Aircraft Manoeuvring for Sensor Aiming

by

© Kevin A. Murrant

A thesis submitted to the

School of Graduate Studies

in partial fulfilment of the

requirements for the degree of

Doctor of Philosophy

Faculty of Engineering & Applied Science

Memorial University of Newfoundland

May 2018

St. John's Newfoundland

Abstract

Airborne sensor aiming can be achieved with a fixed sensor using the manoeuvrability of an aircraft. Such a method offers advantages in potential sensor coverage and reduced payload complexity. Without use of a gimbal, an aircraft can be made more robust and sensor aiming is limited only by the aircraft flight capabilities.

A novel method is developed and demonstrated for performing sensor aiming with a fixed-wing aircraft. A creative mathematical framework is presented for both a 3D path following controller and a method to seamlessly achieve sensor aiming while minimizing path deviation. A simulation environment is developed based on a fit-for-purpose aircraft model identified from live flight testing and the control algorithms are validated.

Flight test data is presented demonstrating efficacy of the 3D path following controller. These demonstrations also serve to validate the aircraft modelling approach taken during controller development.

Two application examples involving airborne radar aiming for detect and avoid and gimbal-less ground target tracking are used to illustrate the sensor aiming method. The proposed sensor aiming methodology is both practical and feasible as supported by results. The proposed method is applicable to both unmanned and manned aircraft. Future work involving the concept of manoeuvrable sensors is proposed in the conclusion.

Acknowledgements

First, I would like to thank my supervisor, Dr. Siu O'Young, for his guidance and support during this work. He continually goes beyond expectation for his students.

I want to thank my colleagues in the Remote Aerial Vehicle for ENvironmental monitoring (RAVEN) project for their support and comaraderie throughout the project and onward. I would also like to thank my colleagues at the National Research Council (NRC), in particular Dr. Jim Millan and Bob Gash for their helpful discussion.

To my wife Marlena, children Anna, Leo, and Gus, and my parents, I owe thanks for all their love and support.

I would like to acknowledge the various funding agencies that supported this research: Atlantic Innovation Fund, Natural Science and Engineering Research Council (NSERC), Bombardier Aerospace, and Raytheon.

Contents

List of Tables

List of Figures

List of Abbreviations

AHRS	Attitude and Heading Reference System
AUV	Autonomous Underwater Vehicle
DAA	Detect and Avoid
DOF	Degrees of Freedom
EKF	Extended Kalman Filter
EOM	Equations of Motion
FOV	Field of View
HAL	Hardware Abstraction Layer
HIL	Hardware in-the-Loop
GCS	Ground Control Station
GPS	Global Positioning System
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
LQR	Linear Quadratic Regulator
MEKF	Multiplicative Extended Kalman Filter
MIMO	Multiple-Input Multiple-Output
MPC	Model Predictive Control
MSE	Mean Squared Error
OTS	Off-the-Shelf
PID	Proportional-Integral-Derivative
PF	Path Following
RAVEN	Remote Aerial Vehicle for ENvironmental monitoring

RC	Remote Control
RMS	Root Mean Square
RTOS	Real Time Operating System
SA	Sensor Aiming
UAV	Unmanned Aerial Vehicle
VTOL	Vertical Take Off and Landing

Chapter 1

Introduction

Unmanned Aerial Vehicles (UAVs) have become widely deployed with recreational, commercial/industrial, and military users. Being unmanned, sensors are required for the operation and control of the aircraft, with additional sensors often included for data collection. Several of these additional sensors require motion in addition to that which can be achieved fixed to the airframe, i.e. gimbaling. Examples include cameras, lidar, or radar, which all normally feature one or more additional axes of motion to collect data.

Considering UAVs as airborne sensors in the context of data collection, the sensor motion can be divided into two main components: flight motion relative to the Earth; and sensor motion relative to the airframe. Flight motion involves manoeuvring the aircraft along the flight path which is chosen to survey an area and comprises the main focus of the mission plan. Sensor motion relative to the airframe may include motion compensation (smoothing), panning, tilting, or scanning and is not necessarily part of the flight plan. This is most commonly achieved with a gimbal mechanism. Sensor

motion relative to the airframe is usually independent of the flight motion to follow the flight path.

Path following (PF) is the high-level basis of flight motion in this context. To retain altitude, an aircraft must maintain a minimum airspeed and therefore is in constant motion towards a target position. The path can be defined as a sequence of target positions in space. PF refers to the tracking of such a sequence by a vehicle, without temporal constraints. This is in contrast to trajectory following in space which requires tracking of a sequence of positions at specified times. At a lower level, the PF control objectives are mapped to the control surfaces which also effect a change on the attitude of the aircraft.

Sensor aiming (SA) involves aligning the sensor field of view (FOV) within a specified orientation, for a given position, for data collection purposes. Conventionally, SA is achieved by combining flight motion with sensor motion relative to the airframe (normally using a gimbal) such that the two are independent.

Sensor motion relative to the airframe poses a considerable design challenge, particularly for small UAVs. Mechanical gimbals increase the payload weight and electrical requirements. Gimbals themselves are susceptible to environmental effects such as water or freezing, or malfunction. The airframe can occlude the sensor in some orientations unless it protrudes from the fuselage, which can affect the aerodynamics. Sensor FOV can also be enhanced optically or electronically, but not without drawbacks in reduced precision and increased complexity.

SA can be achieved for a sensor in a fixed orientation relative to the airframe by augmenting the flight motion. This reduces or eliminates the need for mechanization and FOV constraints. The primary drawback is compromised PF performance as the

aircraft accommodates the additional motion requirements. Secondarily, SA using aircraft attitude is limited to a finite duration, due to flight safety. Therefore, SA and PF objectives must be combined when using the gimbal-less approach.

Sensors deployed via UAV require fulfilment of a variety of requirements to achieve the sensory task. For example, target detection, classification, and/or tracking require a specific time-on-target. These requirements define whether it is acceptable to have interruption in sensor coverage and thus whether SA is practically achieved by the airframe alone.

The concept of combining flight and sensor motion to achieve SA is applicable to both fixed-wing and multi-rotor aircraft; in this thesis, fixed-wing aircraft are the platform considered due to their manoeuvrability characteristics. In comparison to multi-rotor aircraft, fixed-wing aircraft allow an order of magnitude increase in range, payload capacity, and efficiency. The primary drawbacks of fixed-wing aircraft are their difficulty to launch, recover, and hold position. These issues can be overcome by design and training, thus making the fixed-wing platform viable for manoeuvring-based SA.

1.1 Problem Statement

Airborne sensors often require additional motion beyond PF to achieve SA. Achieving this additional motion with a gimbal mechanism involves additional weight, complexity, and is inherently limited in performance. For most small UAVs, the inclusion of an aiming system is prohibitive unless the airframe is specifically designed for that purpose. The problem can be stated as:

> "UAVs can benefit from improved sensing capability and reduced payload complexity by performing sensor aiming without the use of a gimbal."

The proposed solution involves combining aircraft motion via automatic control methods to facilitate SA with a fixed orientation sensor. While intuitively feasible, implementation details and performance metrics remain unexplored in the current state of the art.

1.2 Technical Challenges

The technical challenges of the proposed solution include:

- design of a flight controller that can achieve the motions required for SA.
- integrating PF with SA such that both can be achieved smoothly and with minimum error.
- development of a simulation model and test platform to design & validate the controller.

Separation of PF and SA controllers allows each to be independently designed, which is appropriate for initial development and, as such, is the decided approach. Tight integration of control using a path planning approach is a viable alternative design that offers potential advantages in minimization of path error and control effort but involves far greater complexity in operation.

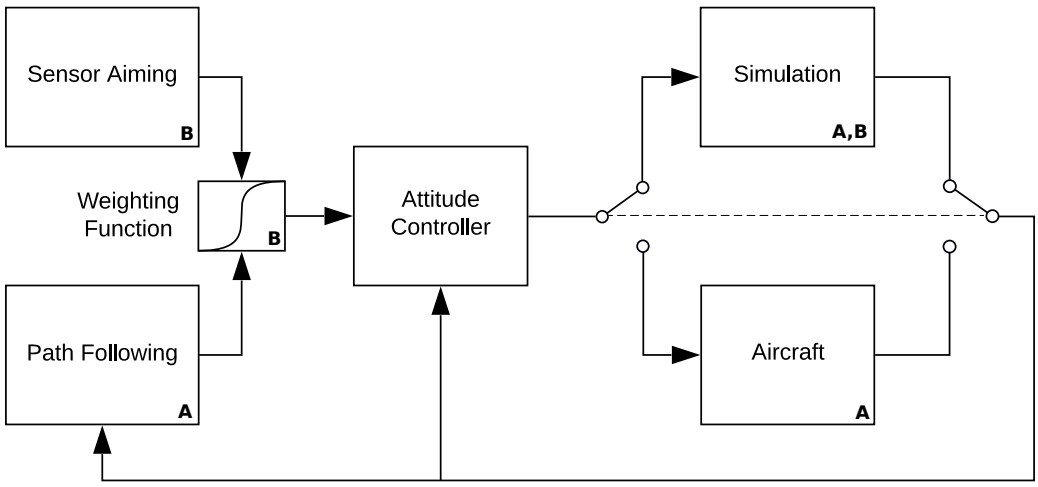

Figure 1.1: Flight control system block diagram indicating publication.

1.3 Contributions

The original contributions of this thesis are shown with the subscripts A and B in Figure 1.1. The subscripts A and B refer to two journal papers produced from this work. Paper A is titled "Attitude-based Path Following for Fixed-Wing UAVs" and Paper B is titled "Manoeuvre-based Sensor Aiming for Fixed-Wing UAVs". These completed papers are pending acceptance for publication.

Contributions in each area are as follows:

1. SA provides a control methodology (Section 3.4) for smoothly integrating SA motion with PF. Example applications examine implementation details and demonstrate capability (Section 4.7.2).

2. PF using aircraft attitude as an intermediate control variable is uniquely compatible with SA. A creative PF method is proposed which reduces the 3D problem to a regulation problem in 2D (Section 3.3).

3. Simulation is used to design and validate the controllers. A fit-for-purpose aircraft model structure is proposed which captures aircraft dynamics sufficiently to constrain the kinematic problem and obtain realistic results (Section 4.2). The aircraft model is shown to be identifiable from open-loop flight data with good (80+ %) fit[1].

