SIMULINK are discussed in section 4.5, the real–time implementation of the algorithms in the PC104 system explained in section 4.6 and the vision–based attitude estimation tested and verified during a real flight test using the Eagle helicopter described in section 4.7. Section 4.8 explains the importance of gravity direction for attitude determination and the advantage of using a vision rather than inertial system to define the gravity direction when both are mounted on a vibrating platform. Finally, a summary of the chapter is provided in section 4.9.

4.2 Simulation of Horizon Projection

The horizon is the separation observed between sky and ground. For a 2D panoramic image, an ideal one is represented by a circle located in its centre (in the case of no roll and pitch rotations from a level condition). However, if there is any roll and pitch rotation, this circle is deformed to another shape such as an ellipse, parabola or a straight line. Typical images taken by the panoramic imaging system (Figure 3.1) are shown in Figure 3.8. To visualise the projection and deformation of the horizon, it is informative to generate artificial panoramic images using MATLAB programming. In this section, equations are derived to provide image (pixel) coordinates for the locus of points representing the ideal horizon when viewed from a panoramic imaging system at an arbitrary roll and pitch angle. To generate a horizon projection, the earth is assumed to be a perfect ball with 6378km radius and the horizon placed at sea level. Figure 4.2 shows the geometrical model of the horizon plane with relation to an object above the earth.

Firstly, the distance to the horizon is calculated by the Pythagorean theorem. The lines connecting points A to B and A to C are tangent lines to the circle circumference and perpendicular to its radius which form right triangles, with the sum of the radius and height as the hypotenuse. The distance to the horizon is given by Equation 4.1 and the half included angle between the normal and the line to the horizon, θ is given by Equation 4.2. The global distance between the object and the centre of the horizon circle is Z_G and is given by Equation 4.3.

$$d = \sqrt{h(2R + h)} \tag{4.1}$$

$$\theta = \sin^{-1}(\frac{R}{R + h}) \tag{4.2}$$

$$Z_G = d\cos(\theta) \tag{4.3}$$

where d is the distance to the horizon, R is the radius of the Earth and h is the height of the observer above sea level.

The global X, Y and Z coordinates of equispaced points on the horizon are then generated for the horizon circle relative to the object, which is defined at the origin. For the purposes of illustrating the horizon in Figure 4.4, we use points spaced at

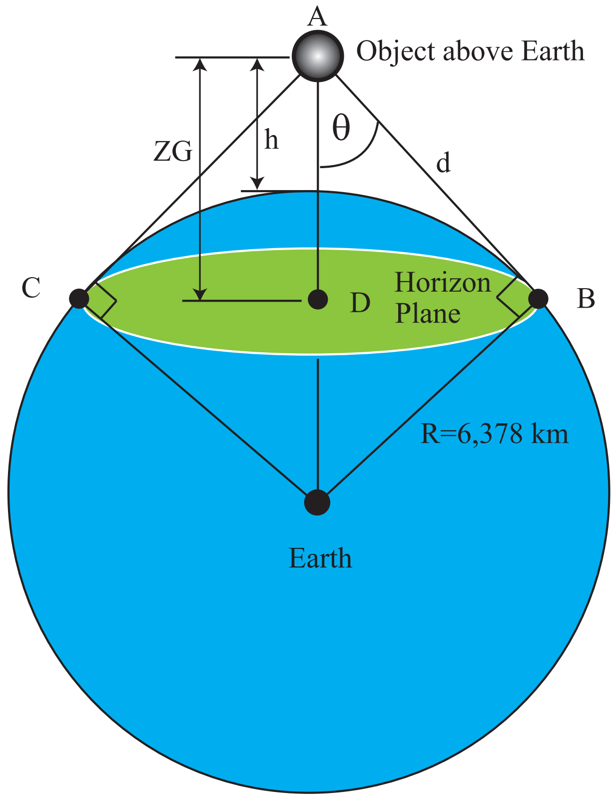


Figure 4.2: Geometrical distance to horizon

$0.1°$ in azimuth around the circumference. Global coordinates are defined parallel to the earth's surface, where the line from the object to the centre of the earth crosses the surface, using the North–East–Down (NED) reference frame. The horizon points are calculated in Cartesian coordinates $[X_G, Y_G, Z_G]^T$ using d in Equation 4.4. The Z coordinate of each point is simply Z_G and for any given height above the earth (h) the Z_G is a constant value.

$$\begin{aligned} X_G &= d\cos\varphi \\ Y_G &= d\sin\varphi \\ 0° &> \varphi > 360° \end{aligned} \tag{4.4}$$

Then, the effect of attitude transformation on the panoramic projection of horizon is simulated using Equation 2.6. In the developed MATLAB program, each point generated using Equation 4.4 is converted to coordinates in the object's local reference frame using the corresponding rotation matrix (Equation 2.6) to obtain points in local coordinates $[X_L, Y_L, Z_L]^T = [B][X_G, Y_G, Z_G]^T$. The new locus of the horizon points is then converted to spherical coordinates to determine the local azimuth (α) and elevation (β) of each point (Equation (4.5)), before their final conversion to a 2D panoramic image projection.

$$\begin{aligned} \alpha &= \cos^{-1}(\frac{Z_L}{\sqrt{(X_L^2 + Y_L^2 + Z_L^2)}} \\ \beta &= \tan^{-1} 2(Y_L, X_L) \end{aligned} \tag{4.5}$$

To obtain the final transformation from the spherical to image coordinates, a property of the panoramic imaging system, which states that the radial distance from the centre of the 2D image is linearly proportional to the elevation of the feature observed, with the azimuth angle of this feature preserved through the transformation, is required. Therefore, the image coordinates [U,V] can be calculated from:

$$\begin{aligned} \mathrm{U} &= k\beta \cos(\alpha) \\ \mathrm{V} &= k\beta \sin(\alpha) \end{aligned} \tag{4.6}$$

where k is the gain of the camera system relating the image radius to the elevation.

Different rotations on a circle make different projection shapes. For example, when the plane is level with 0° of pitch and roll, the 2D projection is a complete circle at the centre of the image as can be seen in Figure 4.3. In Figure 4.3, the 3D horizon is rotated for 0°, 30°, 60° and 90° and converted back to 2D as marked. In the case of having rotations in one axis (roll or pitch), the projection is not a complete circle and the shape of horizon is elliptical. For a combined pitch and roll rotation, the horizon resembles elliptic or parabolic shapes.

This horizon simulation shows an ideal panoramic image. In a real image taken by a panoramic camera, some other factors such as optical refraction and deformation

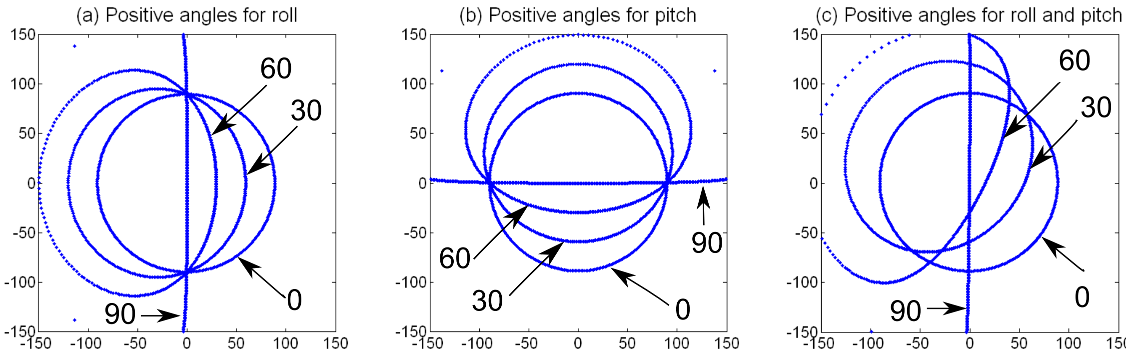


Figure 4.3: Simulation of panoramic horizon with different rotations, (a): Pure roll rotations ($0° - 90°$), (b): Pure pitch rotations ($0° - 90°$), (c): Combination of roll and pitch rotations ($0° - 90°$)

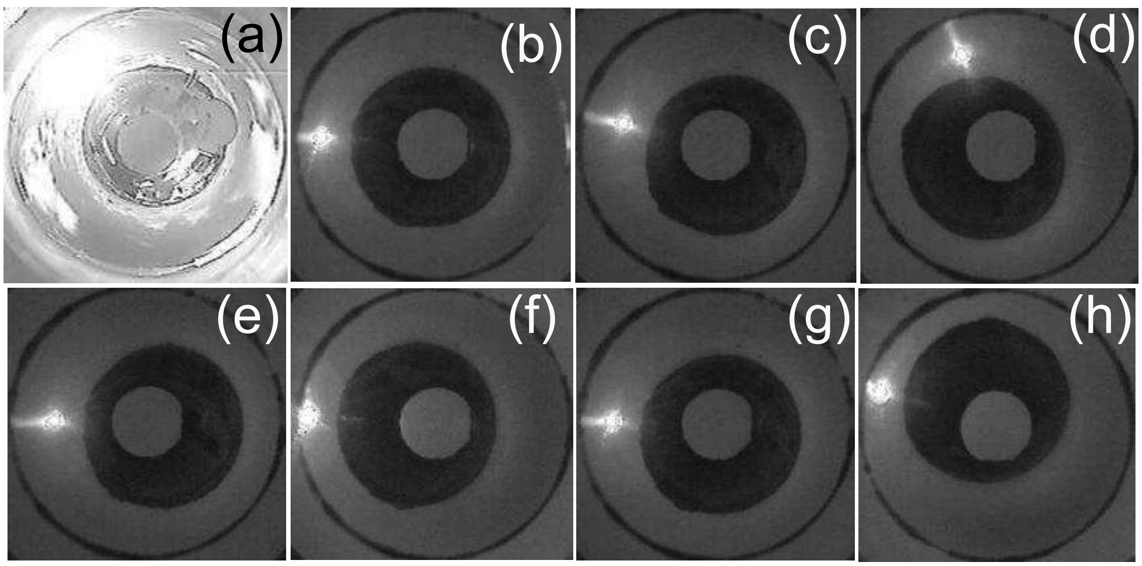


Figure 4.4: Real panoramic image with different rotations (a): grey image, roll=3°, pitch=-8°, (b) roll=-1°, pitch=2°, (c) roll=14°, pitch=8°, (d) roll=-8°, pitch=12°, (e) roll=20°, pitch=3°, (f) roll=-21°, pitch=-2°, (g) roll=2°, pitch=12°, (h) roll=-5°, pitch=-25°

change the image. Different images with different rotations are shown in Figure 4.4. The camera is pointed upward (on a tripod) and roll and pitch angles are measured by an IMU (M3DM).

There are some factors that deform the shape of the horizon when it is projected on a 2D plane. As can be seen in the Figures 4.3 and 4.4, the horizon is a projective 2D circle but when the rotations increase, it becomes deformed into an ellipse or a parabola. Also, optical components affect the 2D image projection, such as Barrel distortion which comes from different radial magnifications of optical components (such as the lens). Deformation can change a 2D panoramic projection to a completely different shape when high degrees of rotations exist. For small rotations (while the horizon projection is still a circle), the center of the horizon circle can be considered as a good representation of rotations. However, finding the center becomes difficult when the shape of horizon is parabolic and is impossible when it is a line (90° in roll or pitch in the Figure 4.3). When a line in a panoramic image represents the horizon, it is difficult to find in which axes it has rotations due to the lack of visibility in 2D. When using the center circle for rotation estimation, these deformations limit the estimation up to where the shape has a geometrical center. In addition, in real conditions, there is a probability of finding some true horizon points which do not belong to the horizon (due to the edges, around of occluding objects). These unwanted points affect finding the true horizon center.

4.3 Image Processing

This section describes how a panoramic image is processed for the horizon extraction.

4.3.1 Sky/Ground Thresholding

Several methods that exist for image segmentation varyin terms of their applicability, efficiency and computational complexity. The challenge of image segmentation is highlighted when real–time image processing is required. A simple method of calculating the threshold value of an image is Otsu [44] which is also called global image thresholding. Otsu is a non–complex method, easy to perform, but inefficient when different parts of the image have different brightness, especially when the sun luminosity affects a part of the image. There are different ways to calculate a proper threshold for a particular image but may not be applicable for images taken during cloudy or sunny days. Another major problem is the sun's luminosity that increases the brightness of surroundings regions and makes horizon extraction difficult. For an image such as a panoramic one, a method is needed that will be able to perform well in most conditions and handle the effect of the sun. It is also difficult to apply a general method of thresholding on a panoramic image because although it has enhanced contract, but the variation in pixel intensity are not sufficiently broad to find the best threshold value at which to separate the sky from the ground pixels.

As a result, a new method for finding optimum threshold values for the sky and the ground is proposed. The method is developed particularly for the UV–filtered panoramic image in this project but is applicable to panoramic images with the same properties. It is based on sampling the sky and the ground pixels by masking them. With the first hypothesis that the central region of the image is occupied by the sky (when no rotations and with downward camera), it is possible to estimate where the sky pixels are, and the average value of pixels on the region can be found by using a circular mask to mask the designated region (Figure 4.5a). The same method is also used for the ground. The location of the regional masks (sky and ground masks) are moving when platform rotations are known. It is assumed that the first image is taken when there are no rotations or the initial conditions are known. When the camera is mounted on the Eagle, the pixels which are in blind spots (e.g. undercarriage or legs) are also taken out of the calculations. The sky still resides in the centre of the image even if the platform rotates up to +/-35° in roll and pitch. In real flight, we expect the Eagle not to exceed more than +/-20° in roll and pitch. In the tests, the masking method has shown its ability to find an acceptable threshold even without moving the position of masks. As mentioned earlier, for testing on the ground (when the camera in mounted on the tripod), the camera points upward so that the central circle is the ground as can be seen in Figures 4.4. The threshold finding method demonstrated outstanding performance in different weather conditions such as cloudy, sunny and foggy days, as well as smoky or hazy horizon.

4.3.2 Edge Detection

After classifying the image pixels, edge detection is performed on each image row. It scans the changes pixel values horizontally and compares each value with those of the

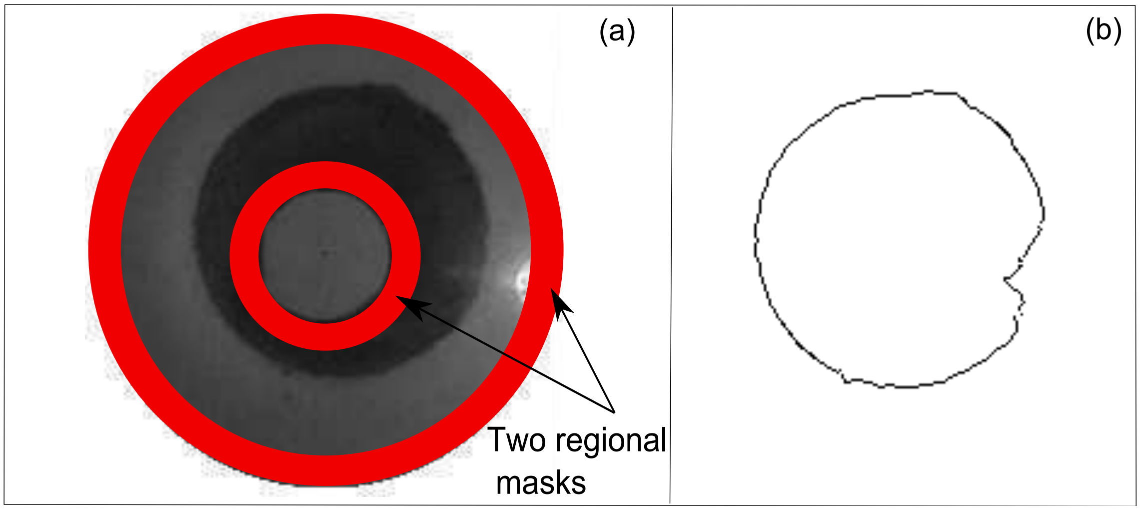


Figure 4.5: (a): Real panoramic image with marked regional masks,(b): Edge detection after thresholding

next pixels to find an intensity variation. The sensitivity of the edge detection depends on the number of pixels compared together to determine the intensity changes. When an intensity change is detected, the first pixel's value becomes zero and any following one 255. There are some situations in which a small part of the horizon line lays on a row which means that it is not completely extractable by the edge detection as it is considered to have invariant pixel values. However, the effect of an undetected horizon part is negligible compared to the detected horizon line. In the process of selecting true horizon points, unwanted ones are eliminated. There are two regions, the inner and outer circles, in which the points after the edge detection do not have useful visual information (4.6b and 4.6c). The points of these regions are eliminated for the edge–detected image. The image before and after applying threshold and the edge detection is shown 4.5. There are also some detected edges due to the Eagle's undercarriage and legs which are also blanked out.

4.3.3 Horizon Extraction

After extracting the image edges, as not all belong to the true horizon, it is important to determine the true edges which are referring to the horizon where the sky and the ground are separated. Extracting the true horizon is the most important part of vision–based attitude estimation. False horizon detection can cause a flight control system to erroneously control an airplane, causing an unwanted dive, climb or roll. Horizon detection is the most challenging part of horizon–based attitude estimation. Horizon detection is a tedious task due to factors such as the variety of texture, effect of the sun and clouds, and when the horizon is not clear because of a smoky or hazy horizon. The panoramic imaging system developed in our work (Figure 3.1) provides UV–filtered images with enhanced sky/ground contrast and also mitigates the effect of the sun. Having this enhanced contrast in a UV image makes it easy to perform sky/ground segmentation using a non–complex method. A new algorithm for horizon extraction is proposed through the following three steps:

1. Calculation of the optimum sky/ground threshold (4.3.1),
2. Edge detection (section 4.3.2), and
3. Recovering of horizon line using Sun–Tracking thresholding (4.3.4).

4.3.4 Sun–Tracking Threshold

Although most horizon points are extracted using the proposed thresholding method, some might be lost after edge detection due to the sun's flare and luminosity, when the sun is close to the horizon or the effect of flare on the image is worsened by optical components. A new method called Sun–Tracking thresholding is proposed in this research which can perform more effective image segmentation in the regions around the sun. It is performed using a moving mask which follows the location of the sun on an image and performs local thresholding. It obtains better sky/ground segmentation than previous methods, by thresholding in the region with high brightness caused by the sun or where the sun's reflection is dominant. This method of Sun–Tracking thresholding uses the following procedure:

1. Find and save the coordinate of pixels with intensities higher than the sun threshold value,

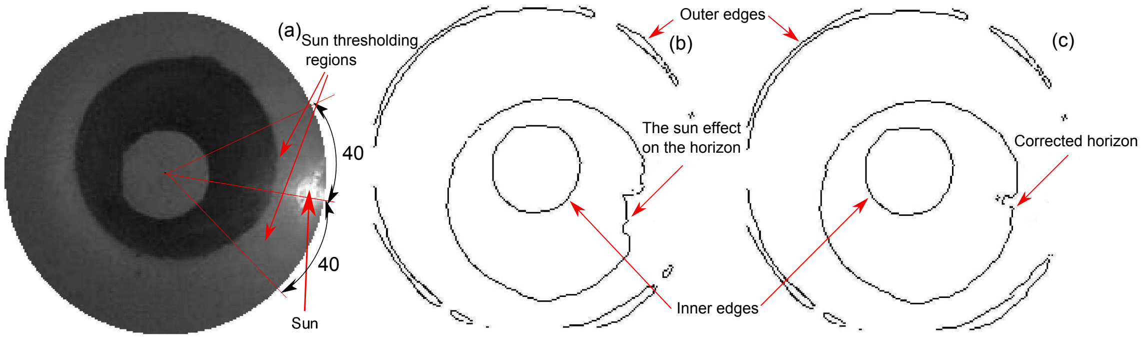


Figure 4.6: (a): Real image,(b): The effect of the sun on horizon line extraction, (c): Corrected horizon part

2. Calculate an approximate center of the sun by finding the average value of the selected high-intensity pixels in x–y coordinate system,

3. Estimate the azimuth location of the sun from the center of high value pixels,

4. Define a starting and the ending azimuth with a sun mask covering -/+40° in azimuth for regional thresholding,

5. Calculate the average intensity of pixels on the defined region (The average intensity value becomes the threshold for step 6 (Otsu threshold [44])),

6. Apply the threshold to transform the pixels in the mask region to binary,

7. Perform edge detection on the pixels in the local region of the sun mask, and

8. Overwrite the regional edge-detection image on the original edge-detected image.

Correcting the horizon using the Sun–Tracking thresholding is shown in Figure 4.6c, in which it can be seen most of the missing part of the horizon is recovered. The algorithm for the Sun–Tracking thresholding takes 6ms of computational time in real–time.

4.4 Horizon–Based Attitude Estimation

Due to its geometry, a panoramic image represents a projection of a spatial–based image of the surrounding environment. Each pixel can be addressed by a corresponding

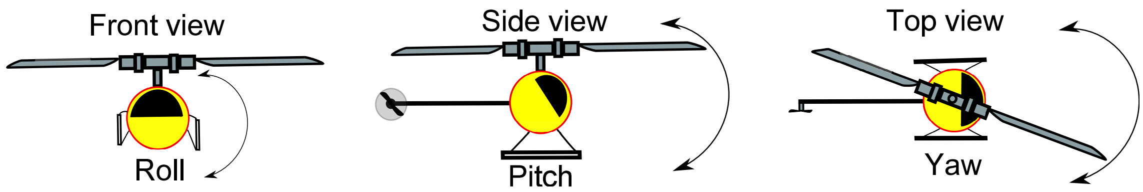


Figure 4.7: Aircraft rotations

spherical coordinate and, apart from its pixel value, it has a unique radius to the centre, an azimuth and an elevation. To estimate the attitude from the horizon, a panoramic image is reconstructed in 3D. Due to the outstanding performance of horizon detection, this 3D mapping of horizon mainly contains the horizon points. As the position and orientation of the horizon has information about the attitude, pilots are trained to rely mainly on visual information during flight, especially of the horizon. In an aircraft, the x–axis points to the nose and rotations around it are called roll. The y–axis points to the right wing and rotations around it are called pitch. The z–axis points downward and rotations around it are called yaw. The direction of positive and negative rotations depends on the standards of flight. Usually rolling to the right is positive, pitching up (nose up) is positive and yawing clockwise is positive. Rotations of an aircraft are shown in Figure 4.7

The plane consisting of the horizon points (Figure 4.2) is a good reference for attitude determination. One way of determining attitude change is by observing the locational changes of point D. The point D is the projection of z–axis of the object which is perpendicular to the x–y plane of the object. The point D is at the centre of the horizon plane when there is no rotation. Any deviation from the centre of the circle means a degree of attitude changes. Attitude can also be determined by calculating the inclination of the horizon plane. Spatial coordinates of the horizon points from the panoramic image can be used to reconstruct 3D horizons to calculate the parameters of the 3D horizon plane. The resolution of the image is 200 by 200 and the centre of the image is at the pixel located at row=100 and column=100. Each pixel has a radius to the centre and an azimuth and an elevation that can be calculated from the Cartesian coordinate of pixels using Equations 4.7. The Cartesian coordinate of edges on the image after the edge detection is stored in a 2D array. The Cartesian coordinate of each edge is converted to the Spherical coordinate using Equations 4.8. Due to the optical properties and the geometry of the lens and mirror, (e.g. lens refraction, mirror shape and camera lens barrel distortion), the elevation varies from one location to another. In the panoramic image, the radius of each pixel to the centre of image is the elevation of the pixel in Spherical coordinates. Throughout experiments, it will be shown that for the expected platform rotations, the effect of the optical distortion on the elevation is insignificant and any misalignment can be corrected by two calibration equations (for roll and pitch angles).

$$\begin{aligned}
&\text{Pixel X-Coordinate, U} = \text{Column number of the edge} - \text{Central column},\\
&\text{Pixel Y-Coordinate, V} = \text{Row number of the edge} + \text{Central row},\\
&(\text{Central column} = 100, \text{Central row} = 100).
\end{aligned} \tag{4.7}$$

$$\begin{aligned}
r &= \sqrt{(\mathrm{U}^2 + \mathrm{V}^2)}\\
\theta &= r\\
\varphi &= \tan^{-1} 2(\mathrm{V}, \mathrm{U}).
\end{aligned} \tag{4.8}$$

Note that normally to convert spherical coordinates to 3D Cartesian coordinates, the distance to each horizon point is needed. However, as we are only interested in the orientation of the horizon plane this information is not required. Instead, we make the assumption that the distance to all points on the horizon are the same, since the horizon is equidistant from the observer in all directions except if the terrain is undulating (e.g. mountains). An arbitrary scale of unity can be used to convert the points into Cartesian coordinates and then 3D mapping of each pixel can be performed by:

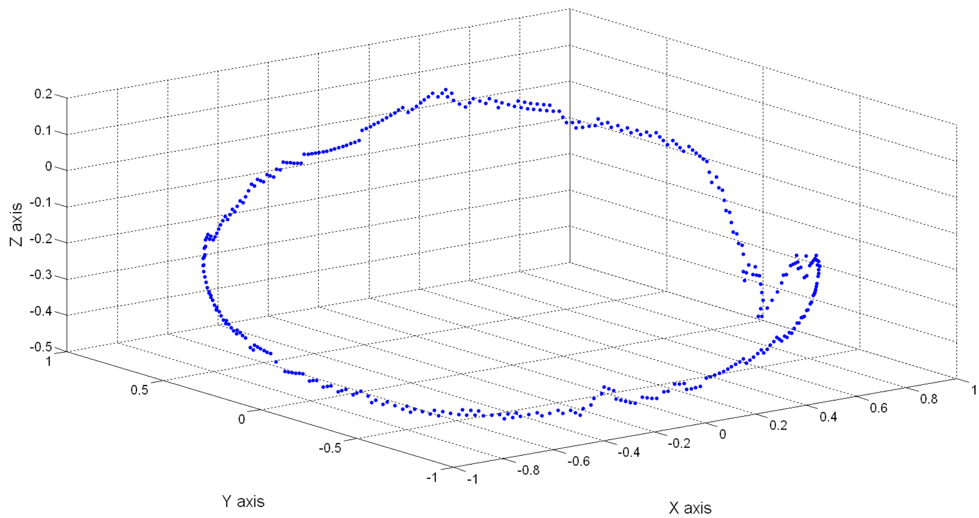


Figure 4.8: 3D mapping of horizon

$$
\begin{aligned}
X &= \sin\theta \times \cos\varphi, \\
Y &= \sin\theta \times \sin\varphi, \\
Z &= \theta \times -1.
\end{aligned}
\tag{4.9}
$$

The elevation is multiplied by -1 to map the Z–axis in the right direction (toward the earth) as in the standard NED coordinates used in aircraft. The 3D mapping of the horizon (Figure 4.6b) is shown in Figure 4.8.

Once the horizon points are mapped in 3D, the next step is to estimate the horizon plane as shown in the Figure 4.2. Least Squares Estimation (LSE) is used for plane data fitting. A 3D plane is considered as the model whose parameters are determined by the horizon points throughout LSE. The 3D plane Equation is:

$$Ax + By + Cz = 1 \tag{4.10}$$

The plane coefficients, A, B, and C are determined by LSE regression where the sum of the squares of the error residuals ϵ in the Equation 4.11 is minimised. In Equation 4.11, x_i, y_i and z_i are the coordinate of i^{th} point.

$$\epsilon = \sum_{i=0}^{n} (1 - Ax_i - By_i - Cz_i)^2 \tag{4.11}$$

the error ,ϵ, is calculated by the matrix form of LSE equation as,

$$\epsilon = (A^T A)^{-1} A^T b \tag{4.12}$$

where A is matrix form of points:

$$A = \begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ . & . & . \\ . & . & . \\ x_n & y_n & z_n \end{pmatrix} \tag{4.13}$$

and b is,

$$b = \begin{pmatrix} 1 \\ 1 \\ . \\ . \\ 1_n \end{pmatrix} \tag{4.14}$$

The LSE estimated plane is shown in Figure 4.9c. The true horizon points are selected iteratively by calculating the LSE of the 3D plane using inliers and taking out the outliers based on the distance to the pre–estimated plane. In this work, the model fitting using inliers is akin to Least Trimmed Squares (LTS) [124] but differs from it as the points are not selected randomly and the initial plane estimation is calculated using all edges. This pre–estimation is based on the fact that the horizon edges are mostly scattered around the horizon plane (despite possible false edges) due to the efficient horizon extraction. The 3D plane parameters are calculated by LSE in each iteration using inliers that were within the model threshold selected in the previous iteration.

Although the edge–detected image might have some outliers (unwanted edges) which are not the true horizon points, the edges are mostly scattered around the horizon (verified by a MATLAB simulation and real test). Due to iterative data refining, as the true horizon points (inliers) are selected in each iteration, less computation time is required and greater estimation accuracy is achieved in the next estimation. In the process of data refining, the outliers which are the result of different visual features such as the sun flare, clouds and occluding objects such as buildings and trees when flying at low altitude. The final plane estimation is made of inliers (red points) while outliers are not included (blue points). It can be seen that true horizon points are selected for the plane fitting after applying the proposed trimming method. Figure 4.6 was chosen to examine the effectiveness of the proposed method on true horizon selection to eliminate unwanted data when flying at low altitude, where surrounding objects can cause unwanted edges on the horizon. The LSE approximates the plane coefficients in Equation 4.11 that are the components of the plane normal vector. The inclination (roll and pitch rotations) of the 3D plane can be calculated from its normal vector $|r|$ (see Equation 4.15).

$$\begin{aligned} |r| &= \sqrt{A^2 + B^2 + C^2} \\ \phi &= \tan^{-1}\left(\tfrac{B}{C}\right) \\ \theta &= \sin^{-1}\left(\tfrac{A}{|r|}\right) \end{aligned} \tag{4.15}$$

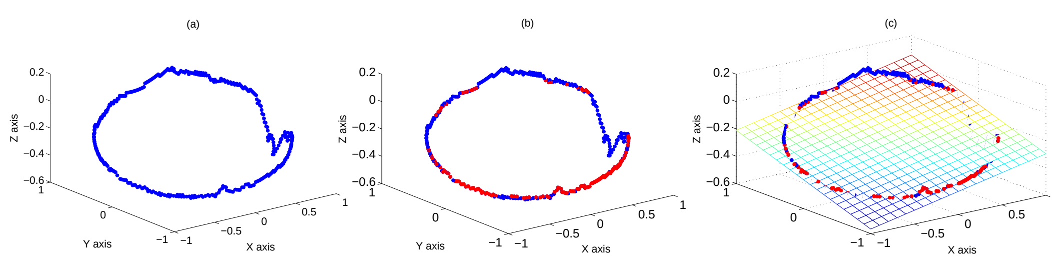


Figure 4.9: Horizon plane estimation, (a): 3D mapping of the horizon, (b): Refined true horizon points (true=red, blue=false), (c): 3D plane fitting using true horizon points.

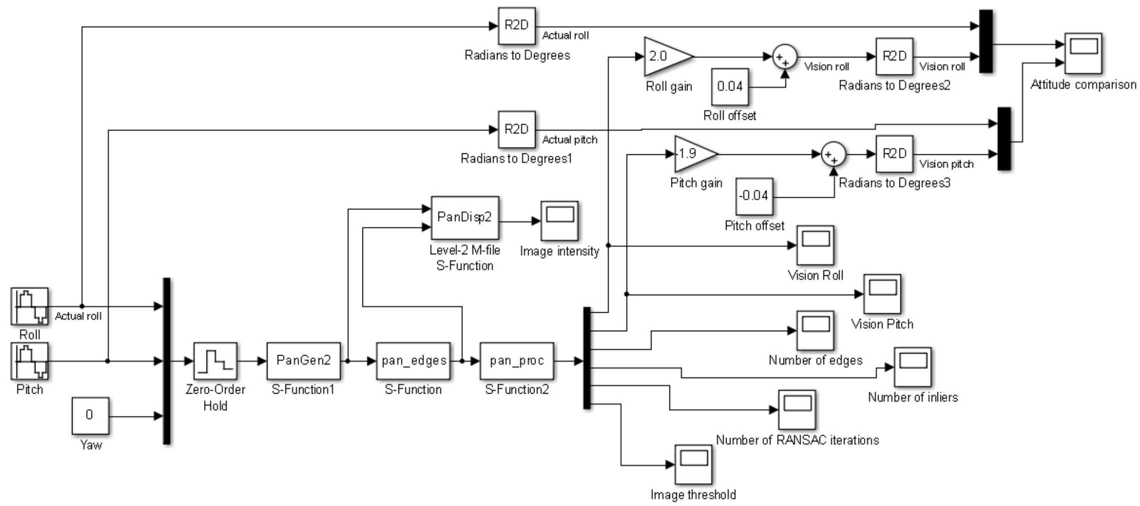


Figure 4.10: MATLAB SIMULINK model of panoramic attitude estimation

4.5 Panoramic Attitude Estimation using MATLAB SIMULINK

A MATLAB SIMULINK model was created to visualise and simulate continuous attitude estimation using synthetic panoramic images. It is very important to be sure about the performance of algorithms before real–time implementation. The SIMULINK model is shown in Figure 4.10. It is assumed that the panoramic projection is from a downward camera that produces an image like that in Figure 4.15b, which consists of a circle in the centre (sky) with white color surrounded by black colour (ground).

The algorithm's main functional boxes are as follows:

1. PanGen2: for creating panoramic images,

2. Pan_edges: for image thresholding and edge detection,

3. Pan_proc: for attitude estimation, and

4. PanDisp2: for displaying panoramic images.

Two sine wave generators are used to generate different roll and pitch rotations with different phases. Two scaling factors are also needed to compensate the difference between the vision attitude estimate and the actual attitude and it is done using gain

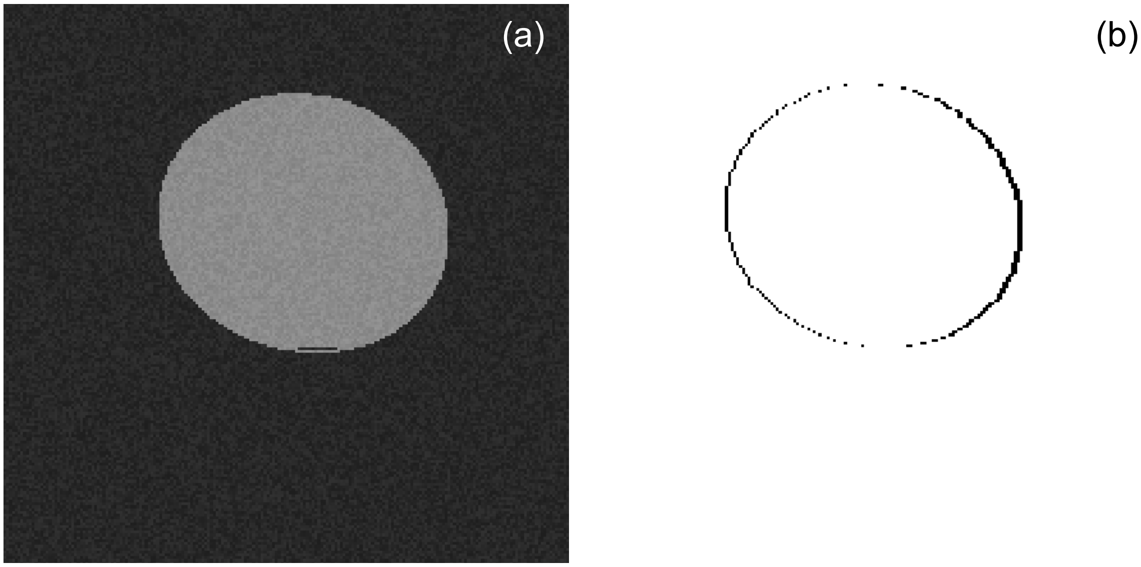


Figure 4.11: (a): Synthetic rotated panoramic image, (b): Synthetic rotated panoramic image after edge detection

boxes. The synthetic panoramic image and also the image after `pan_edges` is shown in Figure 4.11.

The estimated attitude after running the model for 20 seconds is showing in Figure 4.12. The angles are converted from radian to degree with R2D box.

There is a good match between the vision and reference attitudes. It is shown throughout the simulation that when a panoramic image is created from reference rotations, the reference rotations can also be calculated from a rotated panoramic image. It is noted that in real situations, when the camera is mounted on the Eagle in real surrounding environment, the image has many imperfections not present in the simulation as there are uncertainties, such as occluding objects, the effect of the sun and platform vibrations affect the image quality in different ways.

4.6 Real–Time Implementation

After evaluating the quality of the techniques in off–line processing, their properties are tested in a real–time environment. The feasibility of real–time implementation is important because the final goal is to estimate the real–time attitude of an aerial vehicle. Real–time attitude with minimal latency and deterministic delivery is essential

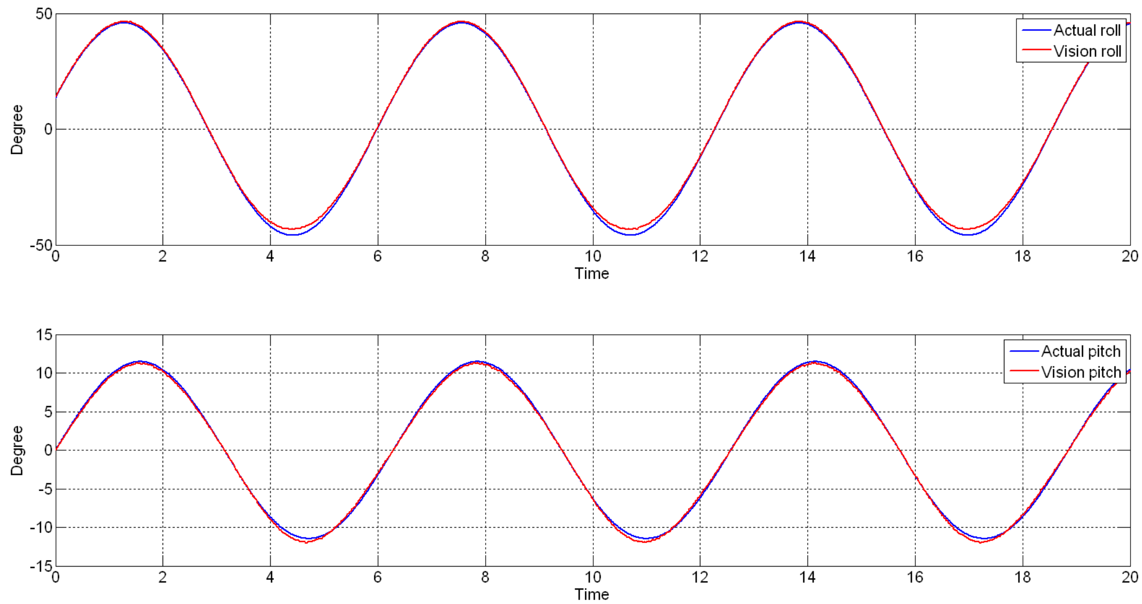


Figure 4.12: Attitude comparison (vision attitude vs actual attitude)

for automatic flight control systems. A PC104 computer system with Real–Time Linux (RTLinux) operating system is the test–bed for real–time programming and examining the performance of the techniques. The PC104 video grabber has two functions: firstly, to capture 50 frames a second, and secondly, to interrupt the system. A thread is woken up by an interrupt from the frame–grabber when a new frame is ready. This results in the thread executing every 20ms. When new interrupt occurs, a real–time handler calls a modular (already loaded into the Linux kernel) containing the programs to be real–time executed.

The main aim is to validate the computational efficiency and the quality of the algorithms when implemented in real–time. Different experiments are conducted in various weather conditions to test and optimise the algorithms for robust and efficient behaviors in all scenarios. All techniques which were explained in the sections 4.3.1–4.3.4 and 4.4 are written in C codes for compilation in the PC104. Figure 4.13 shows the flowchart of the process in which the attitude is estimated.

When the thread is woken, the system then performs specific tasks. In each interval, the following tasks are regularly performed for the horizon–based attitude estimation:

1. A new image (odd or even frame) is captured by the frame grabber,

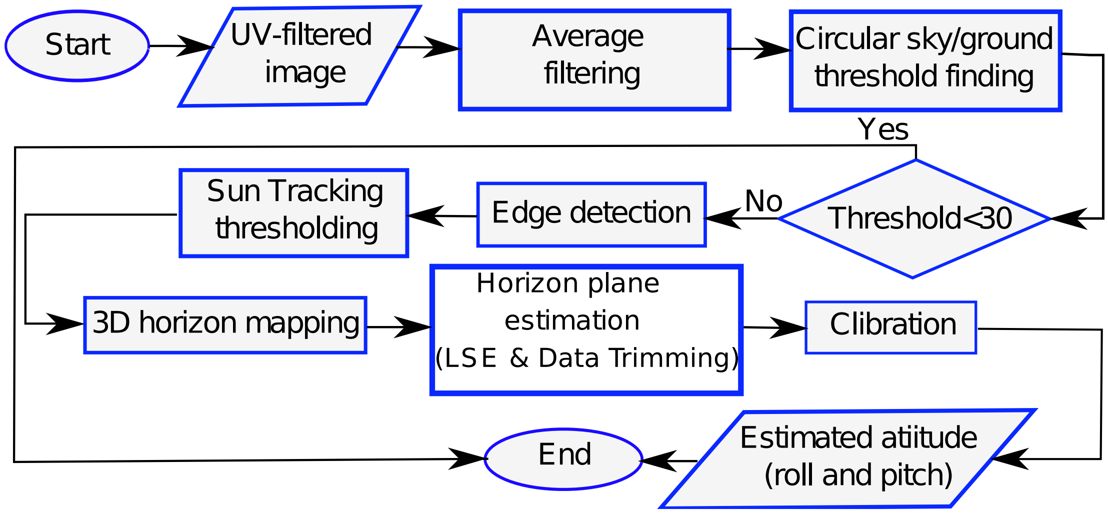


Figure 4.13: Attitude estimation flowchart

2. A new attitude data is provided from XSENS sensor (AHRS) through the RS232 communication port, and

3. Real–time attitude estimation code is executed (flowchart 4.13).

The camera was mounted on the Eagle and images were captured and processed by the PC104, which is attached to the Eagle's undercarriage as shown in Figure 4.14. The image in Figure 4.14a shows the surroundings on a fully cloudy day. The image in Figure 4.14b is a panoramic image after averaging filter with the defocused lens. The camera was defocused and then the image was filtered by averaging filter and this procedure reduced the unwanted scattered edges. The image in Figure 4.14c shows the thresholded image. Figure 4.14d is the image after the edge detection. The image in Figure 4.14e shows purely horizon edge after unwanted edges such as the undercarriage were taken away of the image. Figure 4.14f shows the horizon plane fitting. Figure 4.14g is also the horizon fitting but rotated to show the horizon points better.

The sun flare is still one of the main problems for horizon detection despite using a UV filter (combination of two filters) and mitigating the effect of the sun. The sun luminosity dramatically increases the brightness intensity of parts of the image and has a negative effect on the rest of the image. Sometimes, the sun luminosity reflects from the optics or from an object on the ground. The Sun–Tracking thresholding method

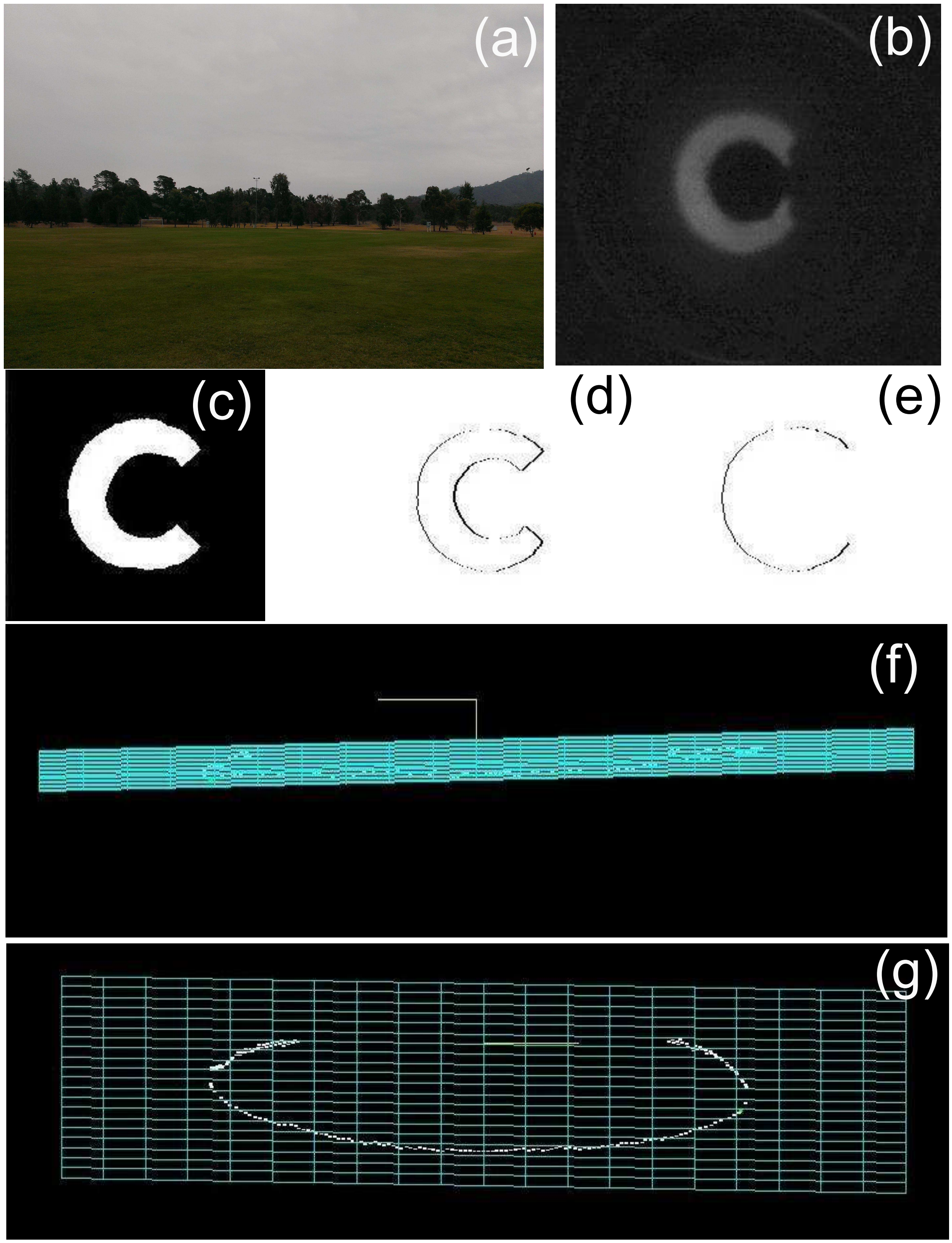


Figure 4.14: Real—-time horizon plane estimation

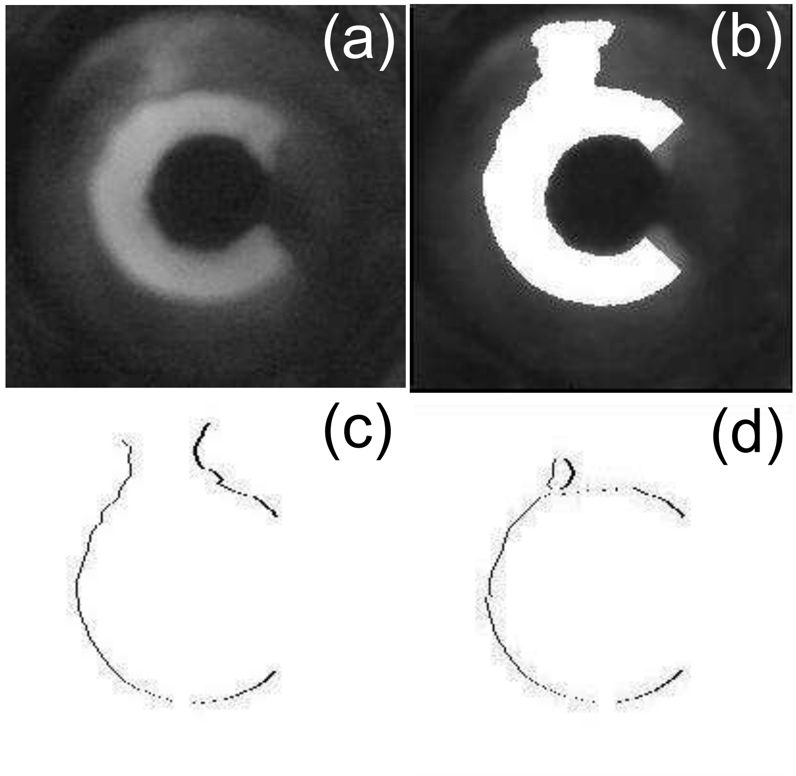


Figure 4.15: Effect of the Sun–Tracking Thresholding on the horizon extraction

which is proposed in this work improves the horizon extraction and its effects can be seen in Figure 4.15. The experiment was conducted on a sunny day to evaluate the performance of the method in the worst weather conditions.

Figure 4.15a shows the panoramic image where the ground darkness is affected by the sun flare. The effect of the sun luminosity on the image quality can be seen by comparing Figure 4.14b and 4.15a. The negative effect of the sun flare on the whole image can be seen in Figure 4.15a. Figure 4.15b shows the sky thresholding without applying the Sun–Tracking threshold around the sun. Edge detection is performed then and shown in Figure 4.15c. Figure 4.15d shows the horizon edge detection after applying the Sun–Tracking thresholding on the image. It can be seen that the quality of the horizon extraction is much better and the missing part has been recovered.

4.7 Flight Test

The most challenging part of the research is when the camera is mounted on the Eagle. Because the Eagle is a rotary–wing aircraft, it faces excessive amounts of vibration due to the rotating parts. This substantially affects the accelerometers, and can result in an unreliable measurement of the gravity vector. One aim of conducting this research is to show the visual definition of gravity vector is more reliable than inertial sensors where there is severe vibration in the platform, and the visual reference of gravity can be used as a reference for vertical direction. The inertial sensors are unable to provide the gravity direction when the inertial measurements are combined with platform vibration (different vibration characteristics with different magnitude due to rotary parts) and platform translational motions. These factors are combined with the gravitational accelerations and cause the gravity vector to deviate from its actual direction. In an advanced IMU, these measurements are highly filtered and processed by on–board processing units. It will be shown that in contrast to the inertial measurement, vision–based estimation (unfiltered estimations) is not influenced by the platform translational movement and is less affected by the platform vibration.

The Eagle body, depending on where the camera is mounted, blocks a region of camera view. As can be seen in Figure 3.10, the camera is attached to the front of Eagle, below the nose. In this mounting location, 90° degrees of the image is occluded by the undercarriage and there is no attitude information, so the visual information of this region should be taken out of the calculations. The undercarriage is visible in Figure 4.15a and consequently the Sun–Tracking thresholding is not also performed on this region (Figure 4.15b). The undercarriage blocks 90° of the image which the pure pitch rotation refers to. All edges in the blockage region are blanked out, with no effect on the attitude calculation. Figure 4.16 shows the Eagle flying in the cluttered environment (trees can be seen) with around 1 meter altitude above the ground. The pilot for all experiments in this work is Dr Matt Garratt [96].

Figure 4.16: Eagle flight in cluttered environment (Pilot: Matt Garratt)

Figure 4.17 shows the results that were taken during a 520 second flight (8.6 minutes) on a sunny day with natural environment with occluding objects around the Eagle. There were two constant attitude offsets (thet were added to the final estimations) on both angles due to misalignment. The offset values were found by comparing different results from the vision and the XSENS. Figures 4.17a and 4.17c show the comparison of roll and pitch angles during the flight. For a better look, Figures 4.17b and 4.17d show a close up plotting between sampling times from 100 to 200 seconds. As can be seen, there is a good attitude match between the vision and the XSENS. The RMS errors of roll and pitch are 0.97° and 1.3° which was calculated for the duration of flight (after take off and before landing).

The vision system can estimate accurate attitude when the Eagle is flying at low altitude (usually one meter above the ground), and in environment where it is evenly surrounded by occluding objects. When flying at low altitude (less than five meters), occluding objects mainly include trees, buildings and moving objects such as vehicles.

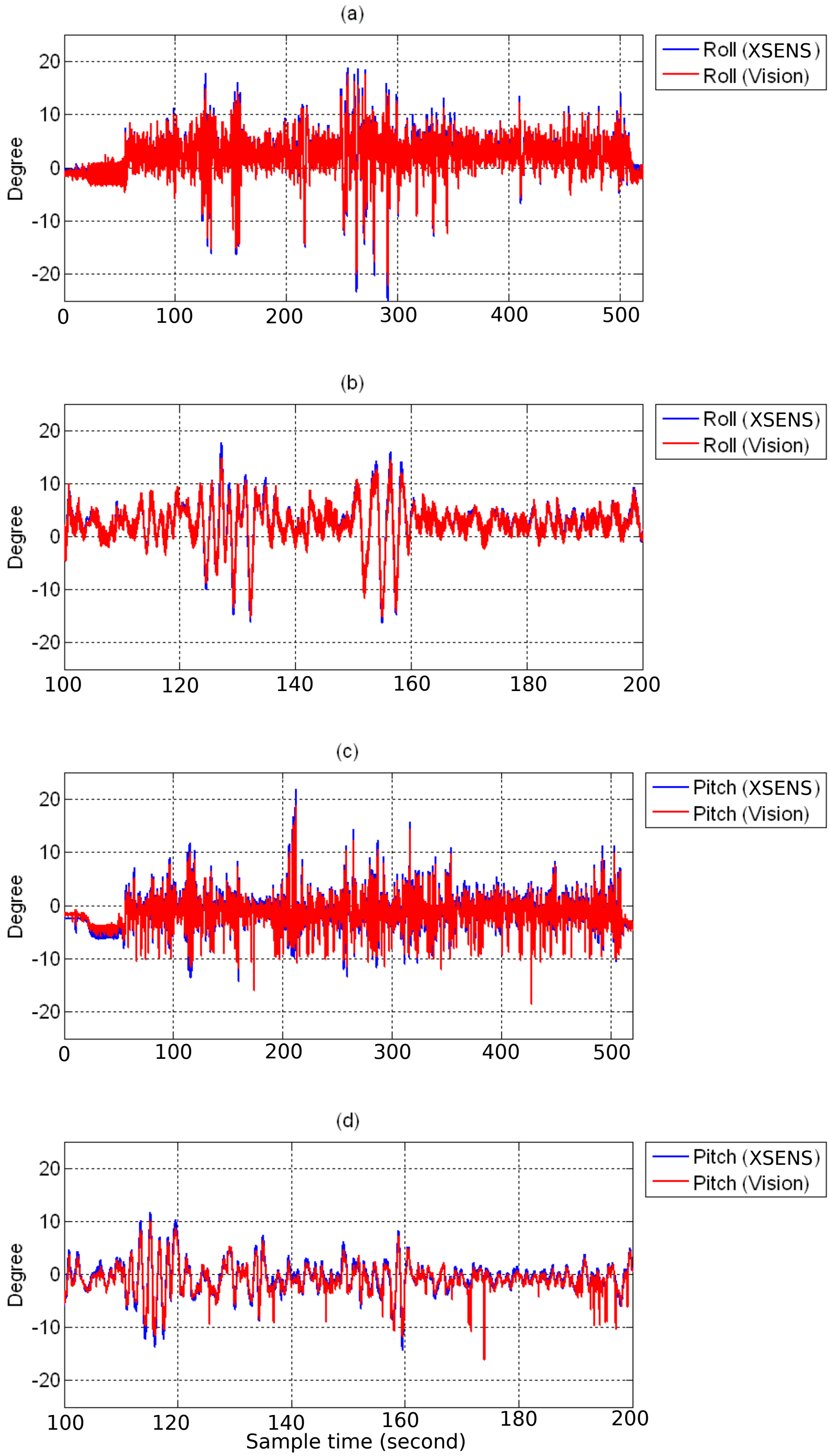


Figure 4.17: Real–time flying Eagle's attitude estimation

The new imaging system and the novel horizon estimation method has some advantages over the previous works. This included:

1. The camera provides a panoramic view of the surroundings with high probability of horizon visibility,
2. It provides good horizon extraction in different weather conditions (sunny and cloudy sky) and also when the horizon is not clearly visible (hazy and foggy horizon),
3. The effect of sun luminosity is mitigated throughout UV filtering of image and the affected horizon part was recovered using Sun–Tracking thresholding,
4. The attitude is estimated using 3D mapping of horizon to incorporate all horizon points into the attitude calculation,
5. It estimates the attitude of a rotary–wing aircraft with excessive amount of platform vibration, and
6. It estimates the attitude is a cluttered environment at low altitude.

There might be an offset value in the attitude estimation due to the effect of close objects. The offset value varies depending on the size of object and its distance to the camera. Many experiments have been done to investigate the effect of close objects on the attitude offset. The offset was negligible in conditions when:

1. The Eagle is not too close to a big object (e.g. less than four meters such as a building),
2. There is no natural big object on at least one side of the scenery (for example a mountain).

4.8 Gravity Direction

One of the objectives of this research is to investigate the effect of vibration on vision and whether vision can provide better gravity direction than accelerometers, used for attitude estimation or correction. In inertial sensors, the direction of gravity is calculated from accelerations on three orthogonal axes. In light of shortcomings of inertial sensors that were mentioned earlier, it is essential to have another way of determining gravity direction when inertial sensors are corrupted by external sources and not reliable.