4. Flight test results are presented for the PF controller (Section 5.2.4), demonstrating stable operation with control gains from simulation, thus providing additional validation of the modelling approach.

Chapter 2 provides a review of relevant literature and additional background information. Flight control theory as applicable to the PF and SA problems is presented in Chapter 3. Chapter 4 describes the modelling and simulation framework, as well as several simulation results for the PF and SA controllers. Flight test results are provided in Chapter 5, with a description of the UAV platform. Conclusions are made in Chapter 6 discussing applicability of these methods in other fields such as autonomous underwater vehicles (AUVs).

[1]Model fit is calculated from the identification data as: $\left(1 - \left|\frac{actual - predicted}{actual}\right|\right) \times 100\%$.

Chapter 2

Background & Related Work

A review of the literature shows that few academic papers have been written on the topic of using aircraft motion to achieve SA explicitly. However, in the growing field of UAV research, many airborne sensing applications are being developed for recreational, commercial, and academic use. Aircraft control and simulation, which are relatively well-developed fields for manned aircraft, are also seeing new and expanded research via UAVs. The literature reviewed as applicable to this thesis can be grouped according to the following sections: sensor aiming, path following, aircraft control methods, aircraft simulation, and UAV flight testing. Each area is discussed in the sections below.

2.1 Sensor Aiming

Most UAV flights can be considered data collection missions in which the overall track is chosen to aid the sensor in covering the data range of interest. In this sense, these UAV operations are intended to increase sensor coverage. Recently, multi-rotor

aircraft have rapidly gained popularity in both recreational and academic use. Most often these aircraft are used to obtain aerial photography, which can be interpreted as an example of sensor aiding wherein the camera FOV is extended by simply changing its position.

A quad-rotor aircraft with tilting motors that can achieve attitudes up to 90° is described in [41]. One intended application of their method is to enhance the field of view available to a camera mounted on the platform. Multi-rotor aircraft are normally under-actuated systems with four control inputs but operating in 6-DOF (degrees of freedom). For this reason, it is not possible to achieve all attitudes without inducing unwanted translational motion. A flight test of a quad-rotor with tilting propellers as a solution to the actuation problem is described in [43]. Using this method, they are able to achieve fully actuated control over the attitude without inducing translational motion. These specific applications require a specialized platform which effectively negates the benefits of gimbal-less operation, namely reduced payload and airframe simplification.

Trajectory planning for a UAV to complete coverage of a target using a fixed FOV sensor is described in [8]. Their approach focuses on trajectory planning to guarantee coverage within a minimum completion time. An interesting aspect is the consideration of aircraft thrust required to achieve climb rates. A 3-DOF gimbal is assumed for the sensor aiming. Another paper considering sensor coverage is [10], which considers groups of vehicles as mobile, tunable sensors. A theoretical approach is taken to guarantee sensor coverage, again assuming that a pan-tilt gimbal is available for sensor aiming.

An area related to SA is the concept of dynamic PF. Broadly speaking, these can

both be considered part of a class of problems where the UAV must remain aligned in a specified direction. Due to the sensor requirement to track and provide a path for the aircraft to follow, the target must normally be kept within the view of the sensor. A method of road following where the path is generated on-the-fly is described in [19]. The control is accomplished using a conventional attitude stabilization loop and a geometric approach to the PF problem. The camera is fixed with respect to the airframe, but consideration is not given to explicitly controlling aircraft attitude to keep the road in view. Rather, the strategy is to keep the roadway in view by minimizing aircraft bank angle, which is achieved by reducing controller gain, sacrificing performance. Road detection and tracking for a UAV is also described in [55]. A multi-rotor with gimbaled camera is used mainly to collect video data for post-processing, the methods of which are the primary focus. Again, explicit control aircraft attitude to assist data collection is not considered.

2.2 Path Following

PF is addressed in the literature for many applications of autonomy in robotics. Pioneering work in PF for the case of wheeled robots can be found in [37] and [44]. Fundamental guidance principles of path-following are addressed in [7]. Their paper uses a kinematics-based approach to address the PF problem and prove convergence for feasible paths.

A survey paper on PF methods for fixed-wing UAVs is conducted in [48]. The authors of the paper implement several PF methods and compare their performance in simulation. These range from the simplest case of waypoint following to more ad-

vanced optimal control-based and non-linear methods. Waypoint following had the worst performance in terms of minimizing path deviation, while the non-linear methods proved the best at reducing cross-track error. All algorithms considered in the survey paper are designed for 2D applications. The 2D limitation prohibits implementation of high-performance PF, as required, for example, in collision avoidance.

A UAV PF method in 3D is described in [29]. Similar to the method presented in this thesis, a coordinate frame tangent to the path is used to develop a control law to guide the aircraft. The results provided consider only large scale paths at a single altitude, which are essentially 2D. Also, the method intrinsically couples the PF kinematics with desired attitude, in contrast to the method developed in this thesis which decouples these two aspects of control. This may come at some cost to performance but offers advantages in modularity of design and ability to implement new behaviours. A PF algorithm for a fixed-wing UAV in wind is described in [2]. They explicitly consider roll and path angle constraints to demonstrate straight line and orbiting paths, for which the control laws are explicitly derived to guarantee convergence.

A PF method for a vertical take off and landing (VTOL) quad-rotor is described in [39]. Their paper describes a control method using attitude as an intermediate control variable, which is similar to the method used in this thesis. However, no specific application is discussed and the separation of attitude control is for design purposes.

Model predictive control (MPC) is used for UAV guidance by [20]. An aircraft model is used to predict the trajectory using a known guidance law and optimize the control parameters to improve path adherence. Their approach shows very good

results in simulation but is difficult to implement in practice. As such, no flight test results are provided. Similarly, MPC is used to track a linear trajectory in 2D by [30]. Results from hardware in-the-loop (HIL) simulation show good results in reducing path overshoot, but again flight test implementation is not done.

Path following using vector fields is addressed for wheeled non-holonomic robots in [31]. Their recent paper demonstrates promising results versus cross-track PF methods. An extension to 3D is suggested using the intersection of two planes which may be useful for UAV guidance.

2.3 Aircraft Control

Control system theory has been applied to aircraft in a variety of forms. An aircraft is a non-linear, coupled, multiple-input multiple-output (MIMO) system. Conventional controllers are based on proportional-integral-derivative (PID) loops and individual tuning of each loop, an effective method that is difficult to tune due to the number of coupled parameters that must be chosen.

A typical flight controller consists of an attitude stabilization inner loop that uses the ailerons, elevator, and rudder to regulate roll, pitch, and yaw angles, respectively. The outer loop in a conventional flight controller uses thrust, pitch, and a combination of bank angle and rudder to control altitude, airspeed, and horizontal turn rate, respectively. This is the approach used in [38].

A survey of advanced flight control theory and application is conducted in [56]. One modern design approach is linear quadratic optimal control [34], which is well known in control theory. This method seeks to minimize a quadratic cost function

with a set of linear differential equations. This method has been applied in the aerospace industry to develop control systems for several aircraft. To counter disturbances and modelling inaccuracy encountered in flight controller design, several methods of robust control design are applied by engineers. These methods include H_∞ control and μ-Synthesis [15] and are used to guarantee stability and performance throughout the entire flight envelope. Another non-linear controller design method is feedback linearization [47]. This method applies a linear transformation of the control input to obtain a linear mapping between the input and output. Related to feedback linearization is the application of quantitative feedback theory [24] which considers uncertainty in the plant model with quantitative performance requirements in controller design. These methods were used in development of control systems for production aircraft until adaptive control began to be used. Adaptive refers to the ability to allow for the controller parameters to adjust according to the current operating state of the plant, reducing the need for gain scheduling [25].

Dynamic inversion [46] is one of the feedback linearization approaches that is widely used in aircraft control. It is particularly useful for controller design when the operating condition is highly non-linear. Design of a full envelope decoupled linear flight controller using this method is described in [45]. They utilize several of the discussed aspects of controller design to achieve the full result. This is indicative of the piece-wise approach often taken in aircraft controller design. Due to the complex nature of the dynamics it is difficult to design a full controller based on a single design approach, but a combination of methods can be applied to adapt the controller to the desired flight envelope such as [1].

Most of these modern controllers are intended to reduce the number of param-

eters required for manual tuning and allowing for performance specification. These approaches to flight controller design tightly couple the attitude and overall velocity vector control. In contrast, the aircraft attitude can be controlled separately. This is addressed for a quad-rotor aircraft in [49], for example. The attitude control problem is approached explicitly for spacecraft applications in [53], from which the attitude control law used in this thesis builds upon.

2.4 Aircraft Simulation

Development of an accurate aircraft model is required both for control system development and for validation and testing of the developed system. Like control design, many methods and models exist for aircraft simulation. While some models are proprietary, open-source models such as JSBsim have received widespread acceptance. However, models like JSBsim are complex, cumbersome, and contain many features unnecessary for controller design. Several commercial and open source simulators for unmanned aircraft are surveyed in [11], finding that it is no longer necessary to develop a simulator from scratch and resources can instead be put towards other research efforts. Despite this conclusion, it may still be necessary to develop a simplified aircraft model from scratch or derive one from a more complex model to design the required control relationships to achieve the specific requirements of the proposed problem. The aircraft equations of motion (EOM) are derived using a conceptual approach in [54], which is helpful in understanding the coupling in terms of aerodynamics and deriving an appropriate control relationship.

A modelling approach for lift similar to that used in this thesis is described in [16].

Linearized dynamics are used in the longitudinal axes to parameterize angle of attack characteristics such that system identification methods can be applied. This thesis uses a similar approach but without considering angle of attack directly, instead using lift acceleration and body frame vertical velocity.

A survey paper on system identification methods for UAVs provides a clear overview of the process in [23]. System identification for an Ultra Stik model aircraft using frequency domain methods is given in [14]. The applicability of the method is demonstrated for control systems research.

A comprehensive textbook on aircraft mechanics [42] describes the development of an aircraft simulator based on 6-DOF EOM. The basic method involves using a set of linearized aerodynamics which are used in conjunction with the kinematic EOM as the basis of the simulator. This textbook method is only applicable to small deviations from the trim condition for which the linearized aerodynamic assumptions hold. The parameterization of the kinematic EOM has much theoretical overlap with attitude state observer design.