The attitude due to the rotational movement (not translational) is calculated directly from three normalized accelerations (Equation 4.16 and 6.33) as:

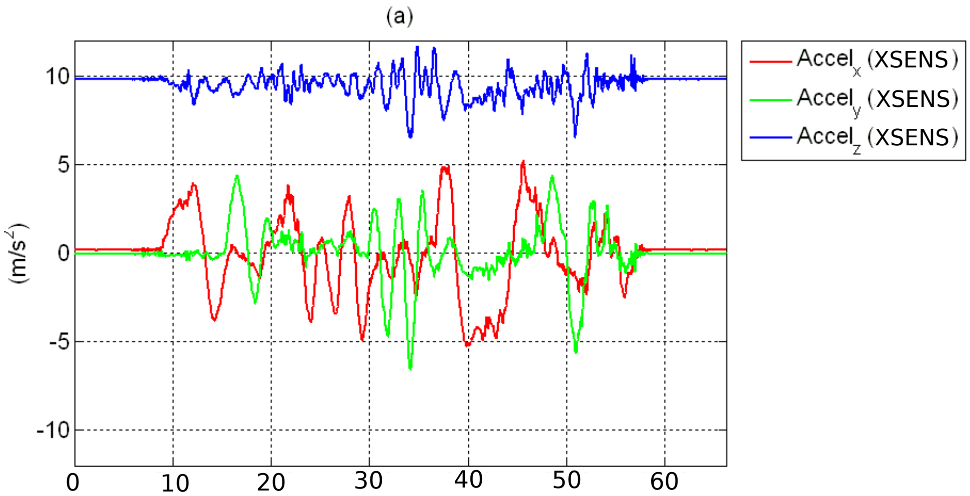
$$\begin{aligned} |A| &= \sqrt{(a_x)^2 + (a_y)^2 + (a_z)^2} \\ \hat{a_x} &= \frac{a_x}{|A|},\ \hat{a_y} = \frac{a_y}{|A|},\ \hat{a_z} = \frac{a_z}{|A|} \end{aligned} \tag{4.16}$$

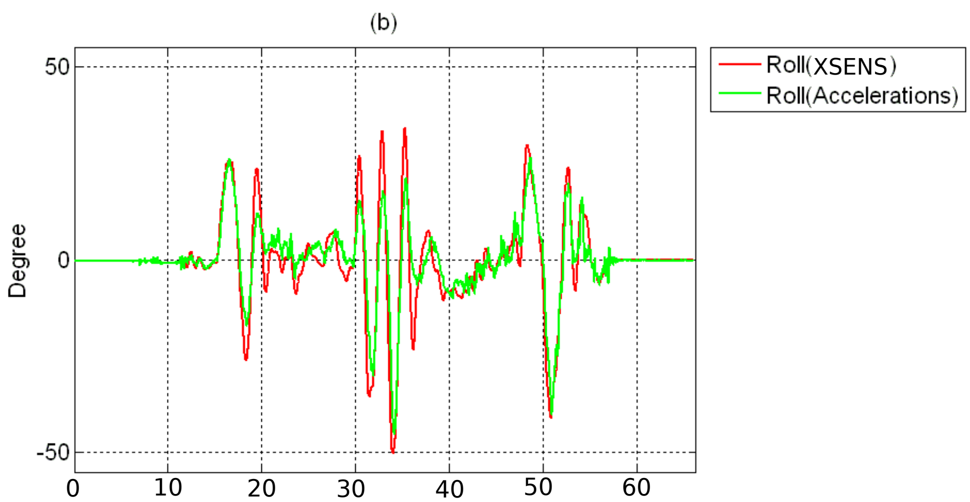
$$\begin{aligned} \phi &= \tan^{-1} \frac{\hat{a_y}}{\hat{a_z}} \\ \theta &= -\tan^{-1} \frac{\hat{a_x}}{\sqrt{\hat{a_y}^2 + \hat{a_z}^2}} \end{aligned} \tag{4.17}$$

where $[a_x, a_y, a_z]^T$ are the accelerations in the X, Y and Z directions, $|A|$ is the magnitude of the acceleration vector and ϕ and θ are the roll and pitch angles respectively.

The vision–based attitude is calculated from the normal vector of the 3D horizon plane which is perpendicular to the plane and represents the direction of vertical. In the Equation of 3D plane (Equation 4.10), the three coefficients A, B and C are the three orthogonal components of the plane in x, y and z direction which correspond to the three accelerations in x, y and z directions from the inertial sensors.

Accurate attitude of a platform can be calculated from the accelerations when the platform faces pure gravity accelerations (see Figure 4.18). Figure 4.18a shows the





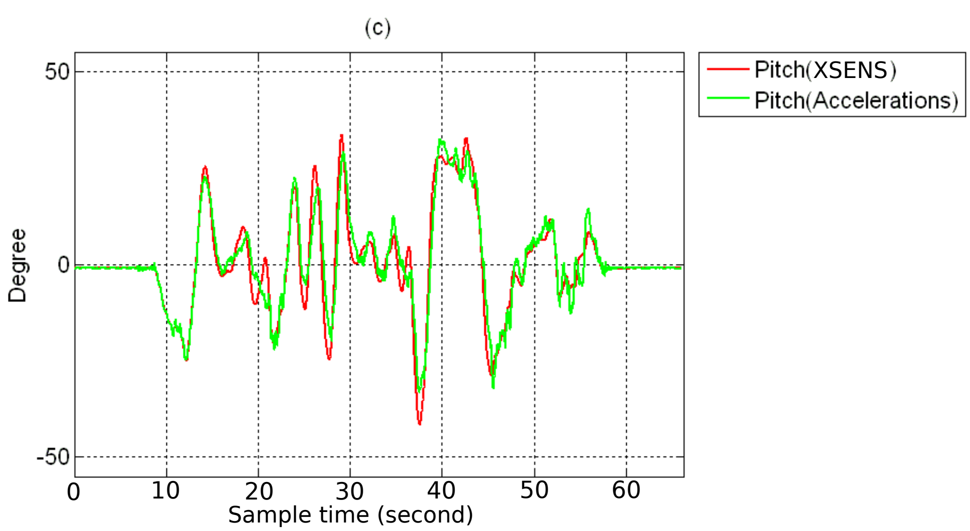


Figure 4.18: Direct attitude calculation using accelerations during stable movement

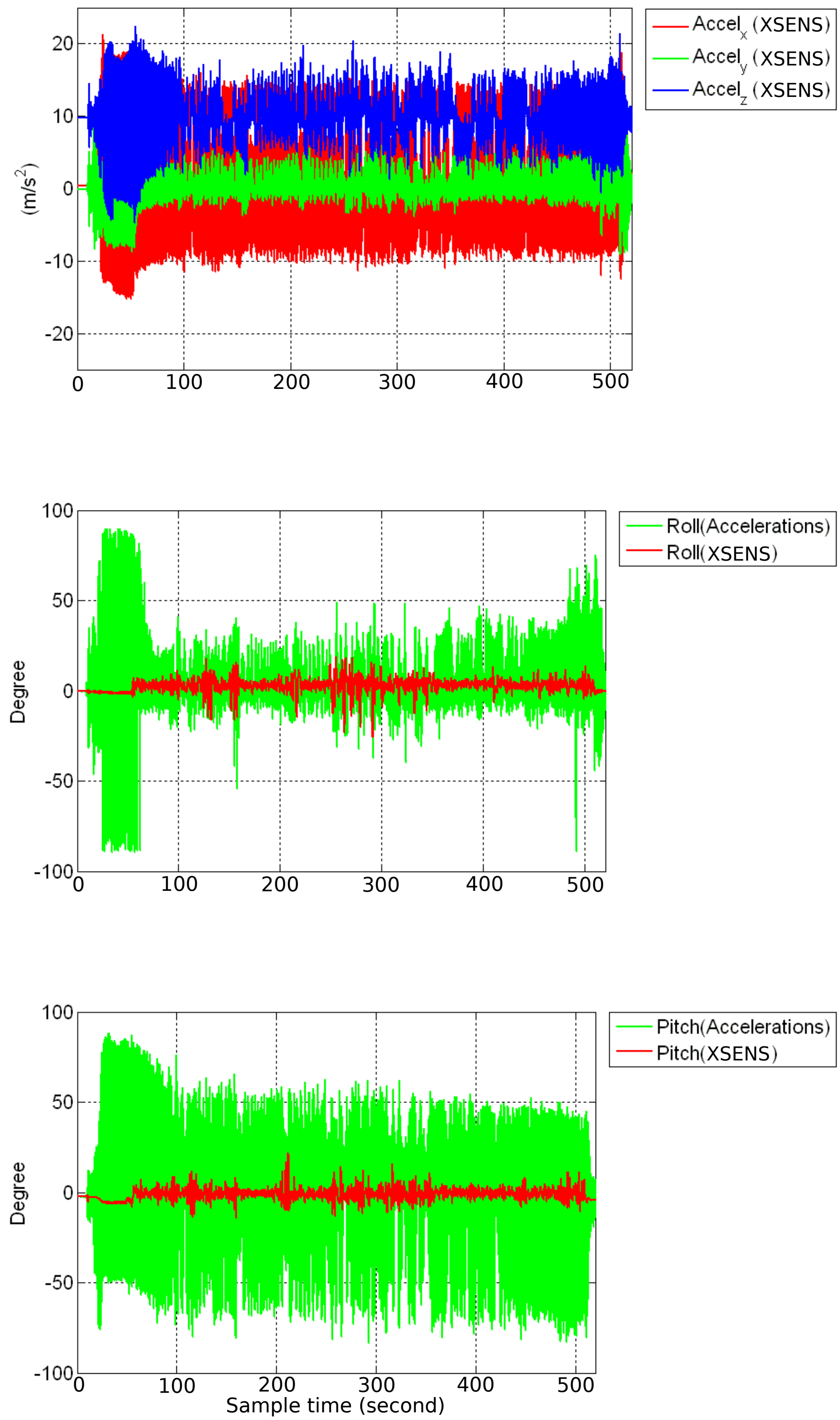


Figure 4.19: Direct attitude calculation using accelerations during flight

accelerations sensed by XSENS accelerometers while the Eagle is held and moved by hand. Roll and pitch angles are calculated from accelerations using Equation 6.33 and shown in 4.18b and 4.18c. It can been seen that with smooth platform movement, the attitude can be calculated from accelerations. When the platform is moving or vibrating, additional forces interfere with the gravity accelerations. In such situations, the gravity direction is corrupted and consequently the attitude estimation becomes incorrect. Figure 4.19a shows the accelerations during an Eagle flight. The effect of the platform vibration on the acceleration can be clearly seen. Figures 4.19b and 4.19c show the roll and pitch angles calculated using Equation 6.33. It can be seen that attitude can not be calculated from the accelerations when vibration exists in the acceleration measurement.

The direction of gravity can be found using the horizon plane normal vector. Figure 4.20 shows the components of vertical direction vector from the XSENS and the vision. Figure 4.20a shows the XSENS accelerations when the Eagle is moving smoothly by hand, Figure 4.20b shows the XSENS accelerations when the Eagle is hovering and Figure 4.20c shows components of the horizon plane normal vector. The data shown in Figures 4.20b and 4.20c is logged during the same flight test. By comparing Figure 4.20b and Figure 4.20c, it can be seen that the vision is less affected by vibration than accelerometers in providing the vertical direction. Roll and pitch angles are calculated from the plane coefficients (Figure 4.20a) and shown in the Figure 4.17.

4.9 Summary

In this chapter, a new method for attitude estimation using the horizon was implemented and verified through the simulations and real flight tests. It was shown that using the horizon's plane rather than line could be used as a reliable reference to estimate the attitude. The advantage of using the vision system rather than inertial sensors to define the gravity direction when there was vibrational and translational movements was highlighted. Sometimes, the horizon line was not properly extracted due to some

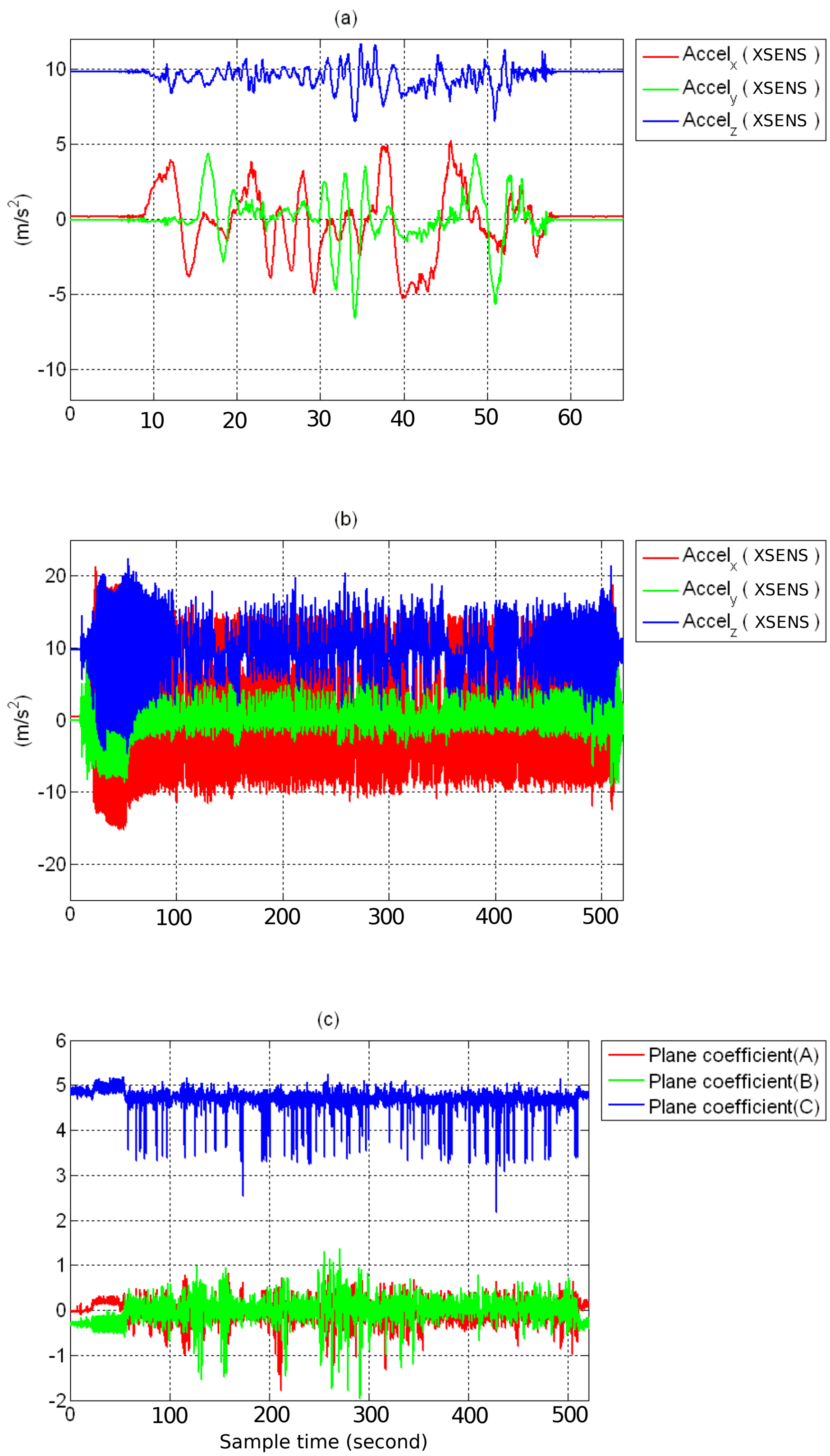


Figure 4.20: Components of gravity vectors, (a) XSENS gravity components of smoothly moving Eagle (not flying), (b) XSENS gravity components during Eagle flying, (c) vision gravity components during Eagle flying

external factors, such as the sun's flare or close objects, and then the attitude estimation was not reliable and another reference was needed. In the next chapter, the rate of attitude change is calculated from a panoramic image using an optic flow technique which provides additional attitude data for use with horizon–based attitude data, and offers more robust attitude estimations with sensor fusion techniques than raw data.

Chapter 5

Optic Flow Estimation

5.1 Introduction

This chapter investigates the feasibility of calculating optic flow in UV–filtered panoramic images. Although optic flow has been used for many years to calculate image motion using a single camera with a limited viewing angle, a camera with a panoramic lens can provide a complete with 360° view of the surroundings. Calculating a panoramic image's motion using optic flow can provide good estimations of the angular rates of the platform to which the camera is attached. Chapter 4 discusses how the attitude can be estimated from the horizon, which can be considered as a reliable attitude reference. However sometimes the horizon–based estimation can be incorrect due to the effect of the sun's flare or occluding objects. In such conditions, another attitude reference can be used to estimate attitude changes, or be integrated with horizon–based attitude estimation to improve accuracy. As panoramic images can almost completely represent the surrounding environment, a movement of the platform (to which the camera is attached) can be projected onto the panoramic lens and captured by the camera. Due to the common errors in IMUs [[21], [22], [125]], it is necessary to reduce the drift caused by bounded and unbounded errors in navigational estimations. Althought, gyroscopes and accelerometers are accurate devices, they are vulnerable to temperature changes. Thermal effects add an offset into measurements (especially on gyros of cheap IMUs) that cause the attitude to drift over time. Experimental work has shown that in contrast to MEMS devices, vision systems are invulnerable to thermal effects and less sensitive to vibration. As, to achieve the goal of estimating the attitude from an independent vision system, angular rates should be estimated, in this chapter, the feasibility of determining them from panoramic images is investigated by implementing an optic flow technique using image interpolation algorithm (I^2A) and comparing the results with those from the XSENS sensor. Calculations of the angular rate from the panoramic images can replace those from the gyros to obtain independent vision–based attitude estimation.

The motivation behind using optic flow for attitude estimation comes from the vision of insects and how they interpret different visual features for navigation. Insect flight has been studied by many researchers, who found that insects relied more on their vision system than any other sensory information. Also, the change in the quality of insect flight by artificially changing visual features has been investigated by several authors. Insects use the horizon as a reference for equilibrium flight control through their Ocelli [126], and when there are not enough visual features (such as the horizon), they rely on apparent pattern motions. Useful motional information can be obtained from the apparent motions of visual features in consecutive images by means of optic flow. The effect of the textural density and light distribution of images on perception were verified in [127] whitch showed the correlation of texture, slant, and distance on the retina. Also, the effects of fluctuations in light's intensity on different receptors has been studied [68]. There are different receptors which are sensitive to different motions, such as Medulla and lobula cells [[69],[68]], LPTCs with different motion sensitivity including horizontal (HS), centrifugal horizontal (CH), and vertical (VS) cells, H1 and H2 cells located in LPTCs which are sensitive for horizontal feature–detecting [69] (see [70] for more). Each cell in LPTCs has the ability to detect visual features, for example CH cell are more sensitive to rotational motions and used specifically to detect rotational optic flow [128]. These types of receptors are used to provide additional visual information to the insect. The effect of different visual patterns on insect optic flow estimations has been investigated by different researchers. Different visual patterns and light intensities can affect insect's visual interpretations and consequently, flight control (as e.g. investigated in [10]). Both day–active insects and night–active insects rely on visual pattern motions, for example, Emily Baired et al [10] compared the flight behaviour of two bee species, a night–active (sweat bee (megalopta genalis)) and a day–active one (bumble–bees (bombus terrestris)). Their investigation showed that despite some differences in their flight speeds, different visual patterns changed the way they bees flew and that they flew faster when optic flow clues were strong but reduces their speed in dim light due to the lack of the visual clues required to obtain a high optic rate. An insect integrates visual clues (a process called temporal summation) to gain a reliable information about its environment. In contrast to the day–active insect (bumble–bee), the night–active one (megalopta genalis) increased its ground speed with fewer clues. The result revealed that visual effects had different meanings for different insects as they interpreted them differently. It was proven through biological experiments (see literature review) that insects instinctively try to keep a constant image velocity on both sides (projected on their compound eyes) to fly safely in narrow passages, by trying to change their position to balance the optic flow speeds on both sides. When visual information is poor (for example flying in a tunnel with no visual features such as the horizon), an insect relies mainly on the optic flow to detect flight motions. This chapter is organised as follows: section 5.2 describes the concept of the I^2A technique, section 5.3 shows the MATLAB SIMULINK implementation of the I^2A, section 5.4 provides the results of I^2A obtained by the I^2A during a real flight test, and section 5.5 presents conclusion.

5.2 The Image Interpolation Algorithm (I^2A)

The angular rates of attitude change are important parameters for attitude estimation. In an AHRS, they are measured by gyroscopes and the attitude is propagated throughout the attitude algorithm by integrating them. Any sensory drift from the actual value can cause the AHRS to fail to provide a reliable attitude estimate. Optic flow can be used as a method for calculating the angular rates when no gyroscope is not used or the system relies completely on vision, and for calculating image displacements from image motions. As a panoramic image (taken by a panoramic lens) contains a spacial representation of the environment, the visual effect of any platform rotations enters it. As, unlike the inconsistent thermal drift in gyroscopes, visual data is not affected by temperature variations and is less sensitive. So the platform rotation can be calculated by tracking image motions. Calculating angular rates from image motions leads us to design a vision system that works independently. Also, vision–based angular rates can be integrated with the horizon–based attitude to provide better attitude estimations. Optic flow methods are categorised into four classes: gradient, regional, energy and phase–based, and each has advantages and disadvantages. The most effective technique (which is used in this work) should:

1. be computationally inexpensive and suitable for real–time implementation in the PC104,
2. perform well on a UV–filtered image with poor texture, and
3. estimate the optic flow in images with rapid sub–pixel displacement and an excessive amount of vibration.

As mentioned in Chapters 3 and 4, a real–time thread is activated every 20ms so that total calculation time should not exceed the time limit as this could cause the system to crash (hang) or provide incorrect calculation.

The quality of an image's texture is important as images with poor quality have less visual information than a rich one. Vibration can affect image quality by blurring and degrading an image's sharpness, with degree of blurriness depending on how greatly the images shack. Usually it is difficult for the naked–eye to detect a low level of shacking effect in an image, especially for a UV–filtered one with poor texture. It is shown that low image displacement (due to the vibration of the platform motion) can be detected by the I^2A. Depending on the platform's vibrational characteristics, vibration shows up as a high frequency modulated signal on the actual attitude measurement and should be discriminated from legitimate measurement. The Eagle faces an excessive amount of vibration due to its rotating parts. The optic flow method should be able to calculate the image motion taken by a camera from a highly vibrating platform. The platform vibration causes the image pixels to be constantly displaced for a particular amount (relative to how much the platform shacks) and this displacement is added to the image motion calculation.

By evaluating the characteristics of existing methods and the limitations of this work, region–based optic flow is chosen for implementation. The Image Interpolation Algorithm (I^2A) was developed by Srinivasan [91] is a non–iterative process that computes image motions (translation and rotation) by interpolating a set of reference images, with no particular feature tracking or image velocity calculation needed. Image velocity estimation is based on the fact that during a displacement, the deformation of two images is linear and continuous. The image displacement in 2D is calculated from six reference frames shifted by two reference shifts in the x and y axes. It is robust to noisy images [91] and its accuracy dependent on the reference shift.

The intent in this work is to measure image translation along two image axes (x and y) of two consequentive images and in a panoramic image, the aim is to track the Earth's movement. When rolling or pitching, the horizon displacement (Figure 5.5) is observable in 2D and the respective image displacements can be measured by the I^2A in the form of pixel displacements. Versions of the I^2A exist for pure translation and combined translation and rotation (yaw) [91]. In this work, the effects of yaw on images have been ignored as yaw rotations are not observable in UV–filtered panoramic images because:

1. yawing does not visually affect the shape of the horizon circle, and

2. as the image has poor ground texture, these is no visual information for the I^2A to interpolated for yaw calculations.

The I^2A relies on the assumption that image motion at time t can be linearly interpolated from f_0 and the reference images. The intensity function is calculated within a patch located on two images denoted by $f(x, y)$ and $f_0(x, y)$. Six reference images are obtained as the result of shifting the image f_0(x, y) for Δx_{ref} and Δy_{ref} and are calculated as follows:

$$
\begin{aligned}
f_1(x, y) &= f_0(x + \Delta x_{ref}, y) \\
f_2(x, y) &= f_0(x - \Delta x_{ref}, y) \\
f_3(x, y) &= f_0(x, y + \Delta y_{ref}) \\
f_4(x, y) &= f_0(x, y - \Delta y_{ref}) \\
f_5(x, y) &= f_0(x, y) \\
f_6(x, y) &= f(x, y)
\end{aligned}
\tag{5.1}
$$

The pixel intensity functions at times t_0 and t are $f0(x, y)$ and $f(x, y)$ respectively where x and y are the image coordinates measured in pixels. The reference images $(f_1 - f_6)$ are formed from the first image by shifting them by the reference shifts (Δx_{ref} and Δy_{ref}) pixels along the x and y axes, so that the Equation 5.1 applies. The image displacement within a patch at time t can be interpolated from f_0 and the reference images using Equation 5.7.

The image translation from f_0 to f can be approximated using the shifted reference images (f_1 to f_6), reference rotation by:

$$
\hat{f} = f_0 + 0.5\left(\frac{\widehat{\Delta x}}{\Delta x_{ref}}\right)(f_2 - f_1) + 0.5\left(\frac{\widehat{\Delta y}}{\Delta y_{ref}}\right)(f_4 - f_3) \tag{5.2}
$$

The motions between f and f_0 are computed by calculating $\widehat{\Delta x}$, $\widehat{\Delta y}$ with minimised error between image f and $\hat{f}$. The error is minimised over window patches by function Ψ. The least–square error is defined as:

$$\mathrm{E} = \int\int \Psi.\left[f - \hat{f}\right]^2 dx.dy \tag{5.3}$$

Substituting Equation 5.2 in Equation 5.3 results Equations 5.4 and 5.5.

$$\begin{aligned}
&\left(\frac{\widehat{\Delta x}}{\Delta x_{ref}}\right)\int\int \Psi.(f_2 - f_1)^2 dx.dy(a)\\
&+\left(\frac{\widehat{\Delta y}}{\Delta y_{ref}}\right)\int\int \Psi.(f_4 - f_3)(f_2 - f_1)dx.dy\\
&= 2\int\int \Psi.(f_6 - f_5)(f_2 - f_1)dx.dy
\end{aligned} \tag{5.4}$$

$$\begin{aligned}
&\left(\frac{\widehat{\Delta x}}{\Delta x_{ref}}\right)\int\int \Psi.(f_2 - f_1)(f_4 - f_3)dx.dy\\
&+\left(\frac{\widehat{\Delta y}}{\Delta y_{ref}}\right)\int\int \Psi.(f_4 - f_3)^2 dx.dy\\
&= 2\int\int \Psi.(f_6 - f_5)(f_4 - f_3)dx.dy
\end{aligned} \tag{5.5}$$

The matrix form of Equations 5.4 and 5.5 is given in Equation 5.6.

$$2\int\int\begin{bmatrix}\Psi.(f_6 - f_5)(f_2 - f_1)\\ \Psi.(f_6 - f_5)(f_4 - f_3)\end{bmatrix} = \int\int\begin{bmatrix}(f_2 - f_1)^2 & (f_4 - f_3)(f_2 - f_1)\\ (f_2 - f_1)(f_4 - f_3) & (f_4 - f_3)^2\end{bmatrix} \times \begin{bmatrix}\frac{\widehat{\Delta x}}{\Delta x_{ref}}\\ \frac{\widehat{\Delta y}}{\Delta y_{ref}}\end{bmatrix} \tag{5.6}$$

The image translations $\widehat{\Delta x}$ $\widehat{\Delta y}$ in Equation 5.7 are two interpolated image displacement calculated with matrix inversion as:

$$\begin{bmatrix} \widehat{\Delta x} \\ \widehat{\Delta y} \end{bmatrix} = \frac{\int\int \begin{bmatrix} (f_2 - f_1)^2 & (f_4 - f_3)(f_2 - f_1) \\ (f_2 - f_1)(f_4 - f_3) & (f_4 - f_3)^2 \end{bmatrix} \times \begin{bmatrix} \frac{1}{\Delta x_{ref}} \\ \frac{1}{\Delta y_{ref}} \end{bmatrix}}{2\int\int \begin{bmatrix} \Psi.(f_6 - f_5)(f_2 - f_1) \\ \Psi.(f_6 - f_5)(f_4 - f_3) \end{bmatrix}} \tag{5.7}$$

5.3 I^{2}A Simulation using MATLAB SIMULINK

The I^{2}A algorithm is first tested using synthetic images created in MATLAB SIMULINK, with the I^{2}A program added to the SIMULINK model shown in Figure 4.10. Each synthetic image has 200×200 pixel resolution, with one 160×160 patch centred on pixel(100, 100). The reference shifts, Δx_{ref} and Δy_{ref}, are set to 15 in the experiments to provide the best I^{2}A performances (note that these values chosen must be larger than the expected image displacement). Image filtering is one of the procedures to be followed for optic flow calculation [78] to increase the signal–to–noise ratio and reduce aliasing. It increases the image smoothness by reducing rapid changes in its pixel intensity. An averaging filter is used because of its low computational time and high image–smoothing characteristic. The type of filter and its effect on the I^{2}A varies from one case to another and it depends mainly on the quality and expected displacement of the image, with the SIMULINK–created image pre–filtered by an 4×4 averaging filter. The choice of filter size varies from one condition to another, depending on the expected image quality or image translation, but shift reference is influenced by the expected image motion. For instance, if the image translation is expected to translate to five pixels, the reference shift of the patches should be greater than five, therefore, six reference images ($f_1 - f_6$) are extracted from the images for use by I^{2}A.

Sometimes an iterative I^{2}A calculation can be considered. In such a method, the I^{2}A is calculated in each iteration with a different filtered image and shift reference and the image translation from this iteration is used for the next iteration to obtain a better

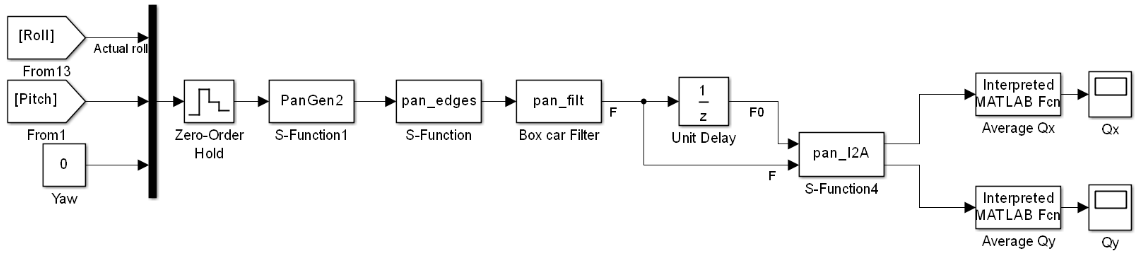


Figure 5.1: I^{2}A MATLAB SIMULINK

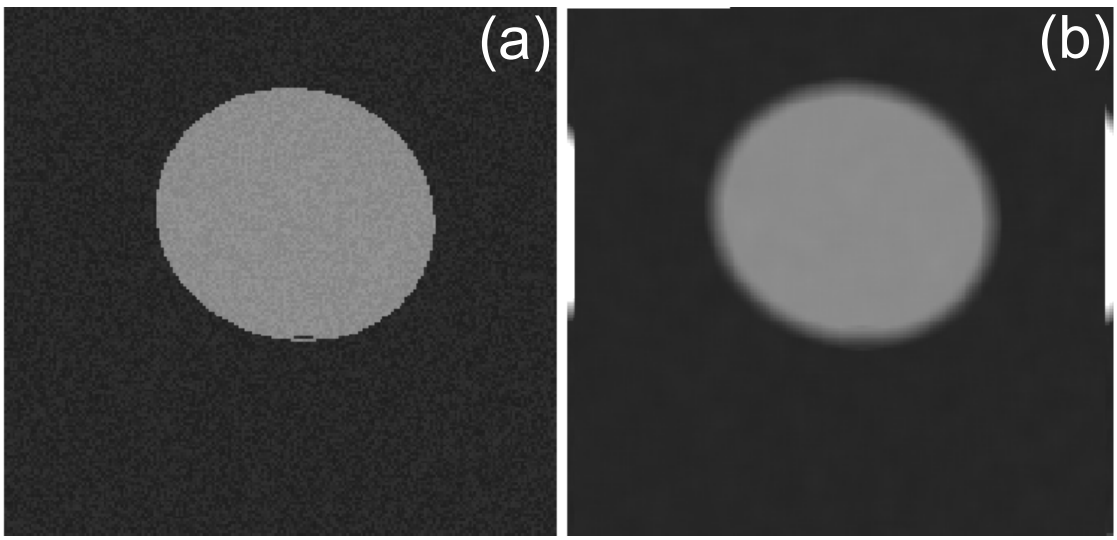


Figure 5.2: (a): Synthetic image, (b): filtered synthetic image

estimation, whereby, the filter size and shift reference value decrease. Although this method is also implemented in this work, no significant changes are seen on the results.

The I^{2}A MATLAB SIMULINK model is shown in Figure 5.1. Two sample series of roll and pitch are created using a movement simulation and used as a reference rotations to test the I^{2}A algorithm. Figure 5.3a depicts the input roll angle to the model. Figure 5.3b depicts the image displacement in term of pixels. Figure 5.3c depicts the integration of image velocity.

The image before and after box car filtering (an averaging filter) is shown in Figure 5.2, and the same I^{2}A calculation performed on pitch angle in Figure 5.4.

As can be seen in Figures 5.3 and 5.4, there are good matches between attitude changes and integrated image velocity although they are of different magnitudes. The magnitude of the input rates are 1.5 and 1.6 times larger than of those of the I^{2}A calculations of roll and pitch respectively. One reason behind obtaining different mag-

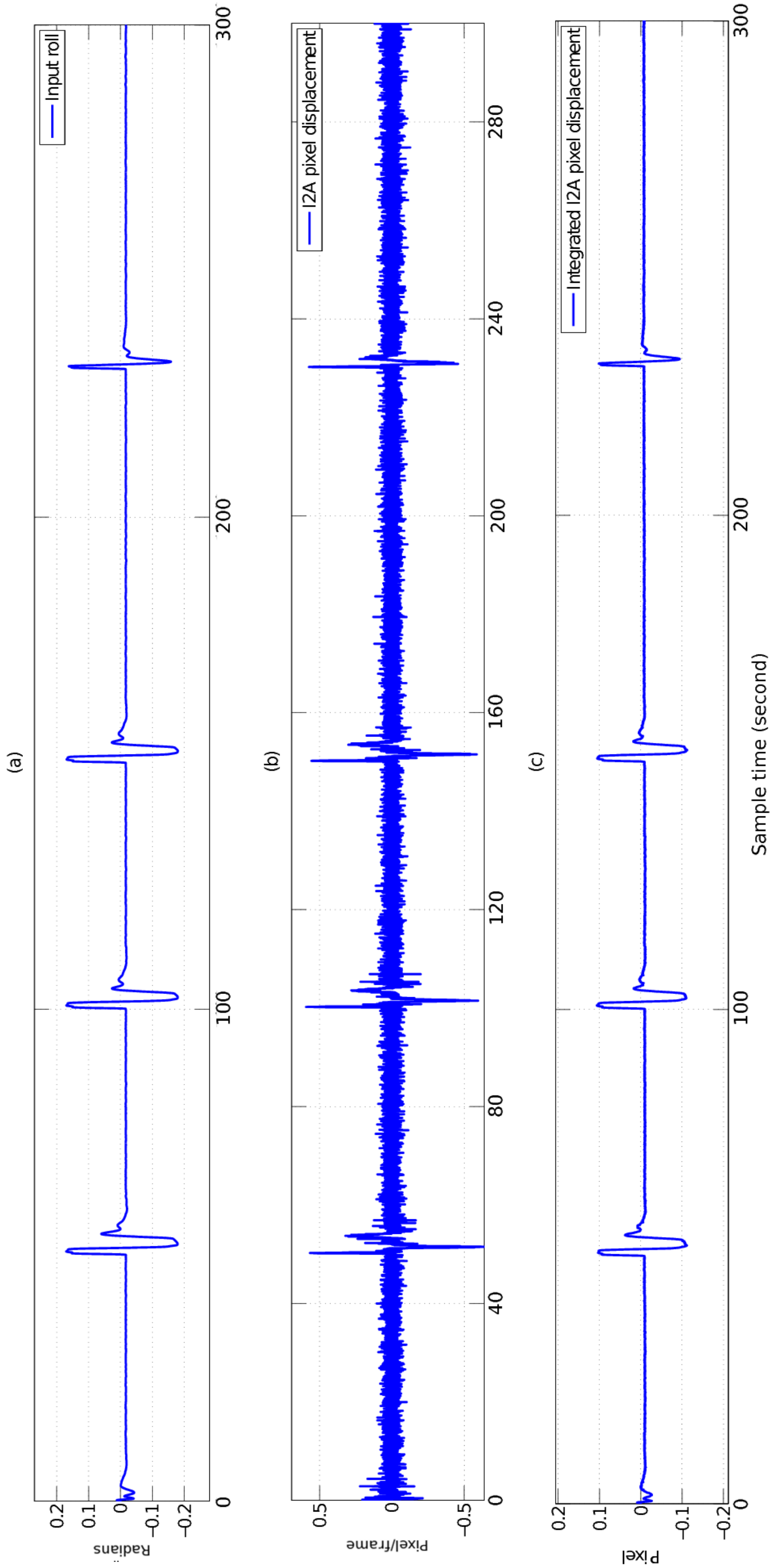


Figure 5.3: (a): Input roll, (b): Image displacement in pixel (I^2A), (c): Integrated image displacement in pixel (I^2A)

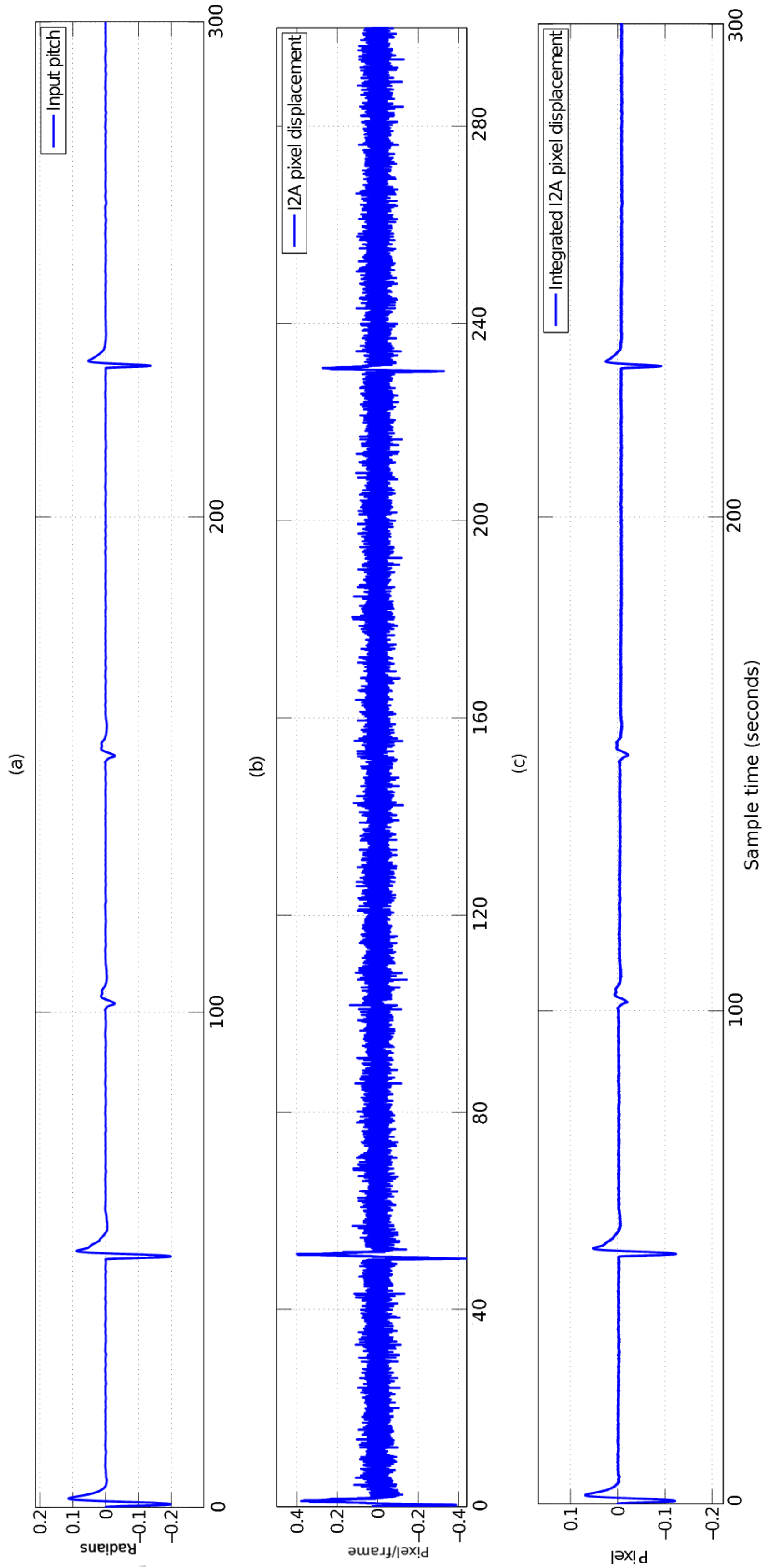


Figure 5.4: (a): Input pitch, (b): Image displacement in pixel (I^{2}A), (c): Integrated image displacement in pixel (I^{2}A)

nitudes is image deformation as, the more rotations, the fewer horizon circle movement are observed in x–y directions (Figure 5.6). It is shown that calculation of panoramic image motions can be considered as a source of representing the attitude changes. The panoramic image motion gets its effect from the rotational platform movement (where the camera is mounted), so the angular velocity of the platform can be calculated from the panoramic image movements.

5.4 Flight Test

As previously mentioned, gyroscopes have some shortcomings such as being sensitive to noise and thermal drift. One of the research objectives is to investigate the feasibility of measuring attitude rates using the I^2A. After evaluating the performance of I^2A in MATLAB SIMULINK, we investigated the quality of its estimations in real time using the flight images taken by the Eagle's camera. The quality and real time implementability of I^2A are also important because it should be executed in real time with time constraints and should be able to measure the Eagle's body rate for attitude propagation. The I^2A algorithm is written in C code in the PC104, and the I^2A program called every 20ms during the system interval when a new image arrives. A real image (when taken by the Eagle's mounted camera) has less motion information than a synthetic image because a quarter of image is occluded by the Eagle's undercarriage.

In Chapter 4, it was shown that the vibration in the Eagle has less effect on vision –based than inertial–based attitude estimation. The positive effect of optic flow on an insect's flight control is discussed in the literature review. Optic flow provides valuable data about the visual motions to be used for attitude estimation and its importance for accurate attitude estimation is highlighted in the sensor fusion chapter. Optic flow allows the system to work entirely through vision with no aid from inertial–sensors. Although, a panoramic image is not directly used by the I^2A, some modification are made on its features so that the I^2A provides best performance. The investigations include:

- thresholding both the sky and ground,
- examining the effect of Sun–Tracking thresholding,
- thresholding the sky with the intact ground, and
- determining the effect of the undercarriage and central in the lens and the I^{2}A patch.

After conducting many experiments, the best result of I^{2}A is achieved from the I^{2}A after applying the following modifications to the original panoramic image:

- the sky pixels are changed to white (pixel value=255) with the sky threshold found from the sky mask, while the ground texture remains intact, and
- Sun–Tracking thresholding is performed to recover the sun–affected part of the horizon.

Eliminating the sky texture by thresholding improves the accuracy of the I^{2}A estimation by reducing the effect of the sun on the image which creates a highly illuminated spot. As the moving sun can also affect I^{2}A's performance for tracking ground movements because its luminosity and flare influence the whole image, it should be tracked. As having a moving object such as the sun introduces an additional feature into the calculation of the Earth's tracking, the sky pixel values are converted into white (255 in a grey scale image). As the effects of the lens central whole and elimination of the undercarriage are negligible, they remained in the image. A typical image used for I^{2}A is shown in Figure 5.5. One square–shaped patch used is marked on the figure with the reference shift Δx_{ref} and Δy_{ref}. Images f_1 to f_5 are the new images created from image f_0. f_6 is the new image $\hat{f}$ in Equation 5.2 that its translations (with known Δx and Δy) is desired by the I^{2}A interpolation.

There are some strategies in which the I^{2}A accuracy may ,or may not be, enhanced depending on the application such as:

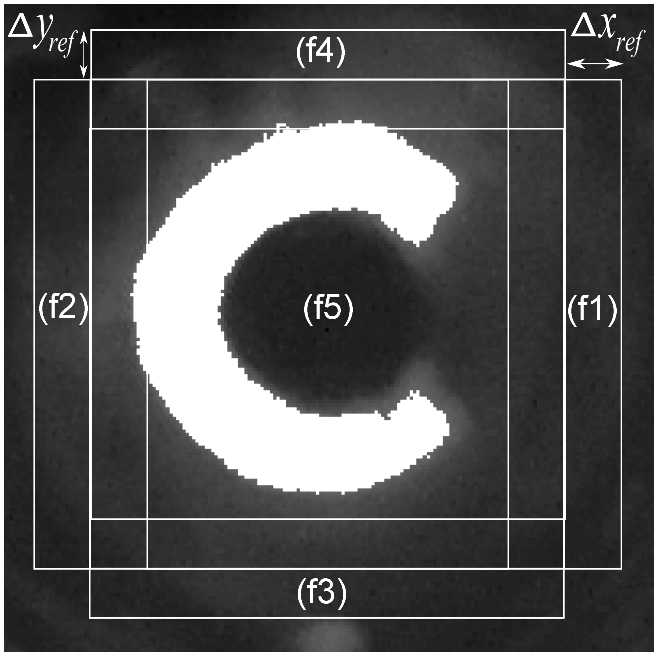


Figure 5.5: Typical panoramic image for I^2A calculation

- using more patches located on different parts of the image,
- performing an iterative I^2A to perform the I^2A calculation several times on the same images to converge to the best estimation, or
- changing the filter size (reducing the averaging filter size) in each iteration.

By increasing the number of iterations and filter size, the sizes of the patches and averaging filter are iteratively reduced. After conducting many experiments, one big patch with 8×8 averaging filter and no iteration had acceptable performance on an UV–filtered image (Figure 5.5) and increasing the number of iterations and the filter size has no significant effect on the result.

The results from the I^2A calculation are Δx and Δy, which correspond to the magnitudes of 2D image velocity. For clarity, the image displacements in the x and y directions (Δx and Δy) are shown as a 2D arrow in Figure 5.6, with a graphical widget created in the PC104 to show the I^2A arrow in real time.

An Eagle flight is carried out on a sunny day to compare the I^2A estimation and XSENS gyroscopic rate. Figure 5.7 shows the real time I^2A calculation of the roll axis

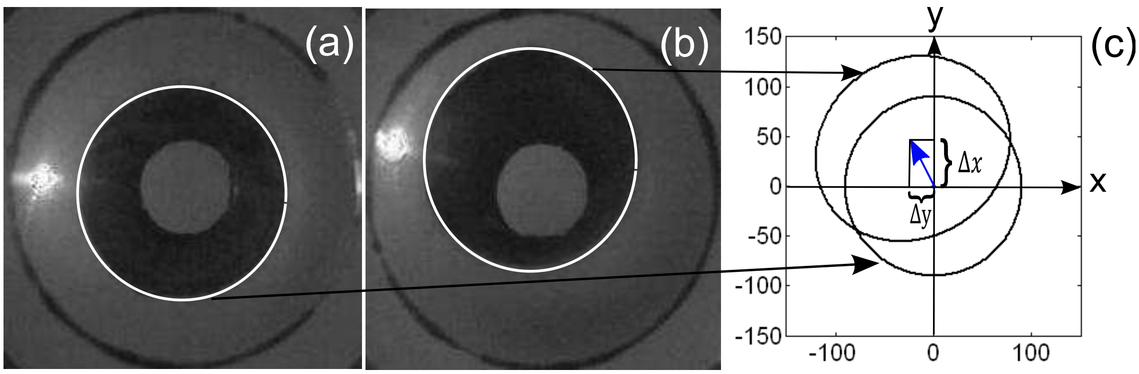


Figure 5.6: Definition of I^2A for UV–image feature tracking

during a 525 second flight. For better data visibility, the plot is magnified and shown in Figure 5.7b. To scale the I^2A with the XSENS, the I^2A results for the roll and pitch rates are multiplied by 2.5 and 0.7 respectively. Due to the effect of the existing non–linearity of image deformation (optical components) on the I^2A calculations, the values of the scaling factors (converting pixels per frame to radians per second) are calculated manually by comparing the results from various tests to try to achieve the best match. The difference between the scaling factors for pitch and roll is most likely caused by the misalignment between the centre of the camera lens and centre of the panoramic image, which results in asymmetry in the barrel distortion effect. It should be noted that Equations 3.6 and 3.7 in Chapter 3 have no effect on the I^2A estimations and are applied to the final attitude estimations (roll and pitch).

The program for calculating the I^2A displacement is run in real time and called after the horizon–based attitude estimation. Note that there is a varying lag between the I^2A and the XSENS due to their different sampling time (XSENS=100Hz and I^2A=50Hz) and computational times. The I^2A compares the new image with that taken at time $f_{t-1} = f_0$ 20ms ago. There is a lag between the XSENS rate and I^2A calculation that is manually compensated in the Figure 5.7b for a better comparison of results. The test is conducted on a sunny day on which the sun flare caused some pixels to become saturated. In this condition, the I^2A estimation provides acceptable results for the roll axis in most cases. As previously mentioned, examining the feasibility of calculating the body rate using panoramic images is an objective of this work. Figures 5.7–5.10

illustrate the I^2A estimations with their respective errors compared with those of the XSENS. Despite showing error in Figure 5.9 and 5.10, the I^2A follows the same pattern of rate changes as the XSENS (Figures 5.7 and 5.8), and it is shown in Chapter 6 how integrating the I^2A and horizon-based estimations can improve the overall attitude estimation.

The undercarriage blocks approximately 20% of the view on the x axis of the image, where the position of the pitch angle has the most effect, and decreases the accuracy of the I^2A estimation for this axis. Apart from that, the sun's flare can also degrade the image quality. A real time I^2A estimation for the pitch angle is shown in Figure 5.8. When the sun affects the brightness of a part of an image, the I^2A interpolates this changes but the result can be incorrect due to these being an unexpected effect on the image and shows up as a rapid change in the estimation. If a specific visual effect (for example the sun's flare) remains on a particular region for more than two images, the I^2A calculates the correct displacement for the next estimation because both images have the same feature. As can be seen in Figure 5.8a, although an unexpected objects affect the estimation in the form of a rapid change, the next I^2A estimation is correct. Despite sometimes having unreal estimations, the I^2A provides mainly acceptable results by following the same pattern. Note that all tne I^2A results are raw data with no filtering which are compared with XSENS gyroscopic data that has an on–board corrections unit for offsets and vibrations.

As mentioned in the literature review, sometimes the optical flow is misconceived by insects when an unexpected visual effect enters their vision system. However, using other sensory information (from the horizon or Halteres) they are able to correct their wrong behaviour by fusing different sources of information and obtaining benefits if one source is reliable. It is demonstrated in the sensor fusion chapter how different sensory information can be fused to provide an accurate attitude measurement and overcome the shortcomings of each sensor.

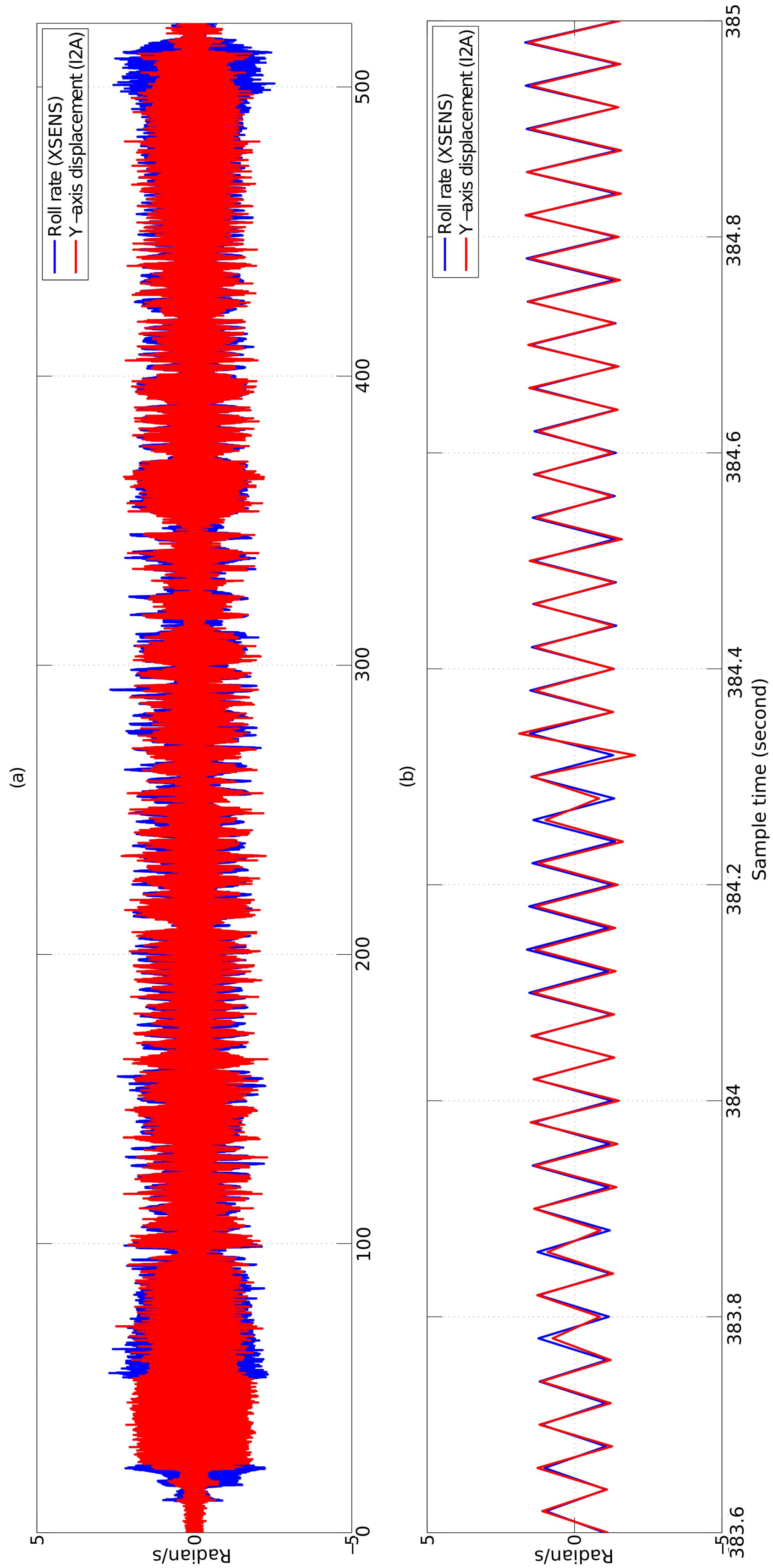


Figure 5.7: (a): Real–time roll rate estimations comparison, (b): Magnified plot

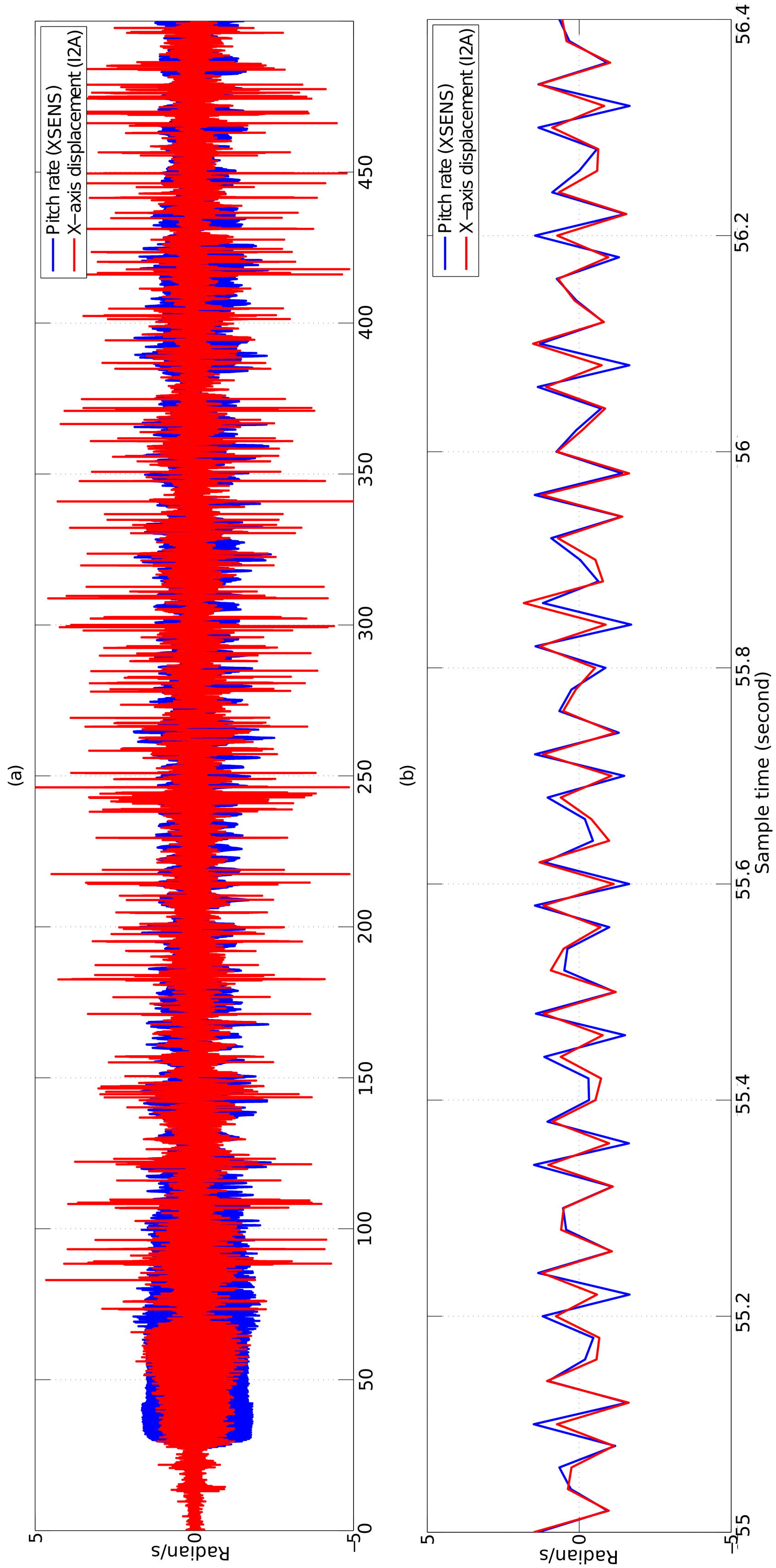


Figure 5.8: (a): Real–time pitch rate estimations comparison, (b): magnified plot