2.5 UAV Flight Testing

Several academic papers have been published regarding the development of UAV platforms. Many of the fixed-wing UAVs are a decade old and most contemporary research is being done using multi-rotor aircraft, however some fixed-wing aircraft platforms continue to be developed.

A description of a pioneering fixed-wing UAV platform is given in [17]. The airframe is purpose-built using a conventional control surface arrangement. The platform

is equipped with a high-end inertial measurement unit (IMU) and custom software, now obsolete. Another early UAV platform is the Georgia Tech GTmax described in a conference paper [26]. The UAV is a large conventional helicopter design based on an industrial helicopter with a high payload capacity and several sensor packages. A real-time operating system (RTOS) is used to implement the flight software.

A small fixed-wing UAV platform is described in [3]. Again, a purpose-built airframe is used with custom software for the autopilot functions. The paper presents AHRS results but without an independent measurement.

A UAV platform for evaluating loss of control accidents in commercial flight is described in [27]. The airframe is a 5.5% scale model of a generic airliner. Their system is extensively designed to provide a platform for scale model-based testing.

The Berkeley UAV software architecture and platform are described in [50]. Their platform is similar to that described in the thesis as it is an off-the-shelf (OTS) airframe using a commercially available autopilot. Custom RTOS software was created to augment the autopilot capability. A more recent paper describing a fully autonomous multi-rotor UAV is in [52]. Their approach follows the trend in using OTS modular multi-rotor platforms with customized autopilot and sensor packages.

A more recent development of a fixed-wing UAV platform is described in [4]. Their approach uses a modified Pixhawk autopilot and a hobby-grade airframe which is similar to the UAV platform developed in this thesis. However, their chosen airframe is much lighter and less manoeuvrable than this aircraft, due to their application which is harvesting wind energy using an aircraft tether.

Chapter 3

Flight Control

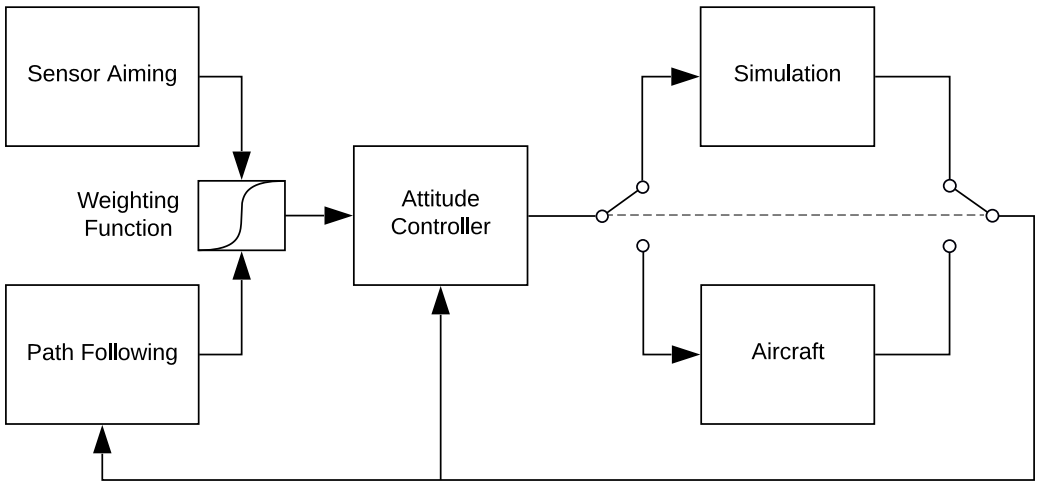

Figure 3.1: Flight control system block diagram.

In this chapter, the theoretical background is presented for the development of the PF and SA controllers. Figure 3.1 shows a block diagram of the complete flight control system. First, several preliminary concepts relevant to the controller development are presented, including coordinate frames, rotation representations, and state

observation. The PF controller theory is then presented. Finally, SA is defined and the integration with PF is described.

3.1 Contributions

The SA and PF controllers (Sections 3.4 & 3.3, respectively) constitute original work in concept of operation and implementation, including the time-varying weighting function used to integrate the control objectives (Section 3.4.1). A combination of theory involving rotations, quaternions, geometry, and kinematics is applied to create a solution to the proposed problem. The attitude control component of PF and AHRS (Section 3.2.5) are adaptations based on the work of others, referenced appropriately.

3.2 Preliminaries

This section introduces the coordinate frames, notation conventions, and background for uncommon mathematical concepts. Much of the notation that follows is borrowed from [18].

3.2.1 Coordinate frames

There are three distinct coordinate frames of interest:

- Earth-fixed, or North-East-Down (NED), frame $\{n\}$
- Path-fixed frame $\{p\}$
- Body-fixed frame $\{b\}$

These coordinate frames are illustrated in Figure 3.2.

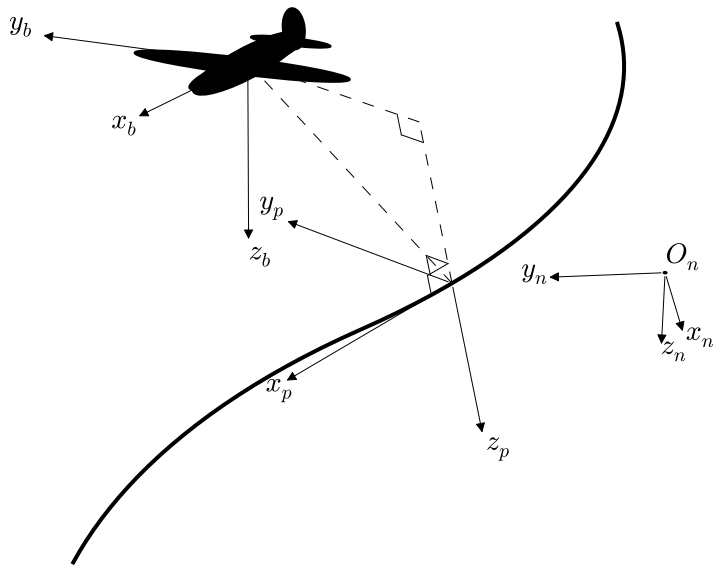

Figure 3.2: Diagram illustrating the relationship between coordinate frames.

The Earth-fixed frame $\{n\}$ has the origin fixed at a point on the Earth's surface designated as zero. It is assumed that this is an inertial reference frame and the curvature of the Earth is insignificant given the magnitude of motions and operational range. The path-fixed frame $\{p\}$ is a moving frame with an origin defined at the point along the path perpendicularly nearest to the current position of the aircraft. It is aligned with the x_p-axis coincident with the forward tangent of the path at the current index, y_p-axis parallel to the horizontal plane of the Earth-fixed coordinate frame, and z_p-axis downward, forming a right hand coordinate system. The body frame $\{b\}$ has its origin at the centre of gravity (CG) of the aircraft, with the x_b-axis aligned forward along the fuselage within the plane of symmetry, y_b-axis normal to the plane of symmetry in the direction of the right wing, and z_b-axis aligned downward

vertical within the plane of symmetry, forming a right hand coordinate system.

3.2.2 Quaternions vs. Euler angles

Aircraft attitude, or orientation, is normally described in terms of Euler angles[1], which use three angles to describe a rotation in three-dimensional space. Specifically, Euler angles using the zyx order of rotations can be represented as $\boldsymbol{\Theta} = [\phi, \theta, \psi]^\top \in \mathcal{S}^3$ where $\mathcal{S}^3$ is the set comprising a sphere in three dimensions. Euler angles, as described, are undefined at $\theta = \pm 90°$. This singularity, known as *gimbal lock*, is an inherent drawback of a minimum parameterization of attitude [33]. This issue is solved by inclusion of at least one additional redundant parameter to define attitude. Redundancy introduces another drawback in that the parameters are now constrained, i.e. no longer independent. However, there is no alternative if a universal attitude representation is required, as is normally the case with aircraft.

Quaternions are a four parameter extension of complex numbers first introduced by Hamilton in 1843 [21]. Quaternions obey a fundamental algebra that makes their use both elegant and practical as a parameterization of rotation. Quaternions are defined in matrix form as:

$$\mathbf{q} = [\eta, \boldsymbol{\varepsilon}^\top]^\top, \quad \boldsymbol{\varepsilon} = [\varepsilon_0, \varepsilon_1, \varepsilon_2]^\top \tag{3.1}$$

where $\eta \in \mathbb{R}$ is the scalar part and $\boldsymbol{\varepsilon} \in \mathbb{R}^3$ is the vector part of the quaternion. For rotations, these parameters are related to an axis-angle representation. A unit

[1]Proper Euler angles involve two non-consecutive rotations about the same axis, whereas the aerospace convention follows zyx order, termed Tait-Bryan angles. The term Euler angles can still be used for familiarity if the order of rotations is specified.

quaternion $\mathbf{q}$ describes a rotation β about an axis $\boldsymbol{\gamma}$ as:

$$\mathbf{q} = [\cos(\frac{\beta}{2}), \boldsymbol{\gamma}^\top \sin(\frac{\beta}{2})]^\top. \tag{3.2}$$

Rotation operations used extensively in this thesis rely on quaternion multiplication, normalization, and inversion as described in [9]. Quaternion multiplication is non-commutative and defined as:

$$\mathbf{q}_{21} = \mathbf{q}_2 \otimes \mathbf{q}_1 = \begin{bmatrix} \eta_2\eta_1 - \boldsymbol{\varepsilon}_2^\top \boldsymbol{\varepsilon}_1 \\ \boldsymbol{\varepsilon}_2\eta_1 + (\eta_2 \boldsymbol{I}_{3\times 3} + [\boldsymbol{\varepsilon}_2\times])\boldsymbol{\varepsilon}_1 \end{bmatrix} \tag{3.3}$$

where $\otimes$ is the quaternion multiplication operator and $[\boldsymbol{\varepsilon}\times]$ is the skew-symmetric matrix:

$$[\boldsymbol{\varepsilon}\times] = -[\boldsymbol{\varepsilon}\times]^\top = \begin{bmatrix} 0 & -\varepsilon_2 & \varepsilon_1 \\ \varepsilon_2 & 0 & -\varepsilon_0 \\ -\varepsilon_1 & \varepsilon_0 & 0 \end{bmatrix}. \tag{3.4}$$

Quaternion normalization, or norm, is defined as $\|\mathbf{q}\| = \sqrt{\eta^2 + \varepsilon_0^2 + \varepsilon_1^2 + \varepsilon_2^2}$. When describing rotations, quaternions are constrained with $\|\mathbf{q}\| = 1$. The inverse of a quaternion is defined as $\mathbf{q}^{-1} = \mathbf{q}^*/\|\mathbf{q}\|$ where $\mathbf{q}^*$ is the conjugate of the quaternion $\mathbf{q}^* = [\eta, -\boldsymbol{\varepsilon}^\top]^\top$. In the unit norm case, the inverse is equal to the conjugate of the quaternion.