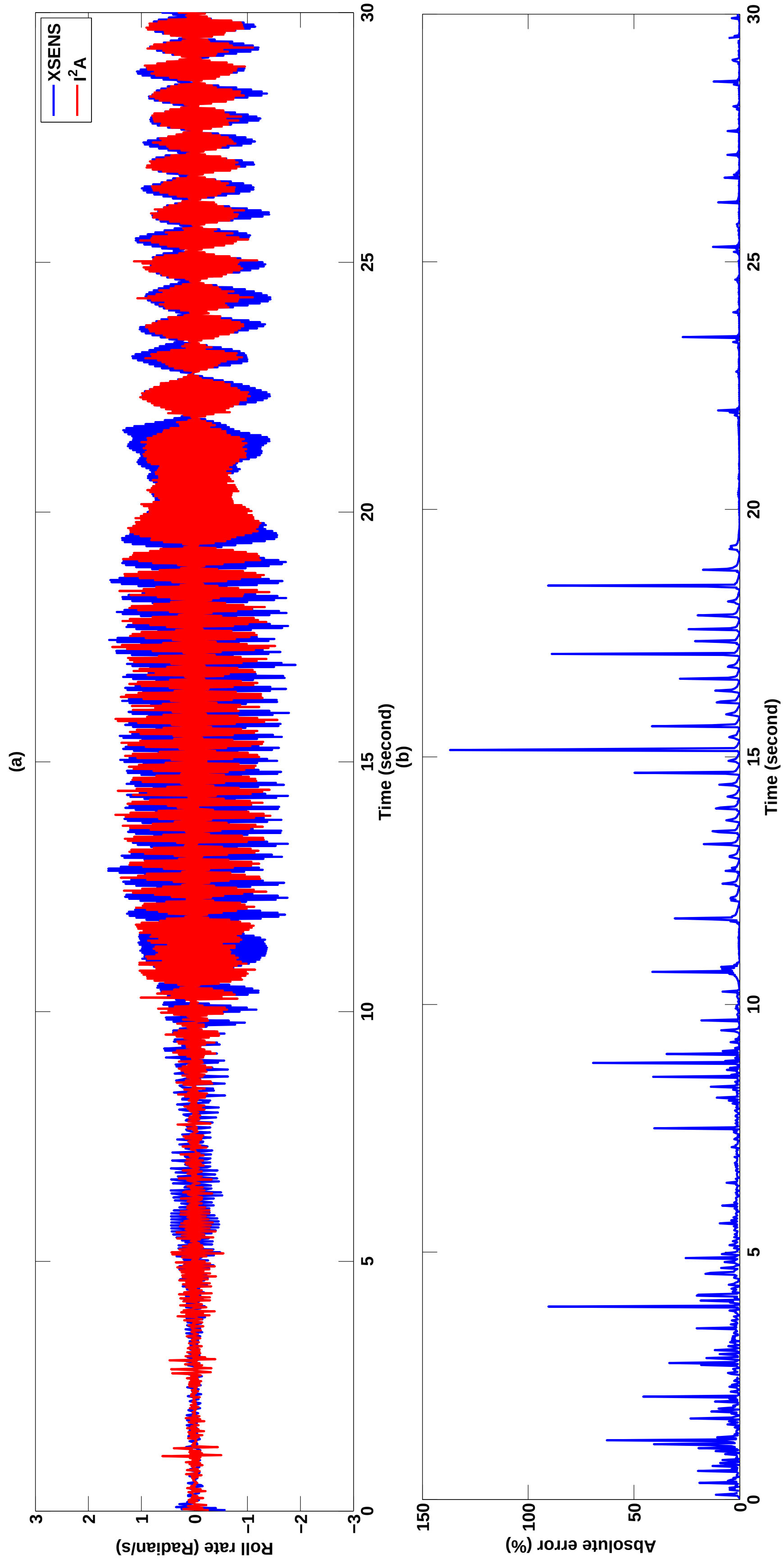


Figure 5.9: (a): Real–time roll rate estimations comparison, (b): Percentage of absolute error

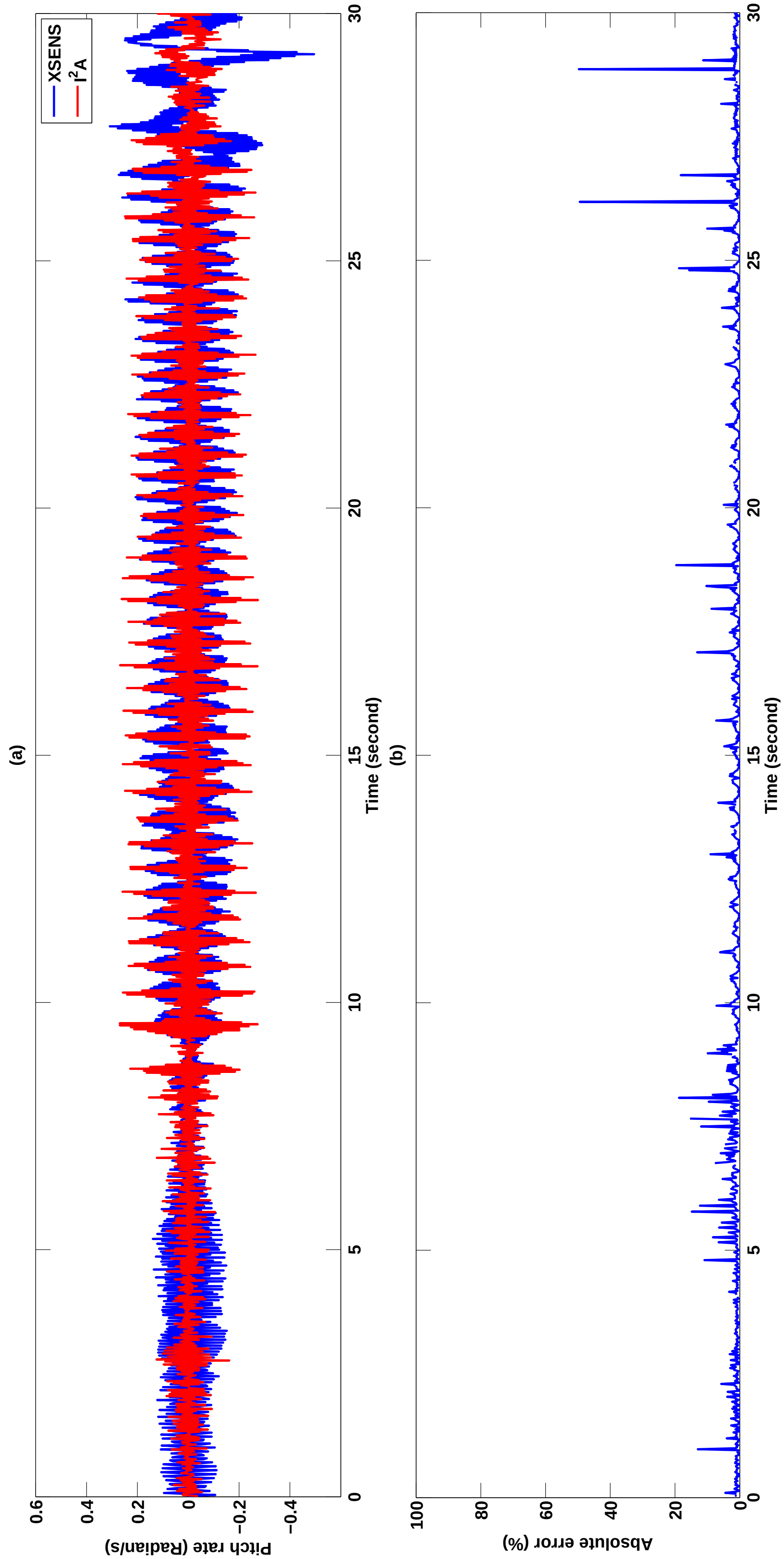


Figure 5.10: (a): Real–time pitch rate estimations comparison, (b): Percentage of absolute error

5.5 Conclusion

In this chapter, the feasibility of calculating the I^2A on UV–filtered panoramic images was tested and investigated. It was shown it could provide a good representation of the angular rate of the platform. The I^2A was chosen as the best optic flow technique for calculating the 2D motions of a panoramic image. The feasibility of its angular rate calculations was demonstrated through simulations and real flight tests, and it was shown that it could be performed on the same panoramic image used for horizon–based attitude estimation (Chapter 4. In Chapter 6, how sensory informations can be fused to improve the accuracy of attitude estimation is discussed.

Chapter 6

Sensor Fusion

6.1 Introduction

In this chapter, the effect of sensor fusion on the accuracy of attitude estimation is examined. Two sensor fusion techniques, an Extended Kalman Filter (EKF) and an Artificial Neural Network (ANN) are used independently to fuse the attitude measurements from the IMU and the vision system. The application of both techniques will be benchmarked and evaluated throughout MATLAB simulation and real–time flight test data. For this work, the angular body rates are measured (and independently used in the sensor fusion techniques) by two different sources, gyroscopes and optic flow. The source of these measurements correspond to Halteres and the compound eyes of insects. The reason behind examining the effect of these two different sources is to highlight the performance of each one on the attitude estimation. Although the positive functionality of Halteres on the insect movement has been proven [6], insects with no Halteres can also fly graceful as they have evolved to visual information. Halteres act as gyroscopes and provide information about body rotations but do not provide information about the surrounding environment. Halteres respond faster to angular velocities due to their mechanism. They also provide complimentary sensory information to the insect [6] to be integrated with the visual information from compound eyes. Insects also use the integration of Halteres sensory information with the visual information for the neck movements [76]. An insect with no Halteres combines different visual clues (sensed by different photocereptors) for flight and navigation. With this highly precise understanding of its environment, the insect can react appropriately. Insufficient visual features can dramatically reduce an insect's flight capability. The significance and richness of visual effects on insect flight control has been investigated by a number of authors. While insects rely mainly on vision, Halteres also provide complementary sensory information. Their significance to insect flight become more pronounced when there is little to no visual information is available to the insect to help it stabilise its body movement. To provide complementary information, there is also Halteres functionality on the insect's head. Removing Halteres disrupts stable head movement as it responds to visual patterns. Although Halteres have positive effect on insect flight, there are some insects with no Halteres. Research by Partha S. Bhagavatula et al [71] showed that birds also use visual information for flying.

To maintain the biological paradigm of this work, in this chapter, different sensory information is integrated by two different sensor fusion techniques (Extended kalman Filter and Artificial Neural Networks), to investigate the effect of the sensor fusion on the quality and accuracy of the attitude estimation. The effect of sensory data fusion to improve results is investigated using data from the inertial sensor and vision system. The aim is not to introduce a new method for sensor fusion, but to demonstrate how to choose appropriate sources incorporated with sensor fusion techniques to increase the proficiency of the EKF and ANN. It should be noted that different flight tests were conducted to examine their performance of EKF and ANN. The results discussed in each section refers to a particular visual condition that might be different from another one. The RMS error of an experiment depends on the quality and accuracy of the data (horizon-based attitude estimation and I2A) obtained from the vision system and no two experiments have completely identical RMS errors.

6.2 Overview of Extended Kalman Filter (EKF)

The capability of the EKF to provide an optimal estimation of a non–linear process has been investigated for decades. The Kalman Filter is a statistical–based filter for estimating a system's states, the statistical model properties and actual noise components of which are known. The EKF is a Kalman Filter that estimates a non–linear system by locally linearising it, although a non–linear system can be modelled by locally linearising it, depending on the system complexity, the linearision can be computationally expensive. The following overview of an EKF is adopted from [108] and [96], and its non–linear dynamic model of an EKF is defined by:

$$x_k = f(x_{k-1}, k-1) + w_k \tag{6.1}$$

The k_{th} estimation of x_k is calculated by a state function f from state variables and with w_k is the random process noise. The process noise matrix consists of zero–mean, white Gaussian noise that is defined as:

$$\mathbf{Q} = E(ww^T) \tag{6.2}$$

The non–linear measurement model is also defined as:

$$z_k = h(x_k, k) + v_k \tag{6.3}$$

where measurement noise v_k matrix is defined as:

$$\mathbf{R} = E(vv^T) \tag{6.4}$$

f and h are the non–linear functions of the input and the output states. w_k and v_k are mutually independent zero–mean Gaussian white noise sequence. Measurement noise matrix contains the variances of each source of measurement. In an EKF, non–linear states of a system are approximated by a linearised state equation. Discrete EKF approximation is done by recursively solving the following Equations:

$$\hat{x}_k^- = f(\hat{x}_{k-1}^+) \tag{6.5}$$

$$\hat{z}_k = h(\hat{x}_k^-, k) \tag{6.6}$$

The state matrix Φ and observation matrix h_k are linearly approximated by:

$$\Phi \approx \left.\frac{\partial f(x)}{\partial x}\right|_{x=\hat{x}_k^-} \tag{6.7}$$

$$h_k \approx \left.\frac{\partial h(x)}{\partial x}\right|_{x=\hat{x}_k^-} \tag{6.8}$$

The fundamental matrix is approximated by Taylor–series expansion (FT_s), [108], and given as:

$$\Phi = e^{FT} = \sum_{i=1}^{N} \frac{(F^n {T_s}^n)}{i!} = I + FT_s + \frac{F^2T_s^2}{2!} + \frac{F^3T_s^3}{3!} + ... \tag{6.9}$$

where N is infinity, I is identity matrix, F is the transition matrix and T_s is sampling time. Calculation of the fundamental matrix Φ is computationally expensive and hence we seek to simplify the calculation to enable real-time computation. Generally, with fast sample rates, deriving more Taylor–series expansion does not improve the overall performance (e.g. adaptation and divergence mitigation). Instead, deriving more accurate dynamic equations for the system and using more appropriate process noise provides better filtering and covers a wider bandwidth [108]. In addition, compared to the first two terms of the the Equation 6.9, the remaining terms contribute less to the total calculation when sampling time is fast (T_s is small). In an experiment, the infinite expansion of FT_s was calculated in MATLAB using matrix exponential e^{FT_s} and six states are examined in section 6.3.1, where it is shown that the effect of the higher order terms is sufficiently negligible to be ignored. Thus, in this work, the first two terms of the Taylor–series expansion are used:

$$\Phi = I + FT_s \tag{6.10}$$

The priori covariance matrix is calculated from linearised state matrix Φ_{k-1} and the process noise Q:

$$\widehat{\mathbf{P}}_k^- = \Phi_{k-1}\mathbf{P}_{k-1}\Phi_{k-1}^T + \mathbf{Q} \tag{6.11}$$

The Kalman gain is calculated from the priory covariance $\widehat{P}_k^-$ and the measurement noise R:

$$K_k = \mathbf{P}_k^- H_k^T [H_k \mathbf{P}_k^- H_k^T + \mathbf{R}]^{-1} \tag{6.12}$$

The update stage is performed by calculating a new Kalman gain, new states' estimations and the posteriori covariance matrix. The posteriori covariance is defined by:

$$\widehat{\mathbf{P}}_k^+ = (I - K_k H_k)\mathbf{P}_k^- \tag{6.13}$$

The new states $\hat{x}_k^+$ are updated by:

$$\hat{x}_k^+ = \hat{x}_k^- + K_k(z_k - \hat{z}_k) \tag{6.14}$$

where $\hat{x}_k^-$ is the estimation of previous time $k-1$, z_k is the observation and $\hat{z}_k$ is the prediction after the states' propagation. Diagonal matrices, **Q** and **R**, represent the standard variance of the noise.

The Kalman filter can estimate the states of a linear system with Gaussian noise distribution when the system is dynamically modelled. But optimal estimation can not be achieved when the system has non–Gaussian characteristics or is not precisely modelled.

6.3 Attitude Estimation using Extended Kalman Filter

The goal is to estimate the helicopter's (Eagle) attitude, particularly its roll and pitch rotations. Gyroscopes suffer drifts over time which make the attitude to deviate (to an inaccurate level). One of the main causes of gyro drifts is thermal effects. The drift shows up as an offset value which is added to the actual gyroscopic measurement. As the temperature changes, the offset will slowly change with time. In addition,apart from a fixed offset, the attitude drifts over time due to the signal integration inherent in the attitude update equations. While an advanced IMU typically uses circuitry or firmware to compensate thermal drift and generally has a filter such as an EKF to estimate the offset from observations, the offset is still a major problem for low–end IMUs (cheap IMUs) [3]. An EKF is implemented in the current work to act as an attitude estimator in the presence of noisy data and offsets in measured rotation rate. It is designed to estimate nine states, including two unit vectors (gravity and north directions) and three gyroscopic offsets. Although a similar EKF was designed by Garratt [96] using gyroscopes, inertial sensors and magnetometers, that in this work differs in the way of it uses different sensory information. The performance of the EKF is investigated using the following sensory sources, including:

1. Gyroscopes and inertial data,
2. Gyroscopes and vision attitude,
3. Optic flow and vision attitude.

The state vector consists of nine states and is defined as:

$$x = [g_x \ g_y \ g_z \ b_x \ b_y \ b_z \ \delta_x \ \delta_y \ \delta_z]^T \tag{6.15}$$

where the gravity direction vector is $g = [g_x \ g_y \ g_z]^T$, magnetic direction vector $b = [b_x \ b_y \ b_z]^T$ and $\delta = [\delta_x \ \delta_y \ \delta_z]^T$ is gyroscope offset vector. Quaternion–based attitude representation is chosen in this work and its related equations derived from [17]. The vectors are dependent on the body axis rotations which are measured by gyroscopes. An actual gyroscopic measurement which is a gyroscope's output measurement with its offset is:

$$P = p + \delta_p, Q = q + \delta_q, R = r + \delta_r \tag{6.16}$$

where P, Q and R are the actual gyroscopic measurement and δ_p, δ_q and δ_r are the offsets. Body axis rotations are subject to a body rate of change over a time step ΔT. The effect of rotational rates on the attitude can be determined by quaternion–based rotation matrix:

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = -\frac{1}{2} \begin{bmatrix} 0 & P & Q & R \\ -P & 0 & -R & Q \\ -Q & R & 0 & -P \\ -R & -Q & P & 0 \end{bmatrix}^T \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{6.17}$$

and in a compact form is:

$$\frac{dq}{dt} = -\frac{1}{2}\Omega q \tag{6.18}$$

where q represents quaternion parameters and Ω is angular velocity tensor. For a small time step ΔT, q can be approximated with the first order expression as:

$$\Delta q \approx -\frac{1}{2}\Omega q \Delta T \tag{6.19}$$

The attitude change over the time step ΔT relative to the existing body axes can be derived from the Equation 6.19 by setting the right hand side of equation to $q = [1\ 0\ 0\ 0]^T$ as:

$$\begin{bmatrix} \Delta q_0 \\ \Delta q_1 \\ \Delta q_2 \\ \Delta q_3 \end{bmatrix} = -\frac{1}{2} \begin{bmatrix} 0 \\ P \\ Q \\ R \end{bmatrix} \Delta T \tag{6.20}$$

The attitude related to the body axes over ΔT is:

$$\begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} = \begin{bmatrix} 1 \\ P\Delta T/2 \\ Q\Delta T/2 \\ R\Delta T/2 \end{bmatrix} = \begin{bmatrix} 1 \\ \bar{P} \\ \bar{Q} \\ \bar{R} \end{bmatrix} \tag{6.21}$$

The effect of changes on the gravity vector (g) and the magnetic vector (b) is determined by B rotation matrix. The EKF state matrix is for nine states is determined as:

$$\begin{bmatrix} g_x \\ g_y \\ g_z \\ b_x \\ b_y \\ b_z \\ \delta_p \\ \delta_q \\ \delta r \end{bmatrix}_{K+1} = \begin{bmatrix} B_{11} & B_{12} & B_{13} & 0 & 0 & 0 & 0 & 0 & 0 \\ B_{21} & B_{22} & B_{23} & 0 & 0 & 0 & 0 & 0 & 0 \\ B_{31} & B_{32} & B_{33} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & B_{11} & B_{12} & B_{13} & 0 & 0 & 0 \\ 0 & 0 & 0 & B_{21} & B_{22} & B_{23} & 0 & 0 & 0 \\ 0 & 0 & 0 & B_{31} & B_{32} & B_{33} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} g_x \\ g_y \\ g_z \\ b_x \\ b_y \\ b_z \\ \delta_p \\ \delta_q \\ \delta r \end{bmatrix}_{K} \tag{6.22}$$

The rotation matrix B is defined in quaternion as:

$$B = \begin{bmatrix} q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_2 + q_0q_3) \\ 2(q_1q_2 - q_0q_3) & q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_2q_3 + q_0q_1) \\ 2(q_1q_3 + q_1q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 \end{bmatrix} \tag{6.23}$$

Substituting from the Equation 6.21 results:

$$B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} \bar{P}^2 - \bar{Q}^2 - \bar{R}^2 & 2(\bar{P}\bar{Q} + \bar{R}) & 2(\bar{P}\bar{R} + \bar{Q}) \\ 2(\bar{P}\bar{Q} + \bar{R}) & -\bar{P}^2 + \bar{Q}^2 - \bar{R}^2 & 2(\bar{Q}\bar{R} + \bar{P}) \\ 2(\bar{P}\bar{R} + \bar{Q}) & 2(\bar{Q}\bar{R} - \bar{P}) & -\bar{P}^2 - \bar{Q}^2 + \bar{R}^2 \end{bmatrix} \tag{6.24}$$

All state updates can be derived from the Equations 6.16, 6.21, 6.22 and 6.24. For an example, the first element of EKF state update (g_x^{k+1} in the Equation 6.22) for the components of the gravity vector in x–axis is derived from the Equations 6.22 and 6.24 as:

$$\begin{aligned} g_x^{k+1} = g_x^k &+ \frac{\Delta T^2}{4}((p + \delta p)^2 - (q + \delta q)^2 - (r + \delta r)^2))g_x^k \\ &+ \frac{\Delta T^2}{2}((p + \delta p)(q + \delta q) + \Delta T(r + \delta r)^2))g_y^k \\ &+ \frac{\Delta T^2}{2}((p + \delta p)(r + \delta r) + (q + \delta q)))g_z^k \end{aligned} \tag{6.25}$$

ΔT is 20ms which is relative to the time intervals in which the data is sampled from the sensors. Equation 6.25 is differentiated partially for each state variable to derive the matrix Φ. Equation 6.25 described the first row of the Φ matrix is derived as:

$$\frac{\partial g_x^{k+1}}{\partial g_x^k} = 1 + \frac{\Delta T^2}{4}((p + \delta p)^2 - (q + \delta q)^2 - (r + \delta r)^2) \tag{6.26}$$

$$\frac{\partial g_x^{k+1}}{\partial g_y^k} = \frac{\Delta T^2}{2}(\frac{\Delta T^2}{2}(p + \delta p)(q + \delta q) + \Delta T(r + \delta r)) \tag{6.27}$$

$$\frac{\partial g_x^{k+1}}{\partial g_z^k} = \frac{\Delta T^2}{2}(\frac{\Delta T^2}{2}(p + \delta p)(q + \delta q) - \Delta T(r + \delta r)) \tag{6.28}$$

$$\frac{\partial g_x^{k+1}}{\partial \delta_p^k} = \frac{\Delta T^2}{2}(p+\delta p)g_x^k + \frac{\Delta T^2}{2}(q+\delta q)g_y^k + \frac{\Delta T^2}{2}(r+\delta r)g_z^k \tag{6.29}$$

$$\frac{\partial g_x^{k+1}}{\partial \delta_p^k} = -\frac{\Delta T^2}{2}(q+\delta q)g_x^k + \frac{\Delta T^2}{2}(p+\delta p)g_y^k - \Delta T g_z^k \tag{6.30}$$

$$\frac{\partial g_x^{k+1}}{\partial \delta_p^k} = -\frac{\Delta T^2}{2}(r+\delta r)g_x^k + \Delta T g_y^k + \frac{\Delta T^2}{2}(p+\delta p)g_z^k \tag{6.31}$$

The observations are two normalised vectors (each consisting of three components for three axes). The measurement matrix is an identity matrix and time invariant as:

$$H = I_6 \tag{6.32}$$

The north direction vector comes from the magnetometers while the gravity vector is defined from two different sensory sources, the accelerometers and vision system, to investigate the EKF's performance. The EKF is used to obtain better attitude estimations by fusing different measurements and not only estimates the attitude from noisy measurements but also filters out transient conditions due to rapid platform movements. The roll and pitch angles can be calculated using Equation 6.33 from the EKF–estimated gravity vector $g_{EKF} = [g_x \; g_y \; g_z]$ as:

$$\phi = \tan^{-1}\frac{g_y}{\sqrt{g_x^2+g_z^2}}, \qquad \theta = \tan^{-1}\frac{g_x}{\sqrt{g_y^2+g_z^2}} \tag{6.33}$$

6.3.1 EKF Attitude Estimation using Inertial data (Off–Line processing)

The EKF measurement and process noise selection depends on complete prior knowledge of the system's properties (which is difficult to achieve in practice). One of the system's parameters is the standard deviation over time, the value of which is written in the device manual but is an estimate for a particular condition when an experiment is conducted. Although these factors can be modified slightly when a condition such as the temperature changed, generally they are chosen experimentally to enable the EKF to handle noise with the maximum likelihood. In practice, it is difficult to define noise parameters that is applicable for all conditions. In this work, it is assumed that the system inputs have uncorrelated zero–mean Gaussian white noise parameters (measurement and process noise). For a particular EKF application in this work using XSENS, the measurement noise was calculated from the standard deviation of the XSENS measurements vector. The XSENS was running on a stable platform for five minutes towhich is assumed to be sufficient to reach its stable condition with the standard deviation and variance were calculated in MALTAB using the following Equations:

$$std = \left(\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2 \right)^{\frac{1}{2}} \tag{6.34}$$

$$Var = (std)^2 \tag{6.35}$$

where std is the standard deviation of x with the mean value of $\bar{x}$, and the Var is the variance. Although attitude changes are modelled by the EKF equations, but no uncertainties (except the white noise) are included in the estimation. The value of the process noise is chosen to be a larger than the measurement noise due to imperfections in the EKF equation (e.g. propagation equation). Real IMU data with 130 second duration was logged from the PC104 for testing the EKF performance in MATLAB. Constant offsets of $\delta p = -3°/\text{sec}$ and $\delta q = 3°/\text{sec}$ were added to the gyro rates. The IMU was rotating up to the maximum expected attitude range $(-/+45°)$. The performance of EKF is shown in Figure 6.1.