The kinematics of the quaternion are defined by:

$$\dot{\mathbf{q}} = \frac{1}{2}\boldsymbol{W}\mathbf{q} \tag{3.5}$$

where

$$\boldsymbol{W} = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \tag{3.6}$$

and ω_x, ω_y, and ω_z are the components of $\boldsymbol{\omega}$.

If the orientation of the rate vector, $\boldsymbol{\omega}$, remains fixed over the update interval, the above equation can be discretized using the matrix exponential as:

$$\mathbf{q}_k = \exp\left[\frac{1}{2}\int_{t_{k-1}}^{t_k} \boldsymbol{W}\,\mathrm{d}t\right]\mathbf{q}_{k-1} \tag{3.7}$$

and the integral can be rewritten as:

$$\int_{t_{k-1}}^{t_k} \boldsymbol{W}\,\mathrm{d}t = \boldsymbol{\Sigma} = \begin{bmatrix} 0 & -\sigma_x & -\sigma_y & -\sigma_z \\ \sigma_x & 0 & \sigma_z & -\sigma_y \\ \sigma_y & -\sigma_z & 0 & \sigma_x \\ \sigma_z & \sigma_y & -\sigma_x & 0 \end{bmatrix} \tag{3.8}$$

where $\boldsymbol{\sigma} = \Delta t\boldsymbol{\omega}$, Δt being the discretization time interval, which gives

$$\mathbf{q}_k = \exp\left(\frac{\boldsymbol{\Sigma}}{2}\right)\mathbf{q}_{k-1}. \tag{3.9}$$

While quaternions are used for most internal calculation, Euler angles are used when presenting data and defining small relative rotations. Euler angles are preferred versus quaternions in these cases because of their familiarity and independent nature.

3.2.3 Vector quantities

The kinematic state of the aircraft is described by the following vectors:

$$\boldsymbol{\eta} = \begin{bmatrix} \boldsymbol{p}^n_{b/n} \\ \mathbf{q}_{nb} \end{bmatrix}, \quad \boldsymbol{\nu} = \begin{bmatrix} \boldsymbol{v}^b_{b/n} \\ \boldsymbol{\omega}^b_{b/n} \end{bmatrix} \tag{3.10}$$

where $\boldsymbol{p}^n_{b/n} = [x_n, y_n, z_n]^\top \in \mathbb{R}^3$ refers to the position of $\{b\}$ with respect to $\{n\}$ represented in $\{n\}$ and $\mathbb{R}^3$ is the Euclidean space in three dimensions, $\mathbf{q}_{nb} = [\eta, \varepsilon_0, \varepsilon_1, \varepsilon_2]^\top \in \mathbb{Q}$ indicates the quaternion representation of a rotation from $\{n\}$ to $\{b\}$ and $\mathbb{Q}$ is the set of unit quaternions, and $\boldsymbol{\nu} = [u, v, w, p, q, r]^\top \in \mathbb{R}^6$ represents the linear and angular velocities in the body-fixed frame of reference.

3.2.4 Rotations

Transformations between coordinate frames $\{b\}$ to $\{n\}$ are fundamentally achieved using the rotation matrix $\boldsymbol{R}^n_b \in \mathrm{SO}(3)$:

$$\boldsymbol{v}^n_{b/n} = \boldsymbol{R}^n_b(\mathbf{q}_{nb})\boldsymbol{v}^b_{b/n} \tag{3.11}$$

where $\boldsymbol{R}_b^n$ represents the rotation matrix from $\{b\}$ to $\{n\}$ and SO(3) refers to the *special orthogonal group* in three dimensions. This implies the properties $\boldsymbol{R} \in \mathbb{R}^3$, $\boldsymbol{R}$ is orthogonal, and $\det \boldsymbol{R} = 1$. The properties of the rotation matrix are such that the inverse rotation is equal to the transpose. Therefore, the rotation matrix from $\{n\}$ to $\{b\}$ is $\boldsymbol{R}_n^b = \boldsymbol{R}_b^{n\top}$. The rotation matrix is composed from the quaternion state parameters by:

$$\boldsymbol{R}_b^n(\mathbf{q}_{nb}) = \boldsymbol{I}_{3\times 3} + 2\eta[\boldsymbol{\varepsilon}\times] + [\boldsymbol{\varepsilon}\times]^2. \tag{3.12}$$

A compound rotation can be decomposed as a quaternion multiplication: $\mathbf{q}_{21} = \mathbf{q}_2 \otimes \mathbf{q}_1$ where $\mathbf{q}_{21}$ is rotation $\mathbf{q}_1$ followed by rotation $\mathbf{q}_2$. The difference between rotations $\mathbf{q}_{21}$ and $\mathbf{q}_1$ is $\mathbf{q}_2 = \mathbf{q}_{21} \otimes \mathbf{q}_1^{-1}$.

3.2.5 State Estimation

A fundamental aspect of flight control involves state estimation. Knowledge of the aircraft attitude is crucial to successful control, but presents a significant challenge due to the absence of an accurate absolute measurement. As a solution, a filtered solution for attitude estimation, or AHRS, is normally employed to combine multiple measurement sources. As covered in Chapter 2, this area has been widely researched.

Most attitude filters employ some form of Kalman filter [28] to combine measurements from multiple sources to create an attitude estimate. A non-linear implementation of the Kalman filter is required to accommodate the attitude kinematics. The

standard is the extended Kalman filter (EKF) which uses a linearization of dynamics about the current state estimate in its calculation [5].

A fundamental problem exists between the choice of attitude representation and Kalman filter solution [12]. The Kalman filter requires unconstrained variables to operate, a condition only met by minimum parameterizations of attitude such as Euler angles, which suffer from singularity of representation. Constraints imposed by redundant parameterizations such as quaternions are not considered by the Kalman filter equations. This problem is addressed and a work-around solution termed the multiplicative extended Kalman filter (MEKF) is presented in [35].

The MEKF estimates the attitude error between the measurement and the true attitude using a minimum unconstrained parameterization of attitude. The true attitude is then equal to the quaternion product of the error quaternion and a reference quaternion:

$$\mathbf{q} = \delta\mathbf{q}(\boldsymbol{a}) \otimes \mathbf{q}_{\text{ref}}. \tag{3.13}$$

The state estimation variable $\boldsymbol{a}$ can be chosen as the Euler angle representation of the rotation between the reference quaternion and the true attitude $\mathbf{q}$. The MEKF equations are now presented for clarity and completeness.

The filter is initialized with a state covariance $\boldsymbol{P} \in \mathbb{R}^{6\times6}$, process noise covariance $\boldsymbol{Q} \in \mathbb{R}^{6\times6}$, and measurement noise covariance $\boldsymbol{R} \in \mathbb{R}^{6\times6}$, assuming two vector measurements are made (normally gravity and magnetic north). The state vector is defined as $\boldsymbol{x} \equiv [\boldsymbol{a}^\top, \boldsymbol{\mu}_\omega^\top]^\top \in \mathbb{R}^6$ where $\boldsymbol{\mu}_\omega$ is the angular rate sensor bias vector as

described by the sensor model in Equation 4.4. For convenience, the state covariance matrix can be partitioned into 3×3 attitude, correlation, and bias matrices as:

$$\boldsymbol{P} \equiv \begin{bmatrix} \boldsymbol{P}_a & \boldsymbol{P}_c \\ \boldsymbol{P}_c^\top & \boldsymbol{P}_b \end{bmatrix}. \tag{3.14}$$

The MEKF procedure begins by predicting the state using the angular rate measurements. Equation 3.9 is used with $\boldsymbol{\omega}_{\text{ref}} = \tilde{\boldsymbol{\omega}}_{b/n}^b - \boldsymbol{\mu}_\omega$ as the angular rate input to update the reference attitude $\mathbf{q}_{\text{ref}}$. The state covariance matrix $\boldsymbol{P}$ is then predicted in discrete time using:

$$\boldsymbol{P}_k^- = \boldsymbol{P}_{k-1} + \Delta t\big(\boldsymbol{F}\boldsymbol{P} + \boldsymbol{P}\boldsymbol{F}^\top + \boldsymbol{G}\boldsymbol{Q}\boldsymbol{G}^\top\big) \tag{3.15}$$

where the linearizations $\boldsymbol{F}$ and $\boldsymbol{G}$ are:

$$\boldsymbol{F}(t) \equiv \begin{bmatrix} \dfrac{\partial \boldsymbol{f}}{\partial \boldsymbol{a}} & \dfrac{\partial \boldsymbol{f}}{\partial \boldsymbol{\mu}_\omega} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix} = \begin{bmatrix} -[\boldsymbol{\omega}_{\text{ref}}\times] & -\boldsymbol{I}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \end{bmatrix} \tag{3.16}$$

and

$$\boldsymbol{G}(t) \equiv \begin{bmatrix} \dfrac{\partial \boldsymbol{f}}{\partial \boldsymbol{w}_1} & \dfrac{\partial \boldsymbol{f}}{\partial \boldsymbol{w}_2} \\ \mathbf{0}_{3\times3} & \boldsymbol{I}_{3\times3} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{I}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \boldsymbol{I}_{3\times3} \end{bmatrix}, \tag{3.17}$$

where $\boldsymbol{f}(\boldsymbol{x}, t) = \tilde{\boldsymbol{\mu}}_\omega - \boldsymbol{\mu}_\omega - \boldsymbol{w}_1 - \boldsymbol{\omega}_{\text{ref}} \times \boldsymbol{a}$.

The Kalman gain is calculated using:

$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{P}_a^- \\ \boldsymbol{P}_c^{-\top} \end{bmatrix} \boldsymbol{H}_a^\top \left(\boldsymbol{H}_a \boldsymbol{P}_a^- \boldsymbol{H}_a^\top + \boldsymbol{R} \right)^{-1} \tag{3.18}$$

where $\boldsymbol{H}_a \equiv \left.\dfrac{\partial \boldsymbol{h}}{\partial \boldsymbol{z}}\right|_{\hat{\boldsymbol{z}}^b_{b/n}} [\hat{\boldsymbol{z}}^b_{b/n}\times]$ is a linearization of the measurement function[2] and $\boldsymbol{z}$ is a concatenation of vector measurements representing an absolute observation of gravity and magnetic north in the body frame. Since the vector measurements are made directly, $\left.\dfrac{\partial \boldsymbol{h}}{\partial \boldsymbol{z}}\right|_{\hat{\boldsymbol{z}}^b_{b/n}} = \boldsymbol{I}_{3\times 3}$. The superscript $^-$ indicates the *a priori* estimates before the measurement update occurs.

The state estimate and state covariance matrix are updated using:

$$\hat{\boldsymbol{x}}^+ = \hat{\boldsymbol{x}}^- + \boldsymbol{K}\left(\tilde{\boldsymbol{z}}^b_{b/n} - \boldsymbol{h}(\hat{\boldsymbol{z}}^b_{b/n}) - \boldsymbol{H}_a \hat{\boldsymbol{a}}^- \right) \tag{3.19}$$

and

$$\boldsymbol{P}^+ = \boldsymbol{P}^- - \boldsymbol{K}\boldsymbol{H}_a[\boldsymbol{P}_a^-, \boldsymbol{P}_c^-] \tag{3.20}$$

where the superscript $^+$ indicates *a posteriori* estimates proceeding from the inclusion of the observation.

Finally, the error estimate state information is transferred to the reference quaternion using Equation 3.13 and the error state $\boldsymbol{a}$ explicitly reset to zero before moving on to the next iteration.

[2]The portion of the measurement sensitivity matrix relating to the bias estimates is zero because the measurement does not depend explicitly on these biases.

3.3 Path Following

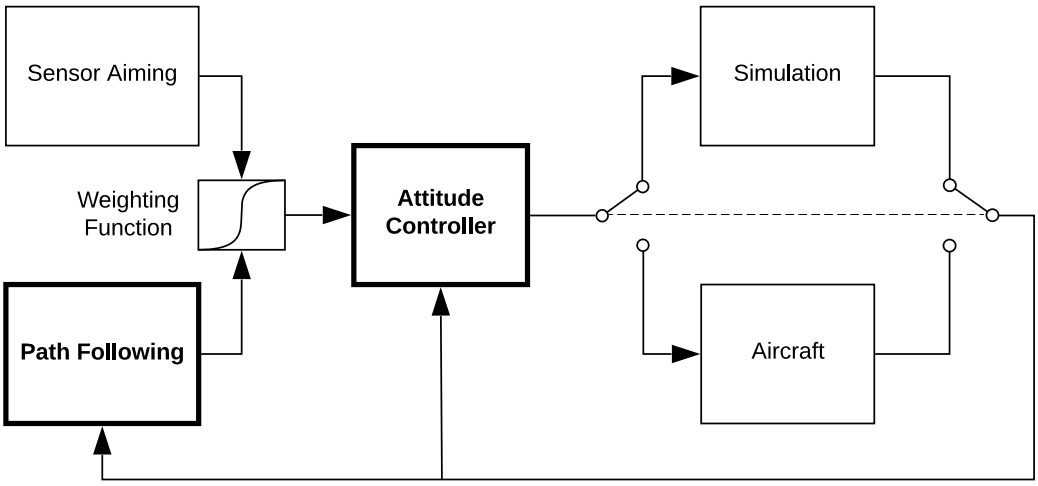

Figure 3.3: PF component of flight control system block diagram.

The objective of the PF controller is to guide the aircraft along a prescribed path in space. Figure 3.3 highlights the PF components of the overall flight control system block diagram. The PF control strategy can be divided into four steps:

1. Locating the current index k of position along the desired path.

2. Calculation of relative position of the aircraft in a plane orthogonal to the path's tangent at the current index.

3. Mapping relative position to a desired attitude in a negative feedback arrangement, comprising the control law for the outer loop.

4. Mapping of desired attitude to control surface deflections, comprising the control law for the inner loop.

The current index k of aircraft position along the path is found using Algorithm[3] 1. This algorithm is called during each iteration of the control loop to update the current progress along the path. It compares the Euclidean distance between the current step of the path vector and the next, updating the path index and previous distance if less. The algorithm considers only a local segment of the path for each update to avoid problems due to overlapping paths.

Algorithm 1 Locating the index along the path vector.

```
prev_dist ← ‖p^n_{b/n} − p^n_{d/n}[k]‖
while prev_dist > ‖p^n_{b/n} − p^n_{d/n}[k + 1]‖ do
    prev_dist ← ‖p^n_{b/n} − p^n_{d/n}[k + 1]‖
    k ← k + 1
end while
```

Using the current index, relative position within the path frame is found using:

$$\boldsymbol{p}^p_{b/p} = \boldsymbol{R}^n_p(\mathbf{q}_{np}[k])^\top(\boldsymbol{p}^n_{b/n} - \boldsymbol{p}^n_{p/n}[k]) \tag{3.21}$$

where $\boldsymbol{R}^n_p$ indicates the rotation matrix from path-fixed frame to Earth-fixed frame and $\mathbf{q}_{np}$ indicates the quaternion representation of rotation from Earth-fixed to path-fixed frames.

Desired attitude with respect to the path using an Euler angle parameterization is calculated using:

$$\boldsymbol{\Theta}_{pd} = K_{\mathrm{PF}}\boldsymbol{p}^p_{b/p} \tag{3.22}$$

[3]In this algorithm and throughout this thesis, $\|\boldsymbol{p}\|$ refers to the Euclidean norm $\sqrt{a_x^2 + a_y^2 + a_z^2}$, where a_x, a_y, a_z are the elements of a vector $\boldsymbol{p}$.

where the gain matrix $K_{\mathrm{PF}} \in \mathbb{R}^{3\times3}$ relates each axis of relative position to a corresponding relative attitude in three axes.

Desired attitude $\boldsymbol{\Theta}_{pd}$ is relative to the path attitude $\mathbf{q}_{np}$ and is with respect to the Earth-fixed frame using:

$$\mathbf{q}_{nd} = \mathbf{q}_{np}[k] \otimes \mathbf{q}_{pd} \tag{3.23}$$

where $\mathbf{q}_{pd} = \mathbf{q}(\boldsymbol{\Theta}_{pd})$.

The inner loop attitude controller [53] torques are found using:

$$\tilde{\mathbf{q}} = \mathbf{q}_{nb}^{-1} \otimes \mathbf{q}_{nd} \tag{3.24}$$

and

$$\tilde{\boldsymbol{\tau}} = k_p \tilde{\boldsymbol{\varepsilon}} + k_v \boldsymbol{\omega}_{b/n}^{b} \tag{3.25}$$

where $\tilde{\mathbf{q}} = [\tilde{\eta}, \tilde{\boldsymbol{\varepsilon}}]$ and $\tilde{\boldsymbol{\tau}}$ is the vector of body frame torques.

The control surface deflections are allocated from the torques using the inversion of the rotational model input matrix B_r^{-1} from Equation 4.1:

$$\boldsymbol{\delta} = k_b B_r^{-1} \tilde{\boldsymbol{\tau}}. \tag{3.26}$$

Constants k_p, k_v, and k_b are proportional, rate damping, and overall gain parameters, respectively. The values are determined using conventional PD tuning methods: $k_p = 3$, $k_v = 1$, and $k_b = 15$.

3.4 Sensor Aiming

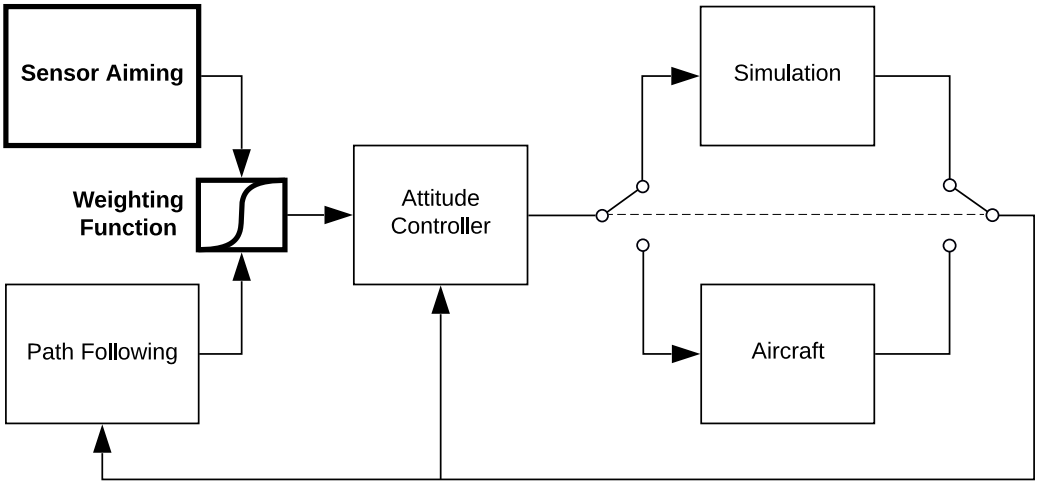

Figure 3.4: SA component of flight control system block diagram.

The objective of SA is to bring the aircraft into a specified orientation (or attitude) such that a sensor measurement can be made. Figure 3.4 contains a block diagram highlighting the SA component of the flight control system. The two main aspects of the SA controller involve a time-varying weighting function used to transition from PF to SA and the generation of setpoints defining each SA event.

3.4.1 Time-Varying Weighting Function

Integration of the SA setpoint with the PF controller is accomplished by a time-varying weighting function. The weighting function is intended to blend the SA and PF setpoints such that the transition is smooth between modes and disruption of the flight is reduced. Therefore, the choice of function is arbitrary and can be used as a tuning parameter to affect performance. Figure 3.5 illustrates a typical logistic curve [40] that can be used to transition between modes.