It takes approximately 10 seconds for the EKF parameters to settle down and the attitude to converge. Some of the convergence time is due to the time that the EKF takes to correct the gyroscope offset. With smooth movements, the EKF estimates the attitude accurately with RMS error (Equation 6.36) of 0.19 and 0.23 degree for roll and pitch respectively in off–line processing and the difference in the estimated and the actual attitude in both Figure 6.1a and 6.1b can hardly be seen. The XSENS is to

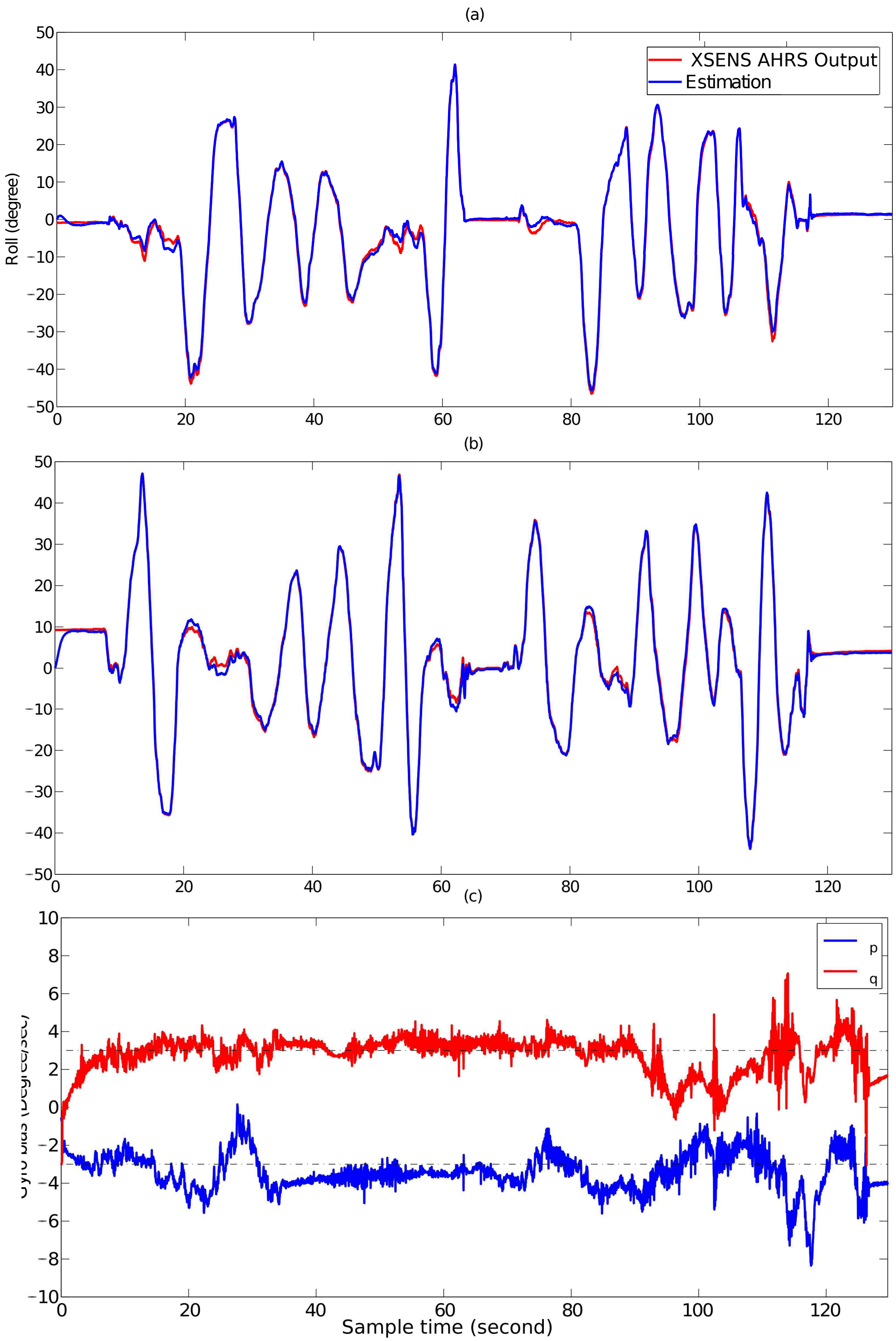


Figure 6.1: EKF results using IMU inertial data

be equipped with different on–board processing units for compensation. As the actual gyro offset is unnoticeable, adding a value to the gyro readings can ensure good EKF performance. Although the EKF calculates the gyro offset, its main application is for estimating the gravity and north vectors to be used for attitude calculations (Figure 6.1).

To examine the effect of infinite expansion on the calculation of the fundamental matrix, the elements of the fundamental matrix was calculated using e^{FT_s} where F is the transition matrix and T_s is sampling time. The performance of the EKF in estimating the six main states (x=[g_x g_y g_z b_x b_y b_z]) and the attitude are examined with respective RMS errors calculated using the XSENS AHRS output as a benchmark. The results are shown in Figures 6.2 and 6.3. As can be seen from Figure 6.2 and respective RMS errors, using infinite expansion reduced the RMS error of the states but has limited effect on the attitude estimation improvement (see Figure 6.3). As mentioned earlier in the chapter, the value of process noise and the correct choice of system equations have the dominant effect on the EKF performance in terms of the final estimation and divergence prevention. With respect to the computation time, the calculation of e^{FT_s} (Matrix exponential) requires longer computation time than the two other terms of the Taylor–series expansion. Therefore, the first two terms of the Taylor–series expansion (Equation 6.10) is used in the real–time implementation of this work.

6.3.2 EKF Attitude Estimation using Inertial data (Real–Time processing)

The EKF was implemented in C code (by Garratt [96]) for execution on the PC104. The EKF is unable to provide good state estimation when the system contains additive uncertainties which were not included in the defined system model or the noise characteristics. In these situations, the covariance matrix (Equation 6.11 and 6.13) can not cope with the situation and produce an incorrect number or diverges to a big number, that causes the EKF to fail. In a real flight, the Eagle's inertial measurement

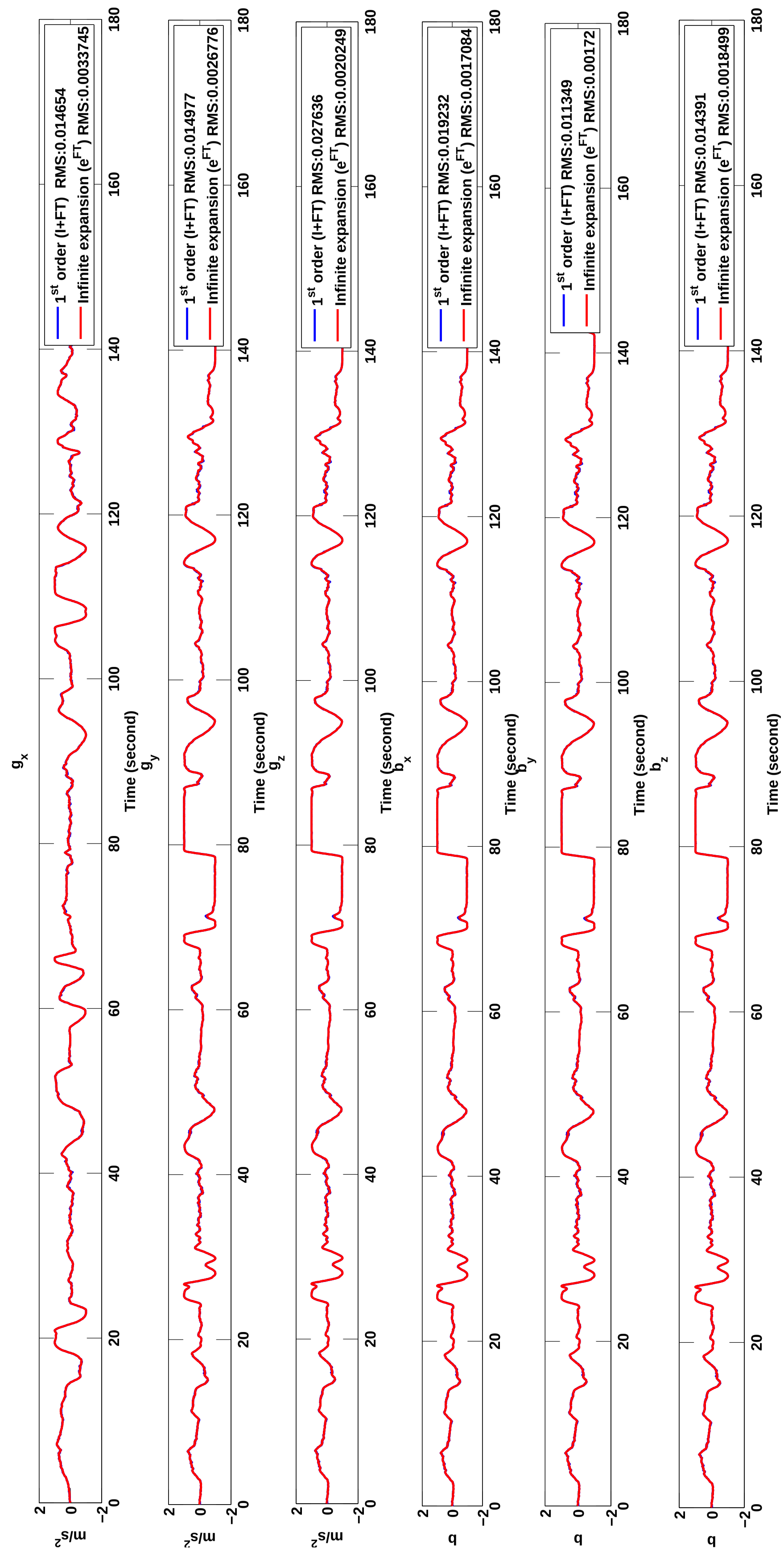


Figure 6.2: RMS errors of EKF states using the first Taylor–series expansion and the infinite expansion.

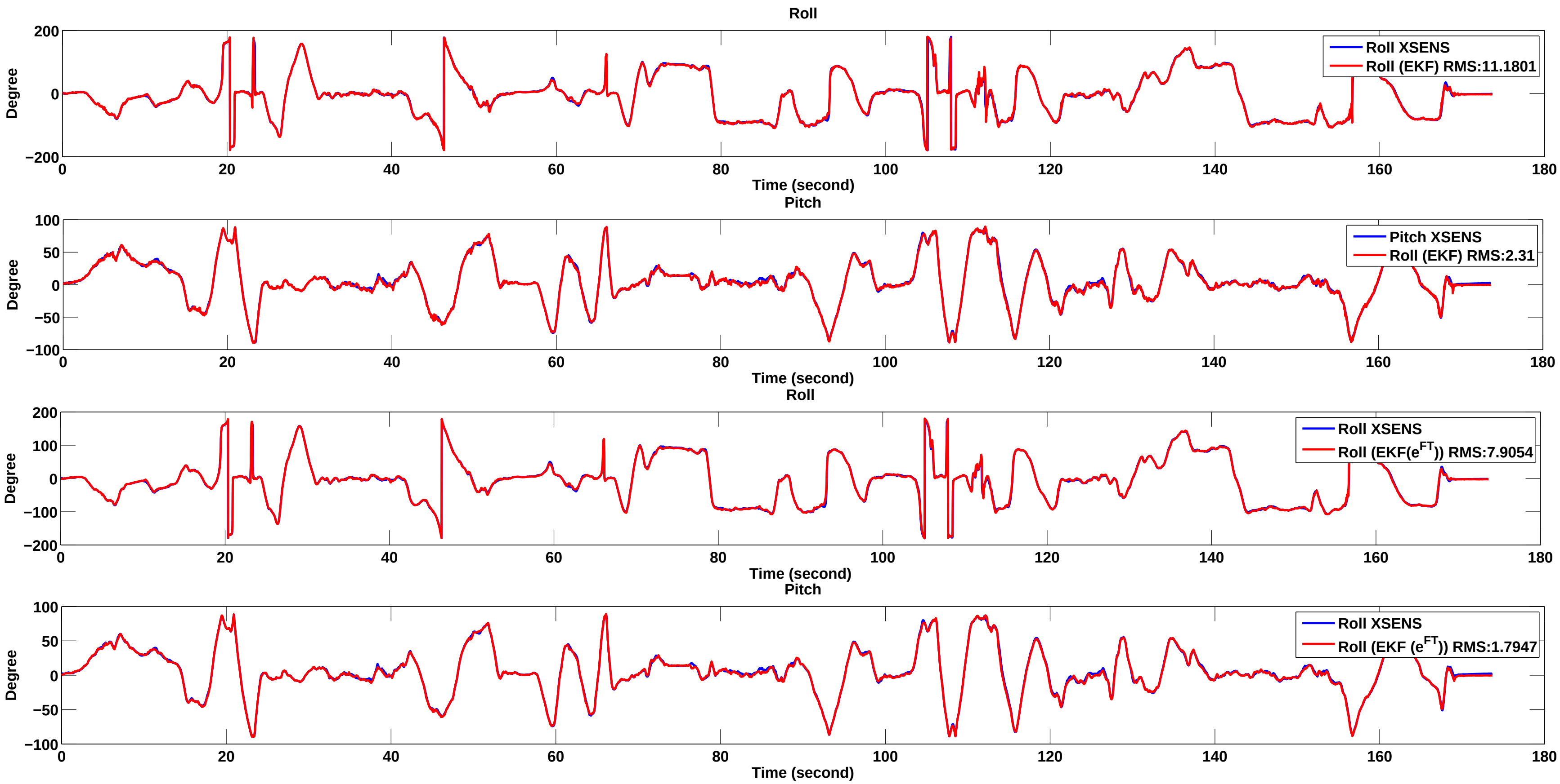


Figure 6.3: Attitude estimation using the first Taylor–series expansion and the infinite expansion.

is mixed up with the excessive amounts of vibration, making the EKF fail to estimate an acceptable result. Mainly the vibration perturbs the measurement of accelerometers and the gyroscopes. To tackle this, the inertial measurement should be highly filtered by a low pass filter to remove the high frequencies. However this may impose a big lag in measurement and may filter some of the desired inertial data. Figure 6.4 shows the components of gravity direction from the XSENS accelerometers and the vision. It is obvious that during smooth movement (Figure 6.4a) the measurements have no vibrational disturbance but during the flight (Figure 6.4b), they are affected by massive vibrations from the Eagle's body.

The system characteristics such as the EKF measurement noise is changed due to the vibration effects. A test was done to calculate the variance of measurement noise when the Eagle faces maximum vibration. The Eagle was settled on the ground with blades spinning at maximum speed (1600 RPM). The new values of measurement and process noise were added to the EKF. During all flights in this chapter, the pilot was trying to conduct a combination of gently hovering and vigorous manoeuvring. During the flight, normalised components of the magnetic field and accelerations were used to correct the EKF estimations (EKF is supposed to estimate the gravity and north vector). The attitude result is shown in Figure 6.5.

Figure 6.5a shows that the accelerations in three axes are totally disturbed by the vibration. Figure 6.5b and 6.5d show the actual attitude from the XSENS and the estimated one from the EKF. For a better investigation, Figure 6.5c and 6.5e are shown that have a magnified part of the attitudes. It can be seen that the EKF was following the attitude but affected by the vibration with high signal fluctuation. Due to signal corruption, the EKF performance dramatically dropped down and was not reliable in the presence of high amounts of vibration. Although the EKF noise matrices were changed to higher level, the EKF still failed to estimate the vectors due to the completely corrupted signal.

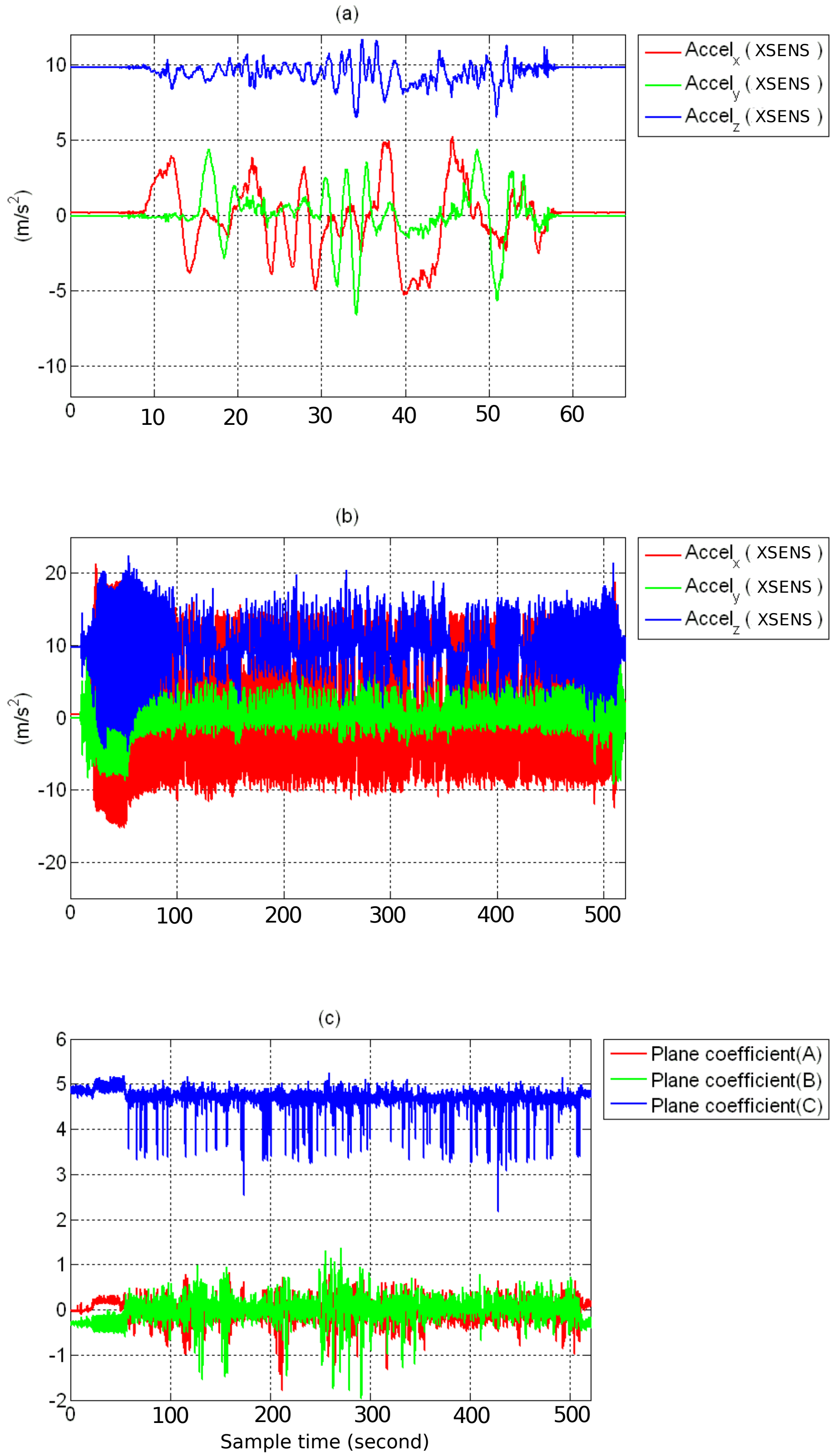


Figure 6.4: (a): Gravity components during stable movement (XSENS),(b): Gravity components during flight (XSENS), (c): Components of gravity direction during flight (Vision) ((Figure 6.4b)).

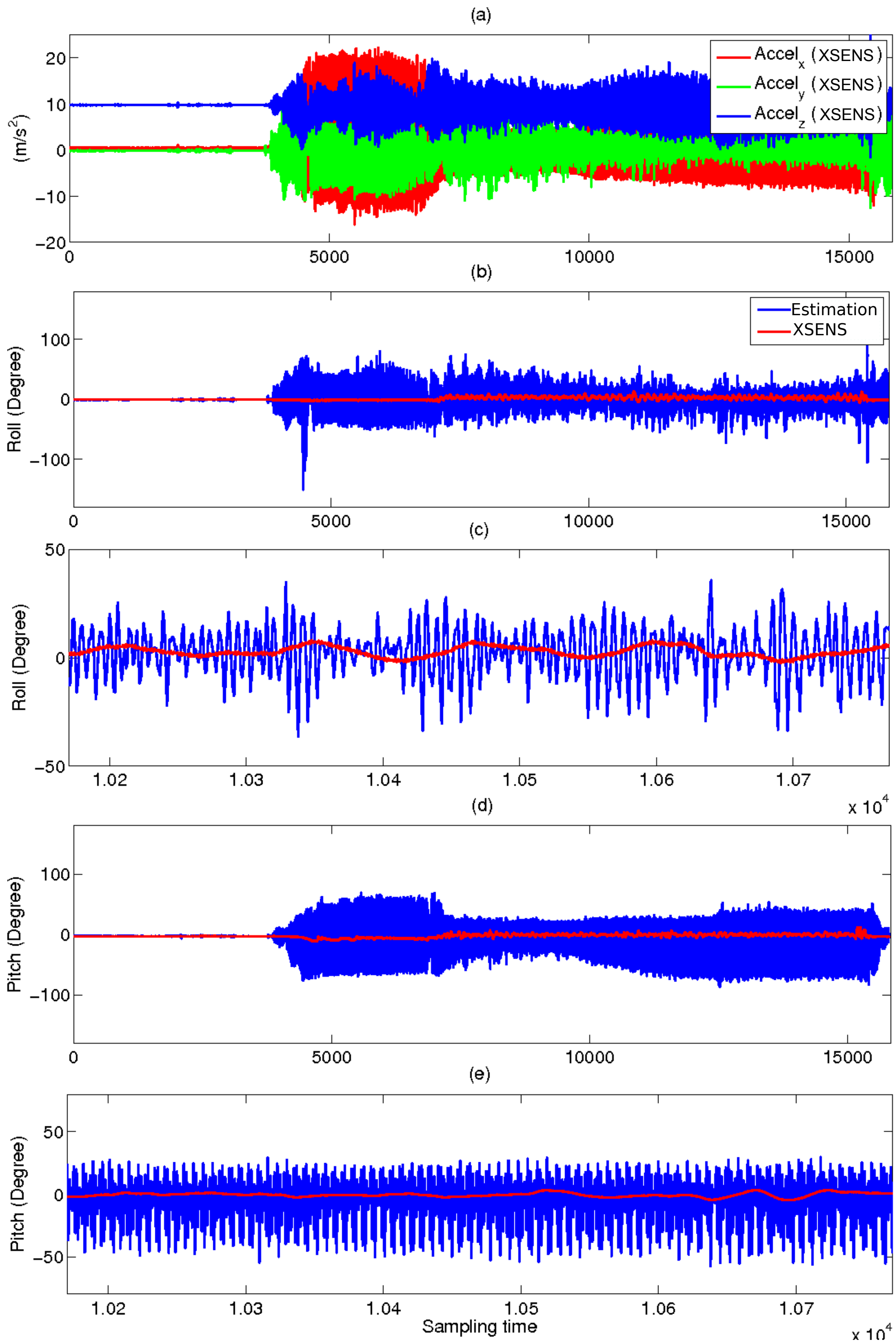


Figure 6.5: EKF attitude estimation using inertial measurement

6.3.3 EKF Attitude Estimation using Gyroscopes and Vision Attitude

In Chapter 4, how vision is beneficial for providing a more accurate measurement of the vertical direction than inertial sensors, in the presence of vibration in the system dynamics is discussed. In section 6.3.2, it was seen that how vibration (which exists in the Eagle platform) corrupted the inertial data, leading the EKF to fail to estimate the gravity direction. On the other hand, ehe panoramic vision system demonstrated its competency in defining the gravity direction even on the vibrating platform. This advantageous feature can enhance the EKF performance by substituting the inertial gravity vector with the vision–based gravity vector that has less vibration interference in it.

Another 192 second flight test was conducted during a fully cloudy day and 9,600 samples were logged (at a sample rate of 20ms). The pilot was trying to conduct a vigorous manoeuvre so the Eagle has rotations up to ±10 and ±5 degrees in roll and pitch. All vision–based estimations are presented in raw (no filtering was done on them) and are compared with the processed data of XSENS. The attitude was propagated using XSENS gyro rates and the estimation received the correction from the vision (showing the gravity direction) and the magnetic field. The results are shown in Figure 6.6. As can be seen, the Eagle was reaching high rotations in roll and pitch. The attitude from the XSENS and EKF estimations are matched up with each other and are in a good agreement. The Root Mean Squared errors (RMS) was calculated for the flight duration(not before taking off) using the following Equation.

$$RMS = \sqrt{\frac{\sum_{i=1}^{i=n} (S_{1i} - S_{2i})^2}{n}} \tag{6.36}$$

Where the S_1 is the XSENS estimate and S_2 is the EKF estimate of attitude. The RMS error of the attitude before using the EKF (attitude from the horizon) were 2.54° and

1.91° for the roll and the pitch respectively but after applying the EKF, they reduced to 1.18° and 0.48°. The attitude estimation was improved by EKF for 53.4% and 74.8% for roll and pitch.

It was shown in Figure 6.6 that the vision system can define the gravity direction to improve the performance of EKF when the inertial sensor is unable to provide reliable inertial measurement (on a vibrating platform).

6.3.4 EKF Attitude Estimation using Optic Flow and Vision Attitude

One of the main objectives of this research is to introduce a new vision system capable of estimating the attitude with no aid from an IMU. During the previous tests, the EKF used the gyro rates from the IMU for attitude propagation. In the new EKF test, gyro rates were replaced with the optic flow. Although the magnetometers were used in the EKF, they have no contribution in roll and pitch calculations and these angles are calculated from the first three states which are relative to the gravity direction. Another flight test was done during a cloudy day with 170 seconds duration and 8,500 samples were logged (with a 20ms sample rate). The results are shown in Figure 6.7. As can be seen, the Eagle was reaching rotations in roll and pitch of up to -/+15 degrees. It can be seen from the Figure 6.7 that the EKF was able to work independently from the IMU and able to estimate the attitude just using the vision–based data. The RMS errors was calculated for the flight duration(not before taking off). The RMS errors of the attitude before using the EKF (attitude from the horizon) were 1.77° and 1.18° for the roll and the pitch respectively but after applying the EKF, they reduced to 1.1° and 0.83°. The attitude estimation was improved by EKF for 37.8% and 29.6% for roll and pitch.