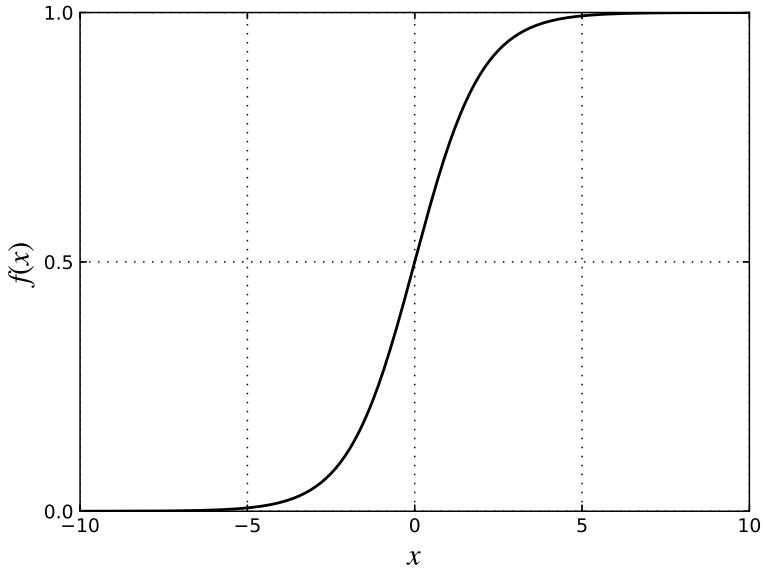

Figure 3.5: General normalized logistic curve.

The general equation for the normalized logistic function is:

$$f(x) = \left(1 + e^{-\kappa(x - x_0)}\right)^{-1} \tag{3.27}$$

where parameters κ and x_0 define the slope of the transition and midpoint bias, respectively. Therefore, the response can be shaped by selecting parameters to provide an appropriate transition. Time differences are used as the basis for the input to the logistic function as the primary factor in both SA duration and PF error. Position can be translated into time if the trajectory of the aircraft is known or can be estimated.

Let $\boldsymbol{\eta}_{t_0}$ be the state of the aircraft at time t_0. The SA objective *on approach* is to bring the state of the aircraft to $\boldsymbol{\eta}_{t_f}$ at time t_f. To minimize the path deviation, $\boldsymbol{\eta}_{t_0} \rightarrow \boldsymbol{\eta}_{t_f}$ shall be reachable within minimum duration $t_s = t_f - t_r$, where t_r is the transition time where full SA duration begins. Time t_s can be calculated under closed-loop control, with full control effort towards SA attitude.

The closed-loop attitude control and rotational dynamics are highly non-linear and therefore difficult to solve in closed form, so a deterministic simulation using the rotational aircraft model and attitude controller is used to compute t_s. A tolerance θ_t is applied to the scalar part of the quaternion difference between the current and desired attitude at t_f as a termination criterion. Time t_s defines the transition time $t_r \equiv t_f - t_s$ on approach for the weighting function. Once time t_f is reached, the setpoint is restored to PF *on return*. From this logic, the approaching weighting function can be written:

$$\Lambda_a(t) = \left(1 + e^{-\frac{1}{2}((t-t_r)^{-2}-10)}\right)^{-1}. \tag{3.28}$$

The transition point to return to PF is t_f, which is immediately after the SA objective has been achieved. Thus, the returning weighting function can be written:

$$\Lambda_r(t) = \left(1 + e^{-\frac{1}{2}((t-t_f)^{-2}-10)}\right)^{-1} \tag{3.29}$$

where t indicates the time of calculation, $1/2$ is a scale factor, and -10 is a midpoint bias used to favour PF when an SA event is not immediate. To make the transition duration sufficiently short, the inverse square difference of the times is used as the argument to the logistic function.

These time-varying weighting functions are used before and after the duration t_s of full SA. The weighting variable λ_s where $\{\lambda_s \in \mathbb{R} \mid 0 \leq \lambda_s \leq 1\}$ is thus assigned according to the logic outlined in Algorithm 2. This algorithm defines an overall non-linear, time-varying weighting function used to transition between PF and SA. That is, a different logistic function argument is used while approaching and returning from a SA event, which is characterized by the duration t_s. The SA event itself is characterized by a required aiming duration pre-determined by the aircraft model. An example of the time-varying weighting function in use is shown in Figure 4.19.

Algorithm 2 Assign SA weighting based on setpoint timing.

```
if t_f − t < t_r then
    λ_s ← 1
else if t < t_r then
    λ_s ← Λ_a(t)
else
    λ_s ← Λ_r(t)
end if
```

Due to the non-linear, intermittent nature of the SA problem, a conventional optimization approach is not appropriate. An approach such as linear-quadratic reg-

ulation (LQR) would attempt to minimize some path deviation, while accommodating some sensor aiming, but resulting in the full pursuit of neither.

Attitude is the aspect of the state controlled directly by the inner loop controller. The weighting variable is used to blend the PF attitude with the SA attitude to avoid abrupt changes in setpoint. The resulting intermediate attitude $\mathbf{q}_u$ is used as the input to the inner loop and is determined as follows.

The quaternion difference between the SA attitude $\mathbf{q}_s$ and PF attitude $\mathbf{q}_{nd}$ is:

$$\mathbf{q}_{ds} = \mathbf{q}_s \otimes \mathbf{q}_{nd}^{-1} \tag{3.30}$$

which can be decomposed using Equation 3.2 to:

$$\mathbf{q}_{ds} = [\cos(\frac{\beta_{ds}}{2}), \boldsymbol{\gamma}_{ds}^{\top} \sin(\frac{\beta_{ds}}{2})]^{\top}. \tag{3.31}$$

According to Euler's rotation theorem [32], a rotation β_{ds} about an axis defined by vector $\boldsymbol{\gamma}_{ds}$ describes the difference between $\mathbf{q}_s$ and $\mathbf{q}_{nd}$. Scaling β_{ds} by λ_s provides a weighted blending of the two rotations that can be converted back to quaternion form and applied to the PF attitude. Therefore, the intermediate attitude is:

$$\mathbf{q}_u = [\cos(\frac{\beta_{ds}\lambda_s}{2}), \boldsymbol{\gamma}_{ds}^{\top} \sin(\frac{\beta_{ds}\lambda_s}{2})]^{\top} \otimes \mathbf{q}_{nd}. \tag{3.32}$$

3.4.2 Setpoint Generation

Each SA event is defined based on the specific requirements of the mission. Defining the SA setpoint involves at least five parameters: time, position, orientation, and the rates of change of position and orientation. The rates of change of position and orientation can be assumed to be small under automatic control. Therefore, in practice, two parameters can be used to define the setpoint: time (or position), and orientation. The events must be sufficiently decoupled by time segmentation such that PF can still be achieved. This spacing depends on the dynamics of the aircraft and the magnitude of the attitude change required for the SA event.

A typical UAV application may involve aerial surveillance of a fixed or moving point on the ground using a camera or other sensor. This aiming is normally achieved using a gimbal mounted underneath the aircraft fuselage. However, a gimbal-less solution is possible using the SA method described.

If a UAV were equipped with a fixed downward-facing sensor facing along the z_b axis of the aircraft, the sensor could be aimed by altering the aircraft attitude towards a target. For a general aircraft path, it is only intermittently possible to alter the attitude for SA. Therefore, the SA events are defined periodically. The derivation of the SA attitude for ground target tracking follows.

Euler angles are used to keep the following derivation intuitive. Euler angles are valid in this application because the attitude change will be less than 90°. The rotation matrix from Euler angles is [22]:

$$\boldsymbol{R}_b^n = \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \sin\psi\sin\phi + \cos\psi\cos\phi\sin\theta \\ \sin\psi\cos\theta & \cos\psi\cos\phi + \sin\phi\sin\theta\sin\psi & \sin\theta\sin\psi\cos\phi - \cos\psi\sin\phi \\ -\sin\theta & \cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \tag{3.33}$$

which can be used in the relationship between the global and body-relative position:

$$\boldsymbol{p}_{b/t}^b = \boldsymbol{R}_b^{n\top} \boldsymbol{p}_{b/t}^n, \tag{3.34}$$

where the subscript t indicates the target.

Letting $\boldsymbol{p}_{b/t}^b = [0, 0, \|\boldsymbol{p}_{b/t}^n\|]^\top$ and using the current heading for ψ, the pitch and roll angles are:

$$\begin{aligned} \theta &= \arctan\left(\frac{x\cos\psi + y\sin\psi}{z}\right) \\ \phi &= \arctan\left(\frac{x\sin\psi + y\cos\psi}{x\cos\psi\sin\theta + y\sin\psi\sin\theta + z\cos\theta}\right) \end{aligned}, \tag{3.35}$$

which can be converted back to quaternion form for inclusion as a SA event.

Chapter 4

Modelling & Simulation

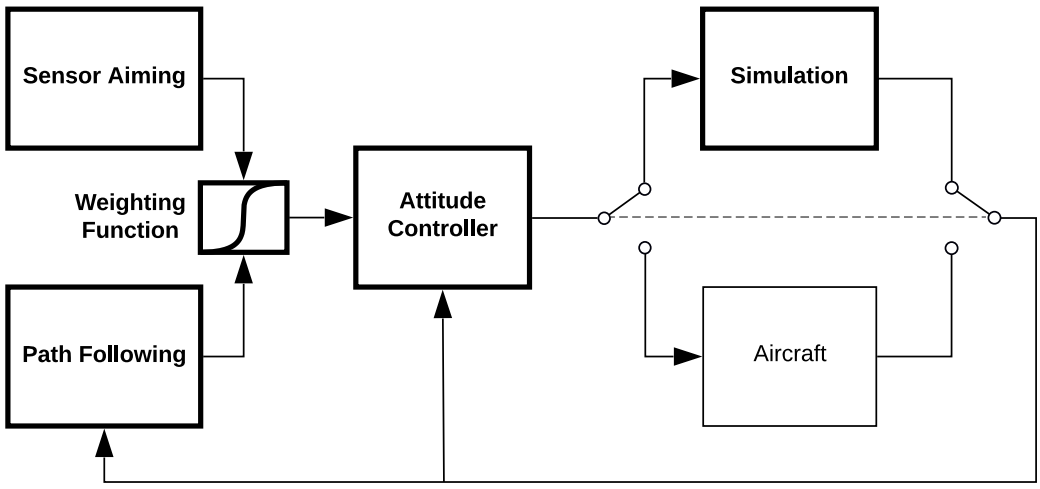

Figure 4.1: Simulation component of flight control system block diagram.