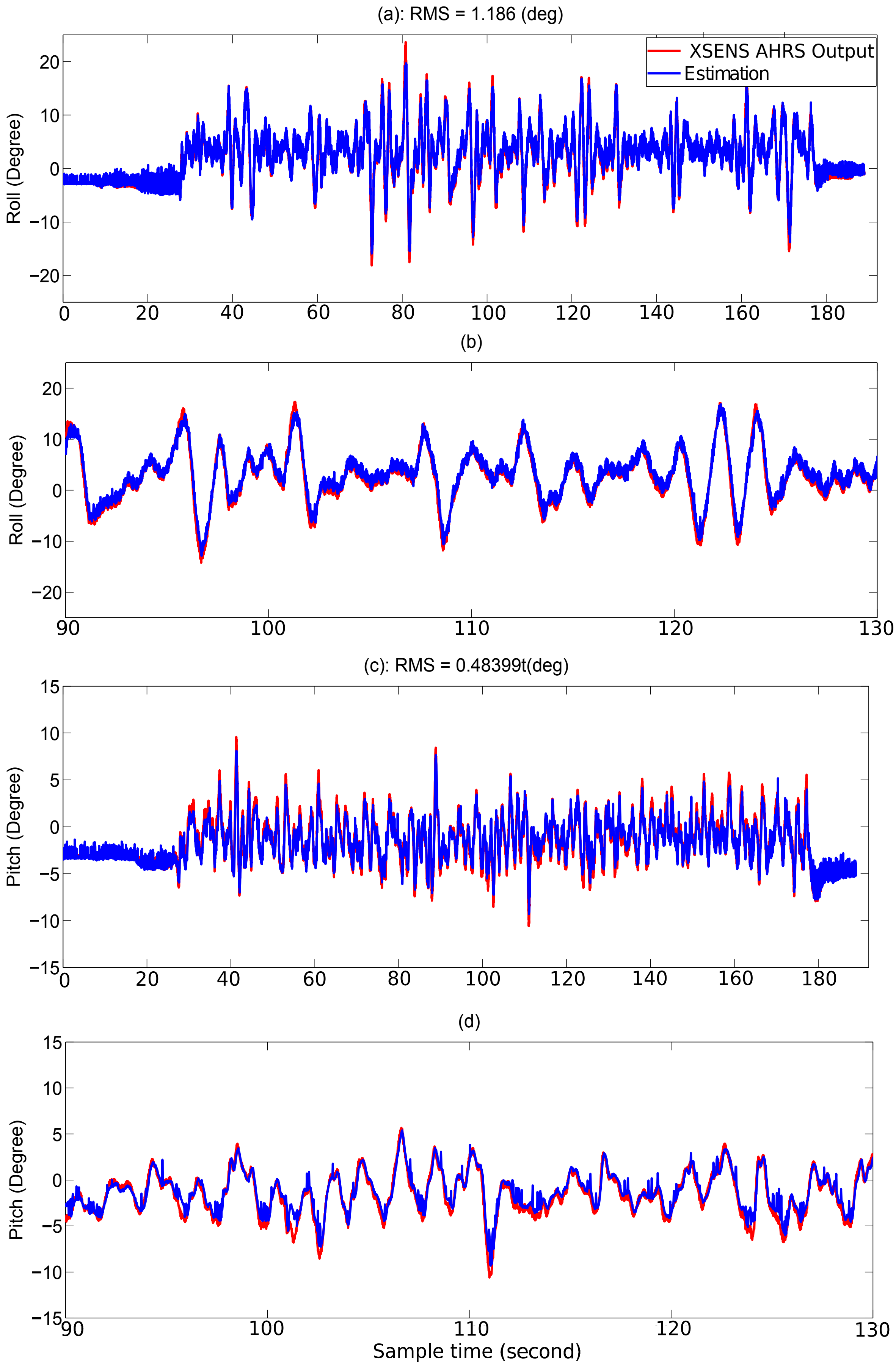


Figure 6.6: EKF attitude estimation using vision

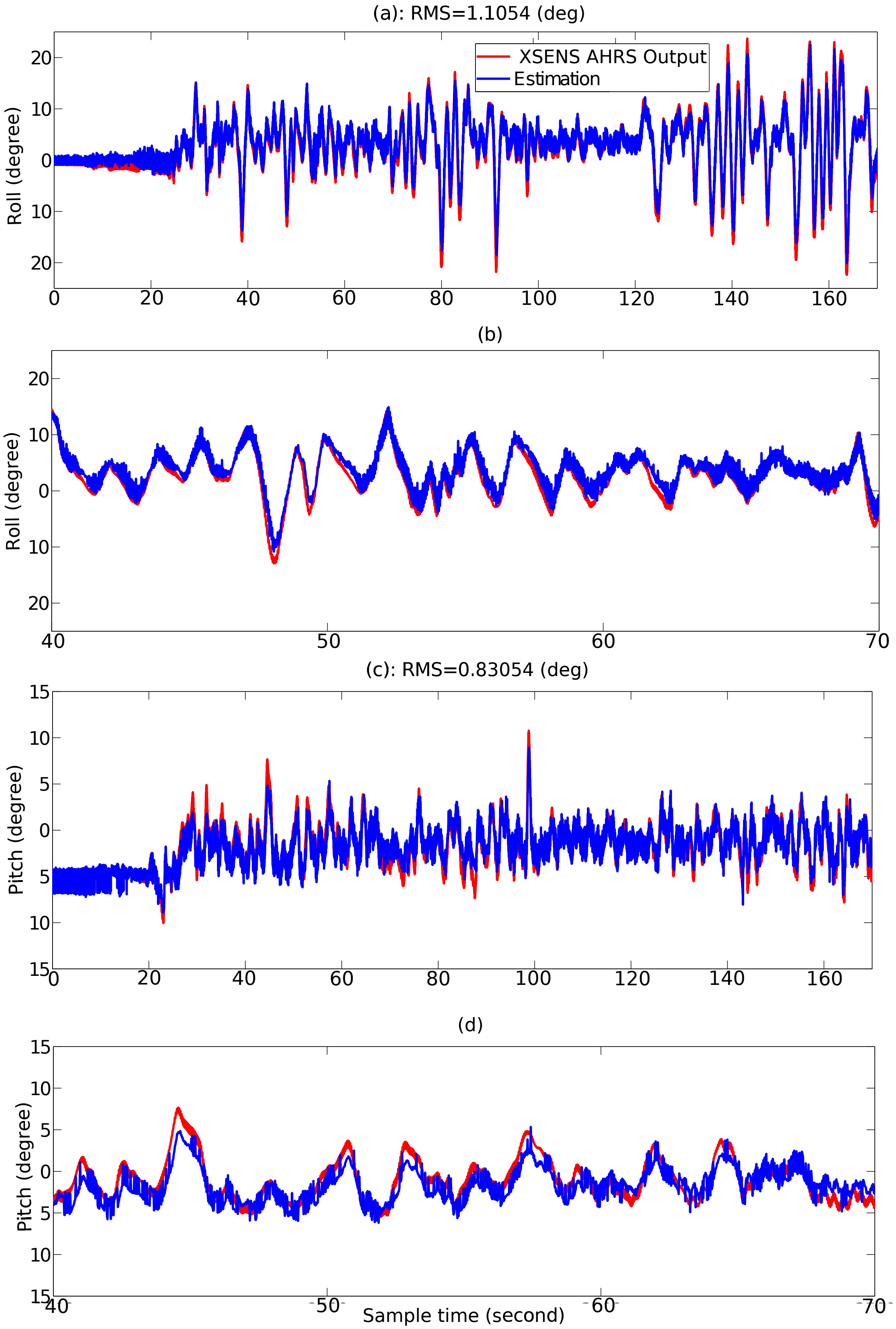


Figure 6.7: EKF attitude estimation using optic flow and vision

6.4 Artificial Neural Networks

To follow up the biological inspiration, another method of sensor fusion technique was implemented using an artificial neural networks (ANNs). Insects have relatively small brains compared to human brain. Recent research by Azevedo et al [66] estimated a human brain (aged 50 to 70) has 170.68±13.86 billion cells. Insects have a small brain with less than a million neurons, for example a honeybee brain has 960,000 neurons [65]. Although the insect brain is small, the insect is capable of achieving sophisticated flight control using their vision system (such as path finding [129] and navigation using UV–Green spectrum [62]). The insect's capabilities comes from its brain that has been trained during evolutionary life to cope with what it needs. EKF models the attitude by its statistical equations and noise variances. An EKF fails when the dynamics of system is not completely included into the process model or when the system has non–Gaussian characteristics.

A Neural Network is a model–based approach that can introduce a linear solution for a non–linear system providing a set of data. The system non–linearity can be constrained by the linear structure of an ANN. When trained, an ANN has constant parameters in it structure that are tuned during the training process to provide a solution for non–linear system, with minimized estimated output error.

When an ANN is trained (tuning) with real system data, the behaviour of the system can be modelled without knowing the explicit model of the system or understanding the system dynamics. In this project, the ANN should be able to calculate the attitude by fusing the visual information (from the horizon–based attitude and the optic flow) and filters by providing more stability and smoothness. It will be shown throughout experiment that an ANN can perform better than an EKF in terms of robustness, preventing divergence and handling unforeseen noise characteristics. For this purpose, a neural network model can be trained with real attitude data from flight tests. The properties of the system are defined by the ANN's structure and its weights and biases during the training process. The ANN was designed, trained and evaluated off–line in

MATLAB, and converted in C code for the PC104. The ANN MATLAB code and also real time ANN C codes (to be executed in PC104) were written by Garratt [96]. The ANN structure can be modified by the user to define any kind of network. For off–line training and the ANN performance investigation, the ANN data was logged during real flight tests. The ANN results were benchmarked with the XSENS sensor and the RMS error of ANN and EKF was compared with the XSENS sensor.

6.5 ANN Structure and Training

Two artificial Neural Networks are designed and implemented to estimate the real time attitude (roll and pitch). Each ANN (in this work) has a globally–recurrent feedforwarded structure with a single hidden layer with 10 neurons. The ANN model was created, modified and evaluated in MATLAB. The model can be modified by the number of hidden layers, number of neurons, number of delays, the neuron's transfer function and the training algorithm. Log–sigmoidal transfer function was chosen to be used for the hidden layer neurons that is shown in Equation 6.37.

$$logsig(n) = \frac{1}{1 + e^{-n}} \tag{6.37}$$

The schematic of the ANN is shown in Figure 6.8a. The feed–forward network that is shown in Figure 6.8b has a feedback of one unit delay from the output to the input (Figure 6.8a) that causes it to become a global–recurrent ANN. The model of the neuron is also shown in Figure 6.8c, [95].

The Levenberg–Marquardt (LM) algorithm [100] was used to train all the networks in this work. The LM algorithm claims to be fastest method to train an ANN, where there is a large amount of training data [101]. The ANN was trained using real flight data. To achieve a well-trained ANN, different flights were done during different weather conditions including sunny, partly cloudy and fully cloudy days in cluttered environment. In all flights, the altitude was approximately one meter above the ground.

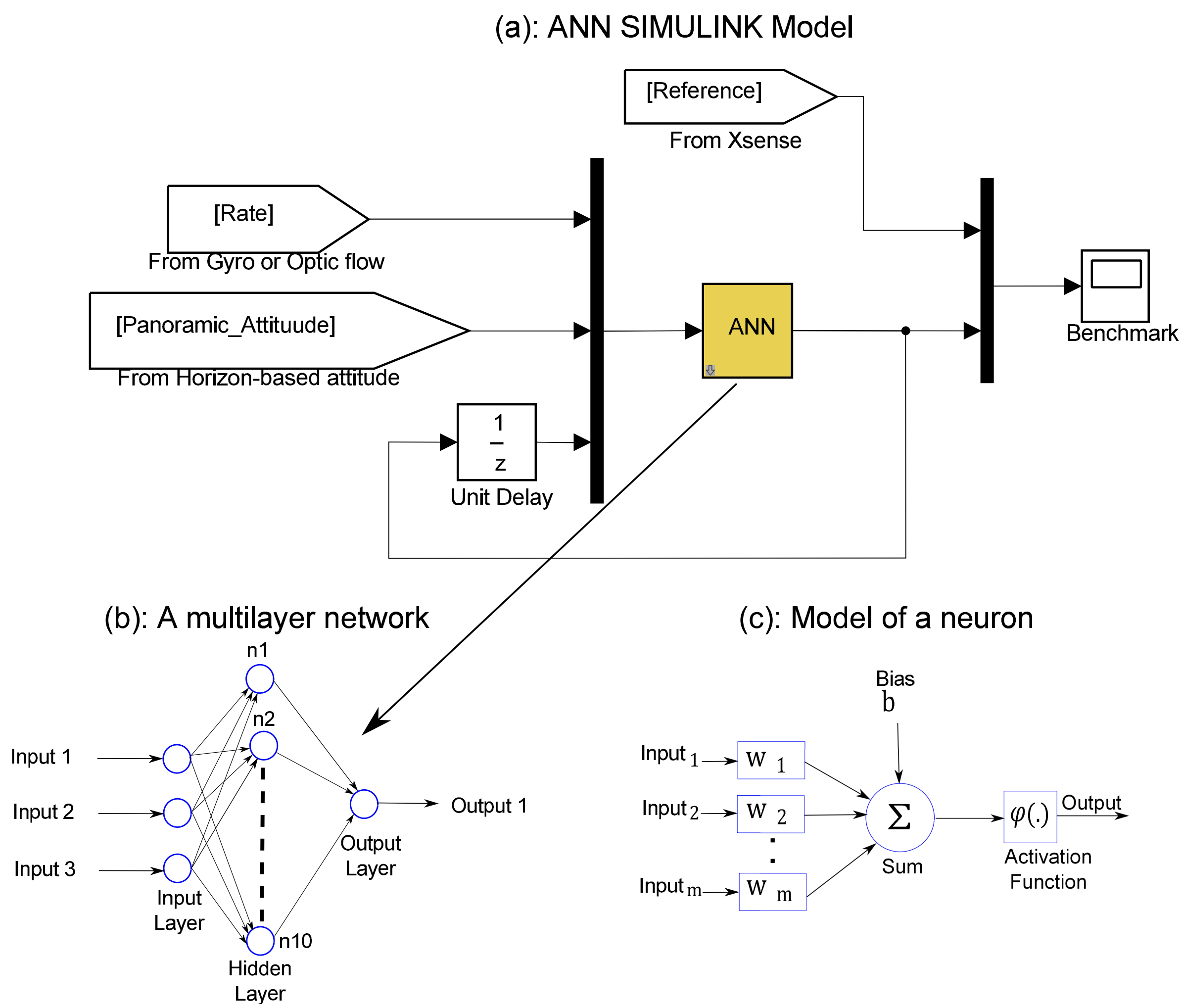


Figure 6.8: (a): ANN SIMULINK model, (b): multilayer network, (c): Model of a neuron.

Data logging was started before taking off and stopped after landing. Data sets from 830 seconds of flight (41500 samples) were concatenated to be used as the training data.

Before feeding the training data into the ANN's training process, the training data was normalised with the zero mean and unit standard deviation of 1 (Equations 2.12 and 2.13) to reduce the data dimension and speed up the training [130]. To prepare normalised data with a standard distribution, the data is translated to be centered around zero mean (calculated from Equation 2.12) and normalised using Equation 6.38. Normalising data using Equation 6.38 ensures that the data is distributed around the mean value and scattered with unity standard deviation (generally 68% of data falls within standard deviation of 1 of the mean). By normalisation, most of the data is confined between the saturation range of the activation function and makes the adaptation (between input and output) easier [94].

$$x_{i_{(normalised)}} = \frac{x_i - \mu}{std} \tag{6.38}$$

The ANN training method was supervised training in MATLAB using unfiltered flight data. During the trainings, the issues of ANN's overfitting and performance were examined and avoided. It is endeavoured to train the ANN using a dataset including all conditions, combinations and possibilities to be adaptive to different scenarios. When trained and verified in off–line, the ANN structure is transferred to the PC104 for real–time execution. During the real–time testing, the ANN worked with real–time data and the ANN results were logged and saved on a flash memory drive. Then, the ANN data was analysed and evaluated in MATLAB. Note that the EKF and ANN were independently estimating the attitude and do not influence each other's estimation. A neural net library was written by Garratt [96] in C programming language to be used in MATLAB SIMULINK and executable in the PC104.

6.5.1 ANNs Attitude Estimation using Gyroscopes and Vision Attitude

The ANNs were trained using flight data from different flight tests, containing different weather conditions and different manoeuvring to include all conditions into the training process. After the training, the ANN was tested during a 200–second flight time. The ANN roll and pitch estimations as shown in Figure 6.8a. Note that the data logging was started before take off but 10,000 samples during the flight were used for plotting. The ANN was trained to estimate the attitude using the gyroscopes and the vision attitude as input data as shown in the Figure 6.8a. The data is plotted just for the period before taking off and after landing. The ANN estimation was compared with the XSENS sensor for verification. The RMS error of the estimation before using the ANN (attitude from the horizon) were 1.2° and 1.77° for the roll and the pitch respectively. The RMS error of attitude estimation reduced to 0.51° and 0.97° for the roll and the pitch respectively after fusing the gyroscopes and vision attitude using the ANN.

The ANN and EKF performance was also compared in terms of RMS error and shown in the table 6.1. Note that the EKF worked completely with visual data (visual horizon and optic flow).

Table 6.1: Attitude estimations comparison (ANN and EKF), using gyroscopes and the visual attitude

	Panoramic Attitude (RMS)	EKF Estimation (RMS)	EKF Attitude Improvement	ANN Estimation (RMS)	ANN Attitude Estimation Improvement
Roll	1.2°	0.96°	20%	0.51°	57.5%
Pitch	1.77°	1.52°	14%	0.97°	44.8%

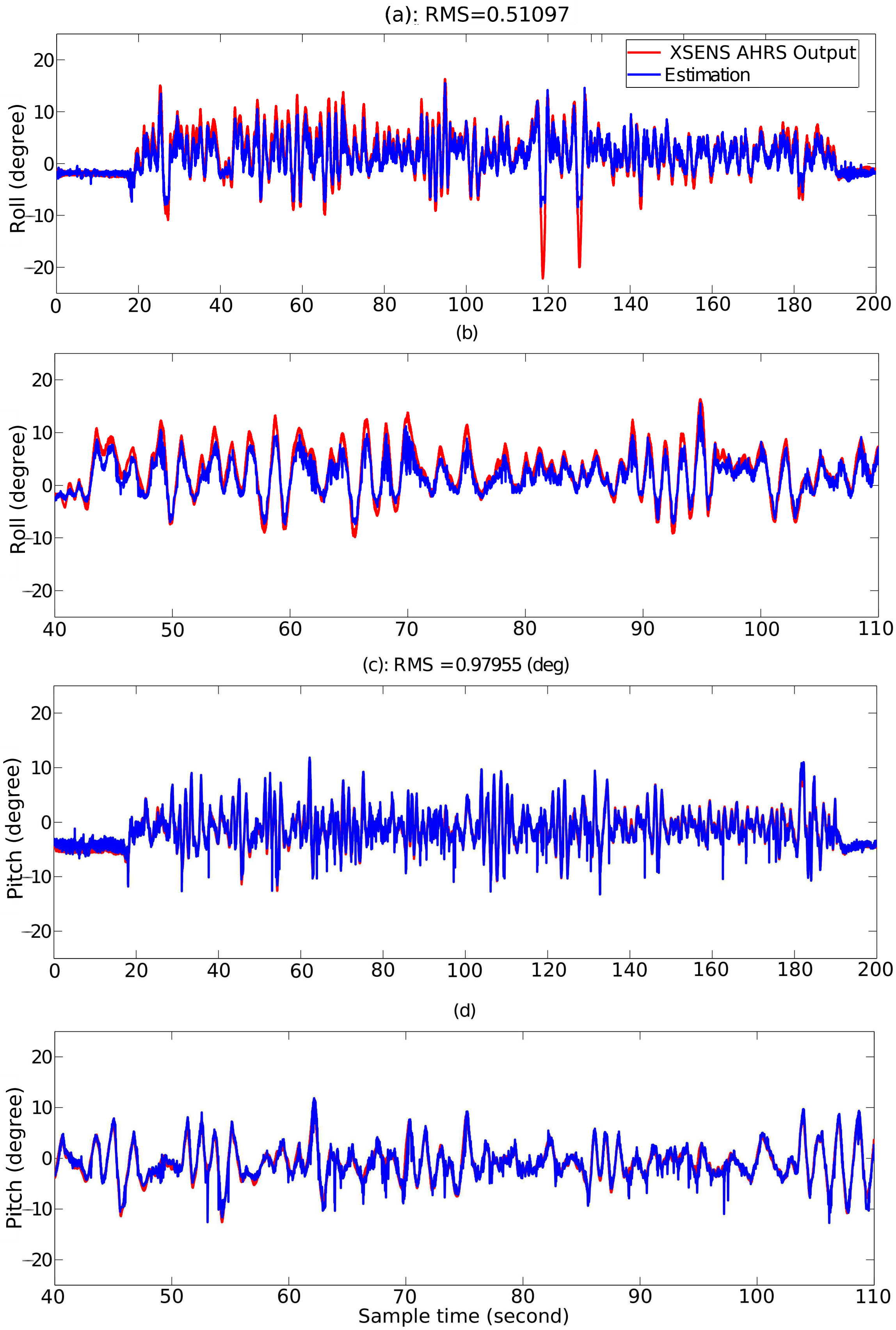


Figure 6.9: ANN attitude estimation using gyroscopes and vision

6.5.2 ANN Attitude Estimation using Optic Flow and Vision Attitude

During the 200–second flight, another ANN was loaded to be tested using optic flow and the vision attitude as its inputs as shown in Figure 6.8a. The data logging started before take off but 10,000 samples (during the flight) were used for plotting. The ANN roll and pitch estimations are shown in Figure 6.10. The ANN was trained to estimate the attitude using the optic flow and the vision attitude as the inputs. The data is plotted just for the period before taking off and after landing. The ANN estimation was compared with the XSENS sensor. The RMS error of the estimation before using the ANN (attitude from the horizon) were 1.2° and 1.77° for the roll and the pitch respectively but after using the ANN, they reduced to 0.55° and 1.05°. The attitude estimation improved by the ANN for 55% and 40.6% for roll and pitch respectively.

The ANN and EKF performance was also compared in terms of RMS error and shown in the table 6.2. Note that the EKF was working completely with visual data (visual horizon and optic flow).

Table 6.2: Different attitude estimations comparison (ANN and EKF), using optic flow and the visual attitude

	Panoramic Attitude (RMS)	EKF Estimation (RMS)	EKF Attitude Improvement	ANN Estimation (RMS)	ANN Attitude Estimation Improvement
Roll	1.2°	0.967°	20%	0.55°	55%
Pitch	1.77°	1.52°	14%	1.05°	40.6%

Although it was shown that using an EKF can improve the performance, it was revealed throughout the experiment that the ANN can enhance the panoramic attitude estimation. As already stated in the current chapter and also the thesis chapter 6, a trained neural network can represent a dynamic system with its structure and parameters and the quality of the ANN system representation is highly dependant on the training data and the applied method. As mentioned in chapter 3 (Equations 3.6 and 3.7), two calibration equations are applied to the vision–based attitude estimations to

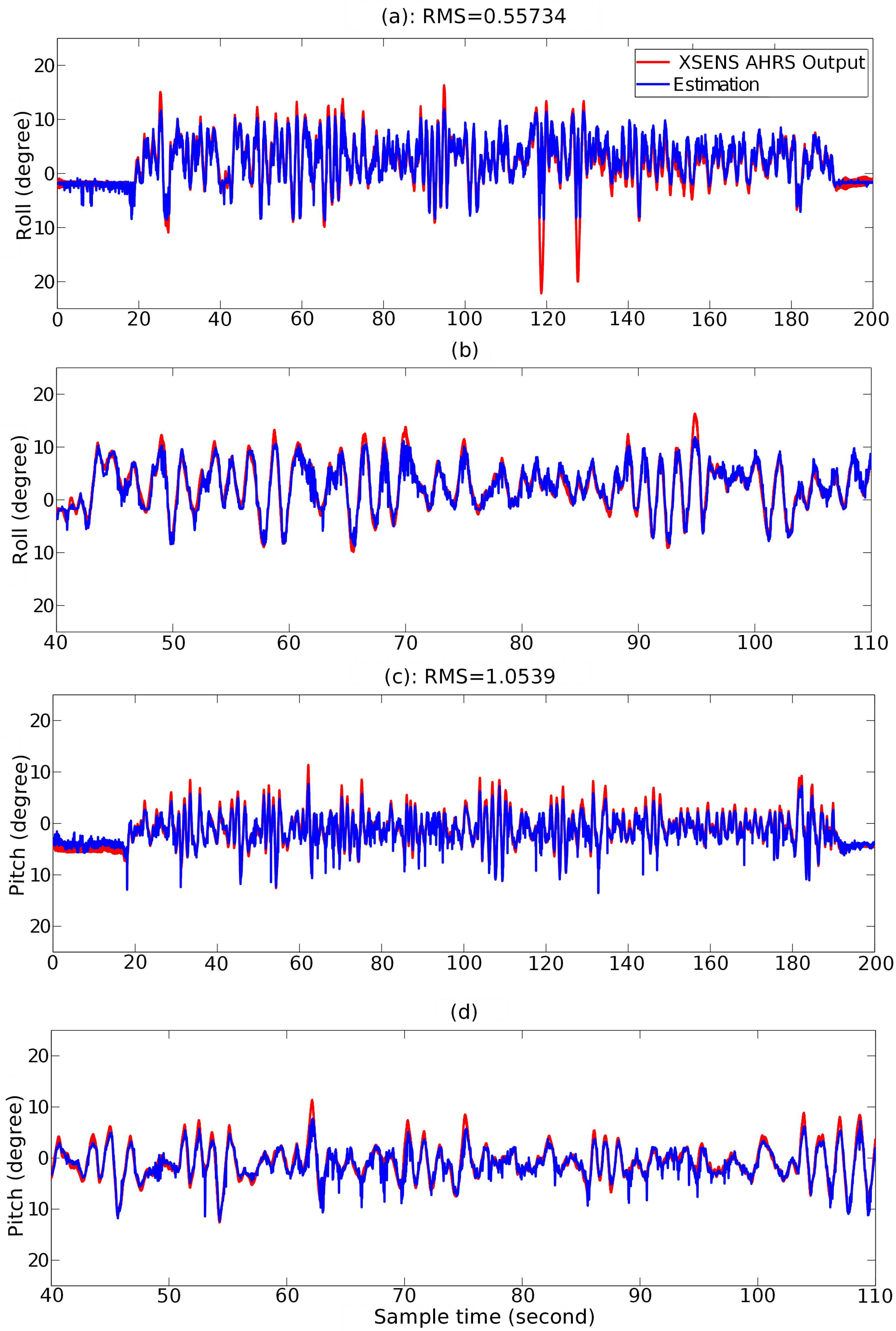


Figure 6.10: ANN attitude estimation using optic flow and vision

scale the attitude with the XSENS. However, such calibration equations are not required when using an ANN as the calibration is inherently dealt with in the training process. A trained ANN thus takes care of the mapping of the measurement nonlinearities due to different factors such as combined optical nonlinearities and also misalignments between the panoramic image and the camera optical axis which causes asymmetric barrel distortion to the panoramic projection. Real-time estimation of ANN are shown in Tables 6.1 and 6.2 and compared with the EKF estimation to justify the better performance of ANN compared to the EKF. Additional reasons behind obtaining more accurate results from ANN compared to EKF include:

1. The residual misalignment between the vision and the XSENS measurement are corrected more precisely inside the trained ANN's structure,

2. The visual measurements (for the horizon–based attitude estimation and the optic flow) are scaled better with the actual estimation by the trained ANN structure, and

3. The attitude estimator is modelled thoroughly inside the trained ANN structure including all dynamical characteristics of the system with better modelling of noise characteristics.

6.6 Real–Time System Architecture

Programs are prioritised to be sequentially executed in real–time. Some programs are called before others to prepare parameters that are to be used by other programs. The programs which estimate the attitude are executed in the following order:

1. Horizon–based attitude estimation,

2. Optic flow calculation,

3. Extended Kalman Filter, and

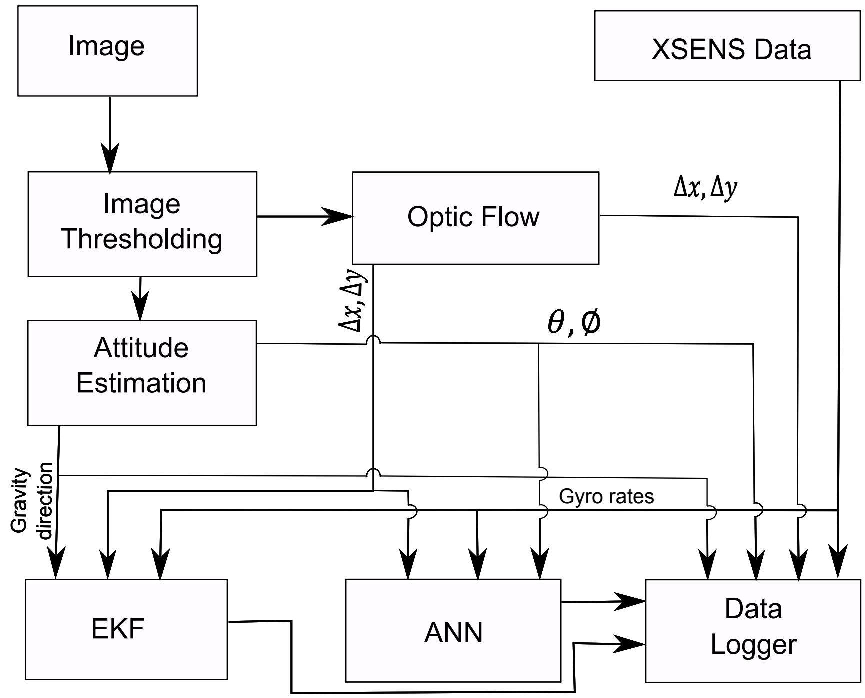


Figure 6.11: System architecture

4. Artificial Neural Networks.

Figure 6.11 shows a brief overview of the system architecture. In this work, as shown in the sections 6.3.3, 6.3.4, 6.5.1 and 6.5.2, the effect and accuracy of using each source of angular rates (gyroscopic or optic flow) on attitude estimation has been examined. During the experiment, just one source of angular rates is provided to the EKF or ANN and in none of the experiments, has the effect of using a combination of angular rates (gyroscopic and optic flow) been examined.

The computation time for the programs are shown in Table 6.3. The image processing of the horizon–based attitude estimation takes 7.5ms and this varies slightly depending on the number of horizon points and the number of iteration for trimming data to select the inliers. In conditions when the sun flare affects the image or a big object obstructs the horizon, more horizon points (image edges) are the extracted (due to the sun flare and affected shape of the true horizon line). Consequently more data points are available for the attitude estimation algorithm and more iterations are required to refine for the best horizon plane fit. In such conditions, the computation time varies around 100–500μs. The Sun–Tracking, the optic flow calculation, the EKF and ANN algorithms need a constant computation time in all conditions, with generally, their total computational time approximately around 13.3ms in real–time execution.

Table 6.3: Real–time computation time

Program	Computation Time (Real–Time)
Horizon–based Attitude Estimation	7.5ms
Sun–Tracking Thresholding	2.5ms
Optic flow	1.5ms
EKF	1.5ms
ANN	300ns
Total Time	**13.3ms**

6.7 Summary

Two sensor fusion techniques (the EKF and ANN) were implemented for attitude estimation and their performances were evaluated using real flight data. It was shown throughout experiments that, compared with raw attitude estimations, the attitude could be estimated more accurately with fewer RMS errors using sensor fusion. Tables 6.1 and 6.2 show that the performances of the ANN were better than those of the EKF for both roll and pitch angles irrespective of the source of the angular rates (gyroscope of optic flow). Both techniques, the estimations of which are compared in section 6.5.2, completely estimated the attitude from vision using optic flow (no gyro) and the horizon–based attitude. The two visual–based attitude estimators implemented were evaluated in sections 6.3.4 and 6.5.2 shown to be capable of independently estimating the attitude with no aid from the inertial sensors. In conclusion, the sensor fusion techniques outperformed the raw attitude estimations obtained from only vision and/or inertials, and visual information could be used independently as a reliable attitude reference.

Chapter 7

Conclusions and Recommendations

7.1 Summary of Achievements

The main aim of this thesis was to develop a novel biologically-inspired panoramic vision system capable of estimating the attitude of a helicopter flying close to the ground in a cluttered environment.