This chapter describes the aircraft modelling approach, simulation environment, and associated simulation results. Figure 4.1 contains the flight control system block diagram with the blocks used in simulation highlighted. System identification results are shown for the aircraft model from open loop test data. Aspects of simulation

design including assumptions, sensor modelling, and environmental modelling are discussed. Simulation results are presented demonstrating the PF and SA controllers. Two example applications are considered for SA: radar aiming for air-to-air target detection and gimbal-less ground target aiming.

4.1 Contributions

The aircraft modelling approach is developed specifically for the control system development in this thesis. This fit-for-purpose model is designed to be used within a simulation environment created to test the PF and SA controllers.

4.2 Aircraft Model

The modelling approach developed is intended to sufficiently capture the aircraft dynamics such that a control system can be accurately developed. Within this context, the model should capture dynamics essential to describe manoeuvring characteristics but not necessarily all aerodynamic effects considered in a traditional flight model. Additionally, the model structure must be amenable to application of system identification methods. This is due to use of live flight tests versus structured aerodynamic testing such as that conducted in a wind tunnel facility.

The model is separated into two parts: rotational and lift. The rotational model is intended to capture rotational dynamics in body frame and is based on first-order responses in each axis with cross-coupling parameters. This is a lumped parameterization capturing control surface response, moment generation, and rotational inertia.

The lift model is based on a second-order parameterization relating pitch rate and lift. The lift model assumes operation within the linear region of the lift curve. This approach is convenient as it avoids explicit calculation of angle of attack, which is difficult to measure in practice.

The structure of the rotational model is:

$$\begin{aligned}\dot{\boldsymbol{\omega}}^b_{b/n} &= A_r\boldsymbol{\omega}^b_{b/n} + B_r\boldsymbol{\delta} \\ &= \begin{bmatrix} a_{d_0} & 0 & a_{c_0} \\ a_{c_1} & a_{d_1} & a_{c_2} \\ a_{c_3} & 0 & a_{d_2} \end{bmatrix} \boldsymbol{\omega}^b_{b/n} + \begin{bmatrix} b_{d_0} & 0 & b_{c_0} \\ 0 & b_{d_1} & 0 \\ 0 & 0 & b_{d_2} \end{bmatrix} \boldsymbol{\delta}\end{aligned} \tag{4.1}$$

where some elements of the input matrix have been assumed zero based on physical layout of the airframe and longitudinal-lateral coupling and control surface arrangement [42].

The vector $\boldsymbol{\omega}^b_{b/n} = [p, q, r]^\top$ represents angular velocities in roll, pitch, and yaw, respectively, which are the outputs of the model. The vector $\boldsymbol{\delta} = [\delta_a, \delta_e, \delta_r]^\top$ represents control surface deflections of the ailerons, elevator, and rudder, respectively.

The lift model structure is:

$$\begin{aligned}\begin{bmatrix} \dot{Z} \\ \dot{w} \end{bmatrix} &= A_l \begin{bmatrix} Z \\ w \end{bmatrix} + B_l q \\ &= \begin{bmatrix} a_{z_0} & a_{z_1} \\ a_{z_2} & 0 \end{bmatrix} \begin{bmatrix} Z \\ w \end{bmatrix} + \begin{bmatrix} b_{z_0} \\ 0 \end{bmatrix} q\end{aligned} \tag{4.2}$$

where some parameters have been set to zero based on the assumption that body-frame vertical velocity is the integration of lift acceleration and there is no feed-forward from pitch rate.

The states in this model include the vertical lift force Z and vertical component of velocity w, both in body frame. The input is pitch rate q. This structure is based on the relationships between Z, angle of attack α, and q. In the linear range of the lift curve, $Z \propto \alpha$. The time rate of change of angle of attack $\dot{\alpha}$ can be approximated by q. Therefore, $\dot{Z}$ can be approximated by q scaled by a constant that can be defined as a system identification parameter. The second-order term w, the integration of Z, is used to capture the inherent damping due to lift.

4.2.1 System Identification

The model parameters are identified from flight test data using MATLAB system identification toolbox. The flight from which the data is taken is detailed in Section 5.2.2. The model structures are provided as grey-box models. This reduces the number of free parameters to be estimated, thereby reducing ambiguity in fit. The identified parameter values for the rotational model is summarized in Table 4.1. The diagonal parameters are estimated individually to provide initial values before identifying cross-coupling parameters. The identified parameters for the lift model are summarized in Table 4.2. All identified parameters are intended for a fixed forward speed of 25 ms^{-1}, which was the mean airspeed during the identification segments.

Figure 4.2 contains an excerpt of the validation data in roll. The output from the identified model follows the flight data closely (92.0% fit to estimation data).

Table 4.1: Identified parameters for rotational model.

Parameter	Value	Parameter	Value
a_{d_0}	-9.067	b_{d_0}	-2.406
a_{d_1}	-7.777	b_{d_1}	-1.068
a_{d_2}	-3.323	b_{d_2}	-0.8684
a_{c_0}	4.856	b_{c_0}	0.3442
a_{c_1}	0.3163		
a_{c_2}	-0.3101		
a_{c_3}	-1.201		

Table 4.2: Identified parameters for lift model.

Parameter	Value
a_{z_0}	-6.578
a_{z_1}	0.03833
a_{z_2}	2.667
b_{z_0}	-18.14

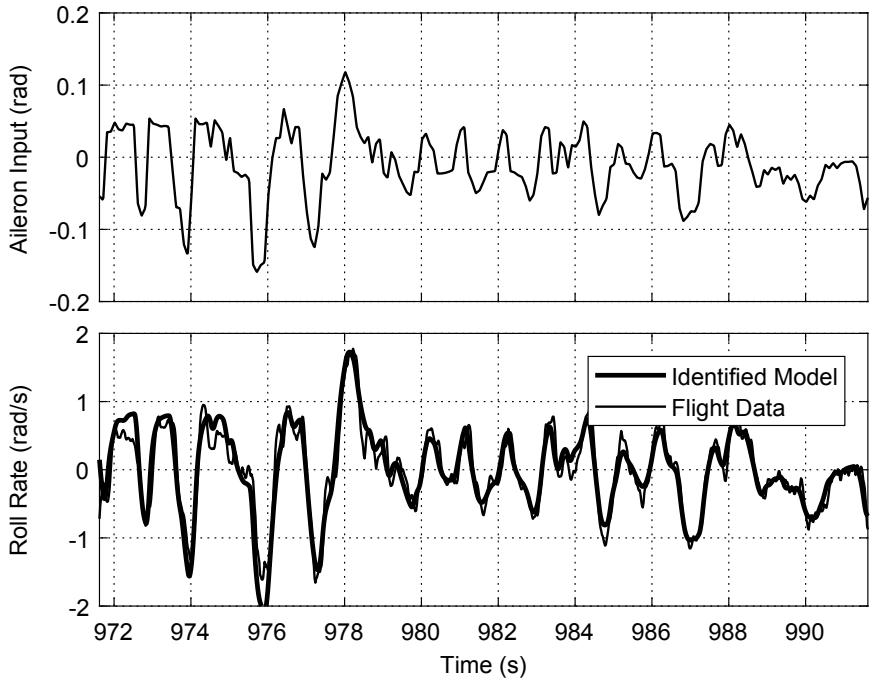

Figure 4.2: Flight data and single axis model output in roll.

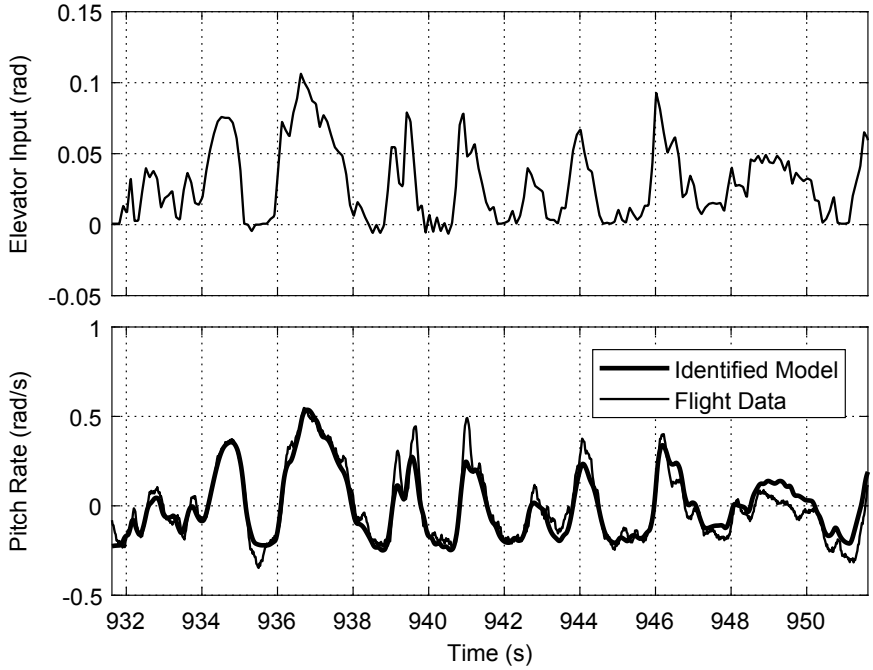

Figure 4.3: Flight data and single axis model output in pitch.

Figure 4.3 contains an excerpt of the validation data in pitch. Again, the single axis model in pitch matches the flight data closely (92.8% fit).

Validation data and yaw axis model output is shown in Figure 4.4. The yaw axis exhibits more coupling behaviour versus roll and pitch. Also, the single axis yaw model can only provide an output when a rudder input is supplied. During such times, the model output for yaw rate is reasonably accurate (84.1% fit).