A new imaging system which was capable of capturing UV-filtered panoramic images with enhanced contrast between the sky and ground was developed. The lens filter's band–pass frequency chosen was based on inspiration from the Ocelli spectrum response of an insect and its functionality for horizon detection. Capturing panoramic images ensured that the horizon was always visible, even if only partially, under almost any circumstance.

Using a UV-filtered image made sky/ground segmentation easier than by processing a coloured or grey image with a deep texture and also mitigated the sun's luminosity. The imaging system was able to provide good contrasted images in all weather conditions, including sunny, partly cloudy and fully cloudy days, and also when the horizon was not clearly visible due to haze or smoke.

Sky/ground thresholding is one of the most critical issues due to its importance for horizon determination. Although two regions in an image (sky and ground) were distinguishable by enhancing the contrast, choosing an appropriate threshold was still challenging. A new method for determining the threshold for panoramic image thresholding was proposed. In addition, the effect of a sun–induced lens flare was corrected using a new method, called Sun-Tracking Thresholding, that recovered missing parts of the horizon caused by the lens flare.

Because of the use of panoramic optics, all associated horizon points were included in attitude estimation. The Least-Squared Error (LSE) was chosen as the estimator for calculating the best gravity direction estimation with minimal error. False horizon points were taken out of the LSE calculation using a new trimming method and combining data trimming with the LSE improved selection of the true horizon which consequently led to more precise attitude estimations. It was also proven through real flight testing that vibration had less effect on visual estimates of the vertical direction than the inertial sensors.

A new method for estimating attitude from the horizon was proposed and its results benchmarked against those from an IMU. It was shown that the body angular rates could be estimated from a panoramic image using I^2A algorithm. This method was able to estimate the attitude of a helicopter flying close to the ground in the presence of surrounding objects, such as trees and buildings.

An optic flow method, inspired by insect vision, was implemented to provide another means of measuring the body angular rates of the pitch and roll to complement the attitude estimation from the panoramic image. The advantage of using optic flow as a part of attitude estimation was proven by the improved the performance and reduction in the RMS error when it was integrated with horizon–based data.

Finally, two sensor fusion techniques for integrating different data were implemented and compared. An Extended Kalman Filter was used as a mathematically based solution for means non–linear state estimation and an artificial neural network was used to follow up on the biological inspiration to mimic the brain's nervous system for the purpose of sensor fusion with both techniques showing improved attitude estimation accuracy. Although fusing optic flow measurement with horizon–based attitude estimation improved the accuracy of attitude estimation, but the performances of both sensor fusion techniques were better using gyroscopic measurements, with that of the EKF. This was due not only to the complete non–linear system mapping of the vision system inside the trained ANN structure but also the ANN's capability to correct various external factors, such as the platform mis–alignments and the non–linearity of the optics.

7.2 Recommendations for Future Work

There are some areas that could be addressed to further develop the work in this thesis. An accurate attitude with a low RMS error could be used independently (with no help from an IMU or GPS) as input data to a closed–loop control system of an aircraft for attitude stabilisation. Although the vision system in this work showed its competency in providing an accurate attitude estimation, the methods and techniques used could be extended to consider more situations. Besides achieving good results, some of the following changes could be made to the hardware and software to improve the quality of this work:

1. A cheap CCD camera was modified by replacing its CCD filter with a UV–pass filter for imaging UV wavelengths (300–400nm). Although this camera functioned acceptably in UV and sky/ground regions were distinguishable by human eyes, distributions of the pixel values (sky and ground) were not sufficiently wide to easily determine the threshold. Therefore, as a camera with greater sensitivity to UV could enhance the sky/ground contrast. A commercially designed UV camera would be a better option for capturing image in UV wavelengths than modifying a general CCD camera.

2. Although the threshold values were well–calculated using two threshold masks, it would be worthwhile calculating them with respect to the locus points of sky/ground regions to select appropriate sampling pixels. With a clear horizon

definition that leads to attitude estimation, the locus points of the sky and ground respect to the 3D estimated horizon plane could be easily be found.

3. In addition to being used for Sun-Tracking thresholding, the position of the sun could also be used to find the north direction which could be used as a reference (provided the geographical location and time were known). In such a case, more consideration could be given to calculating the exact location of the sun because of the brightness-changing effect of its flare for finding its center.

4. Although the ANN improved the quality of attitude estimation (when trained using five flight tests) in different situations, more flight data could be used to train it well and contain all combinations. Different environmental and weather conditions impacted differently on horizon visibility and image quality and affected the algorithms used in this work. To comprehensively train the ANN to cope with all situations, a comprehensive dataset is required.

5. Although good attitude estimations were achieved in the cluttered environments (where the experiments were conducted), close objects could block the true horizon and produce an artificial horizon definition which would have a major effect on horizon–based attitude estimation. Although the effects of close objects were mitigated due to the geometry of the panoramic lens (which varied depending on the size and distance), there could still be an offset in the estimation due to very large close objects such as tall buildings. This could be compensated using other sensory information, such as highly filtered inertial data, as a reference to detect the existence of offsets existence in vision estimations.

7.3 Concluding Remarks

A novel biologically-inspired panoramic vision system was developed to estimate the attitude of a helicopter flying close to the ground in a cluttered environment. The objectives of the work were fulfilled and experimentally verified which demonstrated the advantages of determining the pitch and roll angles of a moving platform from a vision–based rather than inertial–based system. Estimating the body angular rates from panoramic image motions was investigated and proven to be feasible. The horizon–based attitude estimation provides a reliable, passive sensing measurement and less vulnerability to vibration for a small helicopter.

References

[1] J. Hintze, "Autonomous landing of a Rotary Unmanned Aerial Vehicle in a non-cooperative environment using machine vision," Master's thesis, 2004.

[2] T. Hamel and R. Mahony, "Attitude estimation on SO [3] based on direct inertial measurements," in *International Conference on Robotics and Automation*, ICRA, Ed. IEEE, 2006, pp. 2170–2175.

[3] D. Biezad, *Integrated navigation and guidance systems.* AIAA, 1999, vol. 1.

[4] S. Chen, Y. Li, J. Zhang, and W. Wang, "Active vision sensors," in *Active Sensor Planning for Multiview Vision Tasks.* Springer, 2008, pp. 11–38.

[5] J. M. McMichael and M. S. Francis, "Micro Air Vehicles–toward a new dimension in flight," *DARPA document*, 1997.

[6] A. Sherman and M. H. Dickinson, "A comparison of visual and haltere-mediated equilibrium reflexes in the fruit fly Drosophila melanogaster," *Journal of experimental biology*, vol. 206, no. 2, pp. 295–302, 2003.

[7] T. Trevor, *The instrument rating manual : complete preparation for the command instrument rating.* Williamstown, Vic. : Aviation Theory Centre, 1994.

[8] B. Ford Roth, "Spatial disorientation caused f-16 crash that killed california air force pilot, http://www.kpbs.org/news/2013/oct/30/spatial-disorientation-f-16-crash-lucus-gruenther," April 2014.

[9] S. Disorientation, "Encyclopaedia Britannica Online," http://www.britannica.com/EBchecked/topic/558427/spatial-disorientation, 2014.

[10] E. Baird, E. Kreiss, W. Wcislo, E. Warrant, and M. Dacke, "Nocturnal insects use optic flow for flight control," *Biology letters*, vol. 7, no. 4, pp. 499–501, 2011.

[11] R. Menzel, D. Ventura, H. Hertel, J. De Souza, and U. Greggers, "Spectral sensitivity of photoreceptors in insect compound eyes: comparison of species and methods," *Journal of Comparative Physiology A*, vol. 158, no. 2, pp. 165–177, 1986.

[12] G. Stange, J. Van Kleef, R. Berry, and R. Parker, "A biomimetic algorithm for flight stabilization in airborne vehicles, based on dragonfly Ocellar vision," Australian National University Canberra (Australia) research school of biological sciences, Tech. Rep., 2006.

[13] M. Wilson, "The functional organisation of locust Ocelli," *Journal of Comparative Physiology, Neuroethology, Sensory, Neural, and Behavioral Physiology, Springer*, vol. 124, no. 4, pp. 297–316, 1978.

[14] R. Hengstenberg, "Gaze control in the blowfly Calliphora: a multisensory, two–stage integration process," *Seminars in Neuroscience*, vol. 3, no. 1, pp. 19–29, 1991.

[15] D. Koks, "Using rotations to build aerospace coordinate systems," DTIC Document, DTIC Document, Astronautics 42, AUG 2008, technical note.

[16] E. Bekir, *Introduction to modern navigation systems.* World Scientific Publishing Co. Pte. Ltd., 2007, printed in Singapore by Mainland Press.

[17] B. L. Stevens and F. L. Lewis, *Aircraft control and simulation.* Wiley-Interscience, 2003.

[18] R. P. G. Collinson, *Introduction to avionics.* Kluwer Academic Publishers, 1996.

[19] J. Lobo and J. Dias, "Vision and inertial sensor cooperation using gravity as a vertical reference," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1597–1608, 2003.

[20] K. R. B., *Inertial navigation systems analysis.* wiley interscience, New York, 1971.

[21] D. H. Titterton and J. L. Weston, *Strapdown inertial navigation technology.* Peter Peregrinus Ltd, 2004.

[22] A. Lawrence, *Modern Inertial Technology: Navigation, Guidance, and Control.* Springer, 1998.

[23] O. J. Woodman, "An introduction to inertial navigation," University of Cambridge, Computer Laboratory, Tech. Rep. UCAMCL-TR-696, Tech. Rep., 2007.

[24] P. D. Groves, *Principles of GNSS, inertial, and multisensor integrated navigation systems.* Artech House, 2007.

[25] T. Thom, "The flying training manual–pre–flight briefings and air exercises," *Aviation Theory Centre Pty. Ltd., Williamstown, Australia*, 1993.

[26] S. Todorovic and M. C. Nechyba, "Intelligent missions for MAVs: visual contexts for control, tracking and recognition," in *International Conference on Robotics and Automation, Proceedings. ICRA'04.* IEEE, 2004.

[27] S. Todorovic, M. C. Nechyba, and P. G. Ifju, "Sky/ground modeling for autonomous MAV flight," in *Proceedings IEEE International Conference on Robotics and Automation (ICRA)*, May 2003, pp. 1422–1427.

[28] G. D. Hager and P. N. Belhumeur, "Efficient region tracking with parametric models of geometry and illumination," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 10, pp. 1025–1039, 2002.

[29] G. Bao, Z. Zhou, S. Xiong, X. Lin, and X. Ye, "Towards Micro Air Vehicle flight autonomy research on the method of horizon extraction," in *Proceedings of the 20th Instrumentation and Measurement Technology Conference. IMTC'03.* IEEE, 2003.

[30] T. D. Cornall, A. Price, and G. K. Egan, "Measuring horizon angle on a small Unmanned Air Vehicle using digital video camera and an FPGA," in *ICARA International Conference on Autonomous Robots an Agents*, 2006.

[31] S. M. Ettinger, M. C. Nechyba, P. G. Ifju, and M. Waszak, "Vision–guided flight stability and control for Micro Air Vehicles," in *International Conference on Intelligent Robots and Systems, 2002. IEEE/RSJ*, vol. 3, 2002, pp. 2134–2140.

[32] D. Dusha, "Integrated Vision–Based Attitude Estimation," Ph.D. dissertation, Australian Research Centre for Aerospace Austomation, Queensland University of Technology, 2012.

[33] S. Todorovic and M. C. Nechyba, "Multiresolution linear discriminant analysis: efficient extraction of geometrical structures in images," in *International Conference on Image Processing, 2003. ICIP 2003. Proceedings.*, vol. 1. IEEE, 2003.

[34] S. Ettinger, M. Nechyba, P. Ifju, and M. Waszak, "Vision–guided flight stability and control for Micro Air Vehicles," *Advanced Robotics*, vol. 17, no. 7, pp. 617–640, 2003.

[35] I. F. Mondragón, M. A. Olivares-Méndez, P. Campoy, C. Martínez, and L. Mejias, "Unmanned aerial vehicles UAVs attitude, height, motion estimation and control using visual systems," *Autonomous Robots, Springer*, vol. 29, no. 1, pp. 17–34, 2010.

[36] T. G. McGee, R. Sengupta, and K. Hedrick, "Obstacle detection for small autonomous aircraft using sky segmentation," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation, 2005. ICRA 2005.* IEEE, 2005, pp. 4679–4684.

[37] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data mining and knowledge discovery, Springer*, vol. 2, no. 2, pp. 121–167, 1998.

[38] D. A. Forsyth and J. Ponce, *Computer vision: a modern approach.* Prentice Hall Professional Technical Reference, 2002.

[39] R. O. Duda and P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, 1972.

[40] R. C. Gonzalez and R. E. Woods, "Digital image processing, 2nd," *Prentice Hall*, 2002.

[41] D. Dusha, W. W. Boles, and R. Walker, "Fixed–wing attitude estimation using computer vision based horizon detection," *In Proceedings 12th Australian International Aerospace Congress. Melbourne Australia*, no. 1–19, 2007.

[42] S. Todorovic and M. C. Nechyba, "A vision system for intelligent mission profiles of Micro Air Vehicles," *IEEE Transactions on Vehicular Technology,*, vol. 53, no. 6, pp. 1713–1725, 2004.

[43] T. D. Cornall, G. K. Egan, and A. Price, "Aircraft attitude estimation from horizon video," *Electronics Letters*, vol. 42, no. 13, pp. 744–745, 2006.

[44] N. Otsu, "A threshold selection method from gray-level histograms," *Automatica*, pp. 285–296, 1975.

[45] S. Thurrowgood, D. Soccol, R. Moore, D. Bland, and M. V. Srinivasan, "A vision based system for attitude estimation of UAVs," in *International Conference on Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ.* IEEE, 2009.

[46] J. Chahl, S. Thakoor, N. Le Bouffant, G. Stange, M. Srinivasan, B. Hine, and S. Zornetzer, "Bioinspired engineering of exploration systems: A horizon sensor/attitude reference system based on the dragonfly Ocelli for Mars exploration applications," *Journal of Robotic Systems*, vol. 20, no. 1, pp. 35–42, 2003.

[47] D. McCutchen, "Method and apparatus for dodecahedral imaging system," 1991, 31 N. Gardner St., W. Hollywood, California 90046 (US).

[48] S. Saripalli, J. F. Montgomery, and G. S. Sukhatme, "Vision–based autonomous landing of an Unmanned Aerial Vehicle," in *IEEE International Conference on Robotics and Automation, Proceedings. ICRA'02.*, vol. 3, no. 3. IEEE, 2002, pp. 2799–2804.

[49] C. S. Sharp, O. Shakernia, and S. S. Sastry, "A vision system for landing an Unmanned Aerial Vehicle," in *IEEE International Conference on Robotics and Automation, 2001. Proceedings 2001 ICRA.*, vol. 2. IEEE, 2005, pp. 1720–1727.

[50] P. J. Garcia-Pardo, G. S. Sukhatme, and J. F. Montgomery, "Towards vision–based safe landing for an autonomous helicopter," *Robotics and Autonomous Systems, Elsevier*, vol. 38, no. 1, 2002.

[51] S. Thurrowgood, R. J. D. Moore, D. Bland, D. Soccol, and M. V. Srinivasan, "UAV attitude control using the visual horizon," in *Australasian Conference on Robotics and Automation 2010 (ACRA 2010)*, 2011.

[52] J. Puchberger, "patents a daguerrotype panoramic camera with a hand crank driving a swing lens covering a 150 degree arc," 1843.

[53] D. W. Rees, "Panoramic television viewing system," United States Patent, 1970, no. 3, 505, 465.

[54] R. Benosman and S. B. Kang, *Panoramic vision: sensors, theory, and applications.* Springer Verlag, 2001.

[55] S. Baker and S. K. Nayar, "A theory of catadioptric image formation," in *In Proceedings of the 6th International Conference on Computer Vision, Bombay, India*, January 1998, pp. 35–42.

[56] S. Hrabar and G. S. Sukhatme, "Omnidirectional vision for an autonomous helicopter," in *IEEE International Conference on Robotics and Automation, 2003. Proceedings. ICRA'03.*, vol. 1. IEEE, 14–19 September 2003, pp. 558–563.

[57] P. Chang and M. Hebert, "Omni–directional visual servoing for human-robot interaction," in *International Conference on Intelligent Robots and Systems. Proceedings*, vol. 3. IEEE, 1998, pp. 1801–1807.

[58] N. Winters, J. Gaspar, G. Lacey, and J. Santos-Victor, "Omni–directional vision for robot navigation." Published by the IEEE Computer Society, 2000, p. 21.

[59] C. Demonceaux, P. Vasseur, and C. Pegard, "Robust attitude estimation with catadioptric vision," in *International Conference on Intelligent Robots and Systems, 2006 IEEE/RSJ.* IEEE/RSJ, 2006, pp. 3448–3453.

[60] W. Stürzl, D. Soccol, J. Zeil, N. Böddeker, and M. V. Srinivasan, "Rugged, obstruction–free, mirror–lens combination for panoramic imaging,," *Applied optics*, vol. 47, no. 32, pp. 6070–6078, 2008.

[61] M. V. Chahl, J. S. Srinivasan, "Filtering and processing of panoramic images obtained using a cemera and a wide–angle-imaging reflective surface," *OPTICAL SOCIETY OF AMERICA*, vol. 17, no. 7, pp. 1172–1176, 2000.

[62] R. Moller, "Insects could exploit UV–green contrast for landmark navigation," *Journal of theoretical biology, Elsevier*, vol. 214, no. 4, 2002.

[63] R. Menzel, "Colour receptors in insects," *The compound eye and vision of insects*, pp. 121–153, 1975.

[64] H. Autrum and V. Zwehl, "Die spektrale empfindlichkeit einzelner sehzellen des bienenauges," *Zeitschrift für vergleichende Physiologie*, vol. 48, no. 4, pp. 357–384, 1964.

[65] R. Menzel and M. Giurfa, "Cognitive architecture of a mini-brain: the honeybee," *Trends in cognitive sciences*, vol. 5, no. 2, pp. 62–71, 2001.

[66] F. A. Azevedo, L. Carvalho, L. Grinberg, J. Farfel, R. Ferretti, R. E. Leite, R. Lent, S. Herculano-Houzel, and others., “Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled–up primate brain,” *Journal of Comparative Neurology*, vol. 513, no. 5, pp. 532–541, 2009.

[67] E. Baird, T. Kornfeldt, and M. Dacke, “Minimum viewing angle for visually guided ground speed control in bumblebees,” *The Journal of experimental biology*, vol. 213, no. 1625–1632, 2010.

[68] J. K. Douglass and N. J. Strausfeld, “Anatomical organization of retinotopic motion-sensitive pathways in the optic lobes of flies,” *Microscopy research and technique*, vol. 62, no. 2, pp. 132–150, 2003.

[69] M. A. Frye and M. H. Dickinson, “Fly flight: a model for the neural control of complex behavior,” *Neuron, Elsevier*, pp. 385–388, 2001.

[70] M. A. Ali, *Photoreception and vision in invertebrates.* Plenum Press, 1984.

[71] P. S. Bhagavatula, C. Claudianos, M. R. Ibbotson, and M. V. Srinivasan, “Optic flow cues guide flight in birds,” *Current Biology, Elsevier*, vol. 21, no. 21, pp. 1794–1799, 2011.

[72] W. Wu, R. Wood, and R. Fearing, “Halteres for the micromechanical flying insect,” *IEEE International Conference onRobotics and Automation, 2002. Proceedings. ICRA .*, vol. 1, pp. 60–65, 2002.

[73] W. Wu, L. Schenato, R. J. Wood, and R. S. Fearing, “Biomimetic sensor suite for flight control of a micromechanical flight insect: design and experimental results,” in *in Proc. IEEE Int. Conf. Robotics and Automation, Taipei, Taiwan*, vol. 1. IEEE, 2003, pp. 1146–1151.

[74] G. Barrows, J. S. Chahl, and M. Srinivasan, “Biomimetic visual sensing and flight control,” *The Aeronautical Journal, London: The Royal Aeronautical Society*, vol. 107, no. 1069, 2003.

[75] L. Schenato, W. C. Wu, and S. Sastry, “Attitude control for a micromechanical flying insect via sensor output feedback,” *IEEE Transactions on Robotics and Automation*, vol. 20, no. 1, pp. 93–106, 2004.

[76] S. J. Huston and H. G. Krapp, “Nonlinear integration of visual and haltere inputs in fly neck motor neurons,” *The Journal of Neuroscience*, vol. 29, no. 42, pp. 13 097–13 105, 2009.

[77] R. Rosner, M. Egelhaaf, J. Grewe, and A.-K. Warzecha, “Variability of blowfly head optomotor responses,” *Journal of Experimental Biology*, vol. 212, no. 8, pp. 1170–1184, 2009.

[78] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, “Performance of optical flow techniques,” *International journal of computer vision*, vol. 12, no. 1, pp. 43–77, 1994.

[79] H. Nagel, “Displacement vectors derived from second-order intensity variations in image sequences,” *Computer Vision, Graphics, and Image Processing*, vol. 21, no. 1, pp. 85–117, 1983.

[80] B. K. Horn and B. G. Schunck, “Determining optical flow,” *Artificial intelligence*, vol. 17, no. 1, pp. 185–203, 1981.

[81] B. D. Lucas, "Generalized image matching by the method of differences," Ph.D. dissertation, Carnegie Mellon University, 1985.

[82] B. D. Lucas, T. Kanade *et al.*, "An iterative image registration technique with an application to stereo vision." in *Proceedings of Imaging Understanding Workshop*, vol. 81, 1981, pp. 121–130.

[83] A. Singh, *Optic flow computation: a unified perspective.* IEEE Computer Society Press California, 1991, vol. 9.

[84] P. Anandan, "A computational framework and an algorithm for the measurement of visual motion," *International Journal of Computer Vision*, vol. 2, no. 3, pp. 283–310, 1989.

[85] ——, "Measuring visual motion from image sequences," Ph.D. dissertation, University of Massachusetts, 1987.

[86] A. Singh, "An estimation–theoretic framework for image–flow computation," *Proceedings, Third International Conference on Computer Vision*, pp. 168–177, 1990.

[87] D. J. Heeger, "Model for the extraction of image flow," *Optical Society of America*, vol. 7, no. 8, pp. 1455–1471, 1987.

[88] ——, "Optical flow using spatiotemporal filters," *International Journal of Computer Vision*, vol. 1, no. 4, pp. 279–302, 1988.

[89] A. M. Waxman, J. Wu, and F. Bergholm, "Convected activation profiles and the measurement of visual motion." IEEE, 1988, pp. 717–723.

[90] D. J. Fleet and A. D. Jepson, "Computation of component image velocity from local phase information," *International Journal of Computer Vision*, pp. 77–104, 1990.

[91] M. V. Srinivasan, "An image–interpolation technique for the computation of optic flow and egomotion," *Biological Cybernetics*, vol. 71, no. 5, pp. 401–415, 1994.

[92] J. S. Chahl, M. V. Srinivasan, and S. Zhang, "Landing strategies in honeybees and applications to Uninhabited Airborne Vehicles," *The International Journal of Robotics Research*, vol. 23, no. 2, pp. 101–110, 2004.

[93] M. V. Srinivasan, J. S. Chahl, K. Weber, S. Venkatesh, S. Zhang, and M. G. Nagle, "From insects to robots," in *Experimental Robotics VI.* Springer, 2000, pp. 3–12.

[94] D. Mandic and J. Chambers, *Recurrent Neural Networks for prediction: Architectures, learning algorithms and stability.* Wiley New York, 2001.

[95] S. Haykin, *Neural networks: a comprehensive foundation*, 2nd ed. Prentice Hall, 1999.

[96] M. A. Garratt, "Biologically inspired vision and control for an Autonomous Flying Vehicle," Ph.D. dissertation, 2010.

[97] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks, Elsevier*, vol. 2, no. 5, pp. 359–366, 1989.

[98] K. Levenberg, "A method for the solution of certain problems in Least Squares," *Quarterly of applied mathematics*, vol. 2, no. 164–168, 1944.

[99] D. W. Marquardt, "An algorithm for Least–Squares estimation of nonlinear parameters," *Journal of the Society for Industrial & Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[100] M. T. Hagan and M. Menhaj, "Training feedforward networks with the Marquardt algorithm," *IEEE Transactions on Neural Network*, vol. 5, pp. 989–993, 1994.

[101] I. The MathWorks, "Neural network toolbox user's guide," www.mathworks.com, 2014.

[102] C. C. Chang and K. Song, "Ultrasonic sensor data integration and its application to environment perception," *Journal of Robotic Systems*, vol. 13, no. 10, pp. 633–677, 1996.

[103] S. Thrun, "Learning metric–topological maps for indoor mobile robot navigation," *Artificial Intelligence*, vol. 99, no. 1, 1998.

[104] R. Sharaf, A. Noureldin, A. Osman, and N. El-Sheimy, "Online INS/GPS integration with a radial basis function neural network," *Aerospace and Electronic Systems Magazine, IEEE*, vol. 20, no. 3, pp. 8–14, 2005.

[105] N. El-Sheimy, K. Chiang, and A. Noureldin, "The utilization of Artificial Neural Aetworks for multisensor system integration in navigation and positioning instruments," *IEEE Transactions on Instrumentation and Measurement*, vol. 55, no. 5, pp. 1606–1615, 2006.

[106] R. E. Kalman *et al.*, "A new approach to linear filtering and prediction problems," *Journal of basic Engineering*, vol. 82, no. 1, 1960.

[107] Y. Ho and R. Lee, "A Bayesian approach to problems in stochastic estimation and control," *IEEE Transactions on Automatic Control*, vol. 9, no. 4, pp. 333–339, 1964.

[108] P. Zarchan and H. Musoff, *Fundamentals of Kalman Filtering: a practical approach.* AIAA, 2005, vol. 208.

[109] C. T. Leondes, "Theory and applications of Kalman filtering," *AGARDograph*, 1970.

[110] G. Fuquan, D. Chuanhong, and L. Jianfeng, "Initial alignment of strap down inertial navigation system using Kalman filter," in *International Conference on Computer Application and System Modeling (ICCASM)*, 2010.

[111] D. Dusha and L. Mejias, "Attitude observability of a loosely–coupled GPS/Visual odometry integrated navigation filter," in *Australasian Conference on Robotics and Automation (ACRA 2010)*, 2010.

[112] G. Adiv, "Determining three–dimensional motion and structure from optical flow generated by several moving objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 4, pp. 384–401, 1985.

[113] M. Subbarao and A. M. Waxman, "Closed from solutions to image flow equations for planar surfaces in motion," *Computer Vision, Graphics, and Image Processing*, vol. 36, no. 2, pp. 208–228, 1986.

[114] A. D. Wu, E. N. Johnson, and A. A. Proctor, "Vision-aided inertial navigation for flight control," in *Journal of Aerospace Computing, Information, and Communication*, vol. 2, 2005.

[115] V. Gupta and S. Brennan, "Terrain–based vehicle orientation estimation combining vision and inertial measurements," *Journal of Field Robotics*, vol. 25, no. 3, pp. 181–202, 2008.

[116] J. S. Chahl and M. V. Srinivasan, "Reflective surfaces for panoramic imaging," *Applied Optics*, vol. 36, no. 31, pp. 8275–8285, 1997.

[117] "http://www.slackware.com/," *The Slackware Linux Project*, 2014.

[118] "Fsm labs, rtlinux free," www.rtlinuxfree.com, 2014.

[119] "Gcc, the gnu compiler collection," http://gcc.gnu.org/, 2014.

[120] "3dm-gx1 imu," http://www.microstrain.com/, 2014.

[121] "Xsens imu sensor," http://www.xsens.com/, 2014.

[122] *MobaXterm, the complete toolbox for remote computing*, http://mobaxterm.mobatek.net/, 2014.

[123] *EZWGL Library*, http://www.ma.utexas.edu/users/mzou/EZWGL/, 2014.

[124] C. Pavel and V. Jan Ámos, *Least Trimmed Squares.* Springer.

[125] R. J. Noriega-Manez, "Inertial navigation," Submitted as coursework for Physics 210, Stanford University, October 2007.

[126] H. Mittelstaedt, "Physiologie des gleichgewichtssinnes bei fliegenden libellen," *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, vol. 32, no. 5, 1950.

[127] J. J. Gibson, *The perception of the visual world.* Houghton Mifflin, 1950.

[128] A. Borst and J. Haag, "Neural networks in the cockpit of the fly," *Journal of Comparative Physiology A*, vol. 188, no. 6, pp. 419–437, 2002.

[129] R. Wehner, B. Michel, and P. Antonsen, "Visual navigation in insects: coupling of egocentric and geocentric information," *Journal of Experimental Biology*, vol. 199, no. 1, pp. 129–140, 1996.

[130] Y. Le Cun, "Efficient learning and second order methods," *Tutorial presented at Neural Information Processing Systems*, 1993, Denvar.