The output of the three axis coupled model with another segment of validation data is shown in Figure 4.5. Model fits have improved in roll and pitch by including the identified coupling parameters. The yaw axis is more susceptible to wind disturbance and, during the data collection, winds were light and variable in direction. As such, there are additional un-modeled variations in yaw versus the other axes. However, the inclusion of the roll coupling parameter improves the overall model accuracy (93.8%

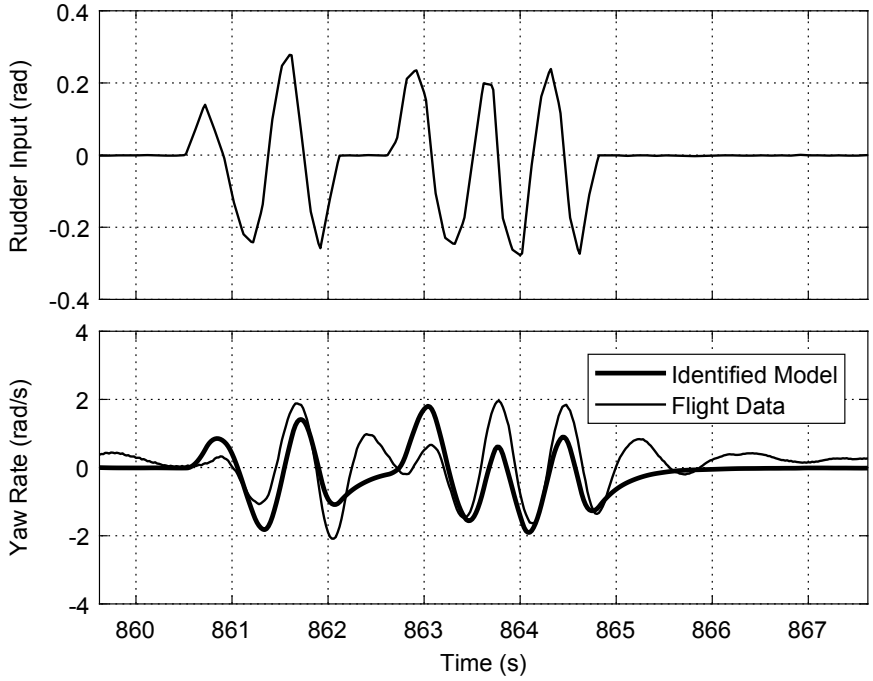

Figure 4.4: Flight data and single axis model output in yaw.

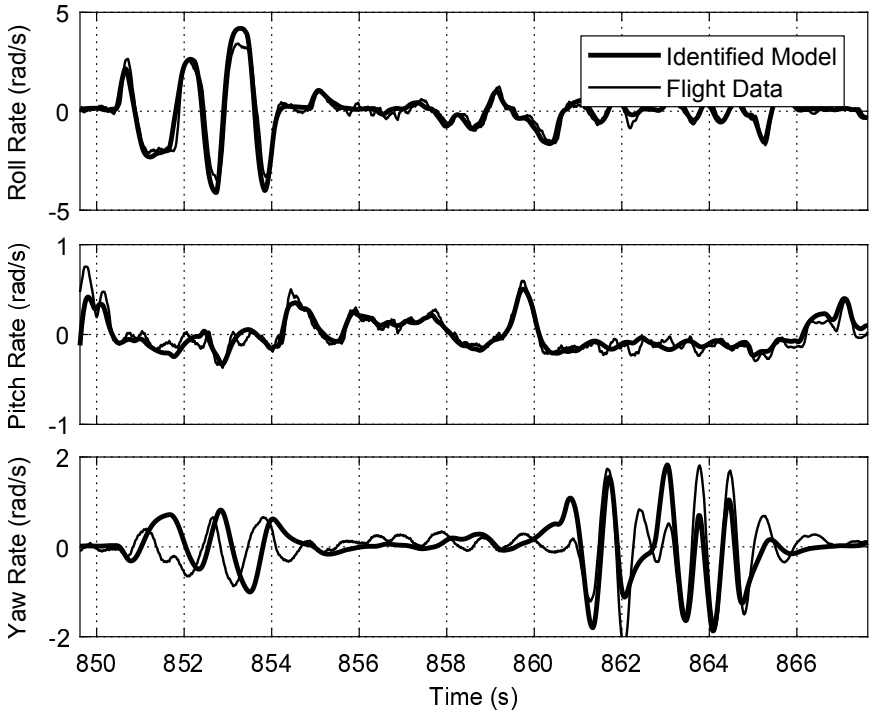

Figure 4.5: Flight data for validation segment containing all three rotational axes.

fit), particularly when no rudder input is being applied.

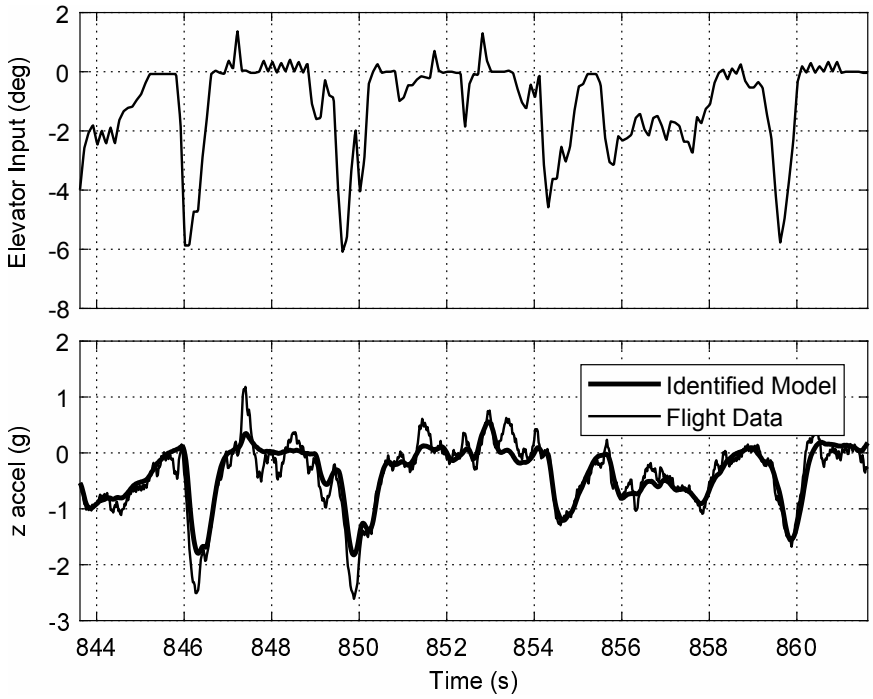

Figure 4.6: Validation flight data with lift model output.

The vertical acceleration output of the lift model during a segment of validation data is shown in Figure 4.6. As the vertical velocity in the body frame is explicitly calculated in simulation and, as such, is not relevant as an estimated quantity. Rather, the effect of this velocity on lift (i.e., damping) is required. Model estimate of vertical acceleration during the validation segment is reasonably accurate (80.8% fit). This model is used primarily to capture the dynamics of the banked turn.

4.3 Simulation

The simulation is non-linear 6-DOF and used for controller development using identified models for angular velocity and lift with a fixed forward speed. The aircraft is thus treated as a point mass subject to gravity and the following assumptions are made:

1. Rotational and lift dynamics are with respect to a fixed forward speed which can be maintained by appropriate control of thrust.

2. Rotational response due to control surface deflection is independent of attitude.

3. Side slip effects are sufficiently captured by the lumped-parameter model.

The primary intent of modelling lift is to capture horizontal motion when performing a banked turn and to penalize the system for attitudes that do not support proper lift versus gravity. Side slip effects are not explicitly modeled, however these effects are less significant for a straight shoulder wing aircraft as is considered throughout this thesis.

Attitude of the aircraft is stored as a quaternion $\mathbf{q}_{nb}$ and updated according to Equation 3.9 [51]. Position of the aircraft is stored as $\boldsymbol{p}^n_{b/n}$ and updated using:

$$\dot{\boldsymbol{p}}^n_{b/n} = \boldsymbol{R}^n_b(\boldsymbol{\Theta}_{nb})\boldsymbol{v}^b_{b/n}. \tag{4.3}$$

Figure 4.7 shows a block diagram of the simulation. The simulation equations are discretized and evaluated at a fixed time-step of $\Delta t = 1/50$ seconds. Similarly, the PF and SA controllers are implemented in the simulation in discrete time.

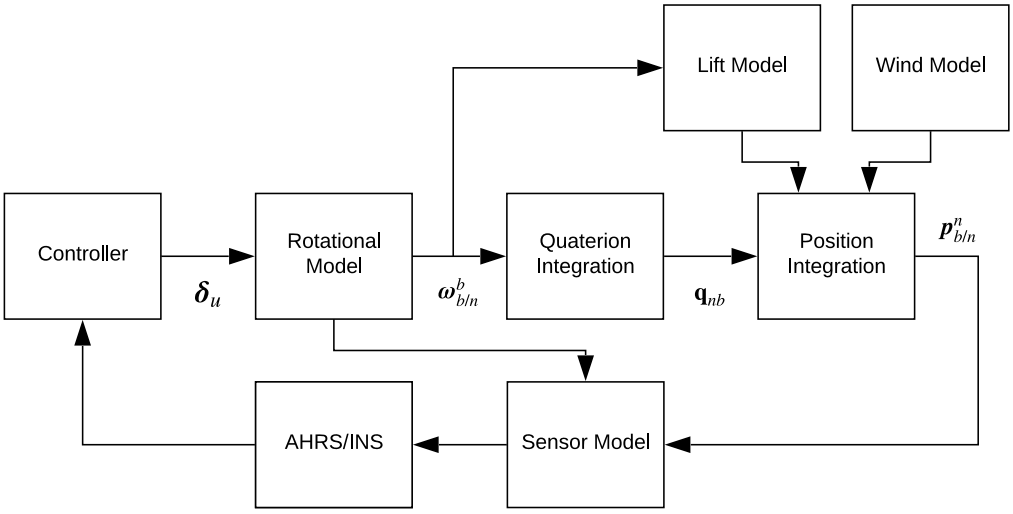

Figure 4.7: Simulation block diagram.

The simulation is implemented in Python using an object-oriented approach. The autopilot and AHRS are implemented as objects to manage their own states. The autopilot class receives the aircraft state estimate from the AHRS which is analogous to reality. A main simulation loop calls the autopilot with the current path information to determine the required control surface deflections. SA operates in parallel to the PF control and the attitude weighting changes based on the SA event timing. Figure 4.8 contains a software sequence diagram describing the implementation of the simulation.

Each simulation begins by defining the pre-determined path for the given scenario. The aircraft state is initialized at the beginning of the path and a controller and AHRS instance is created. At each time-step, simulation gets the most recent control surface deflections from the controller instance and updates the aircraft state according to the previously described simulation equations. The simulation continues until either

the maximum duration has elapsed or the end of the pre-determined path is reached.

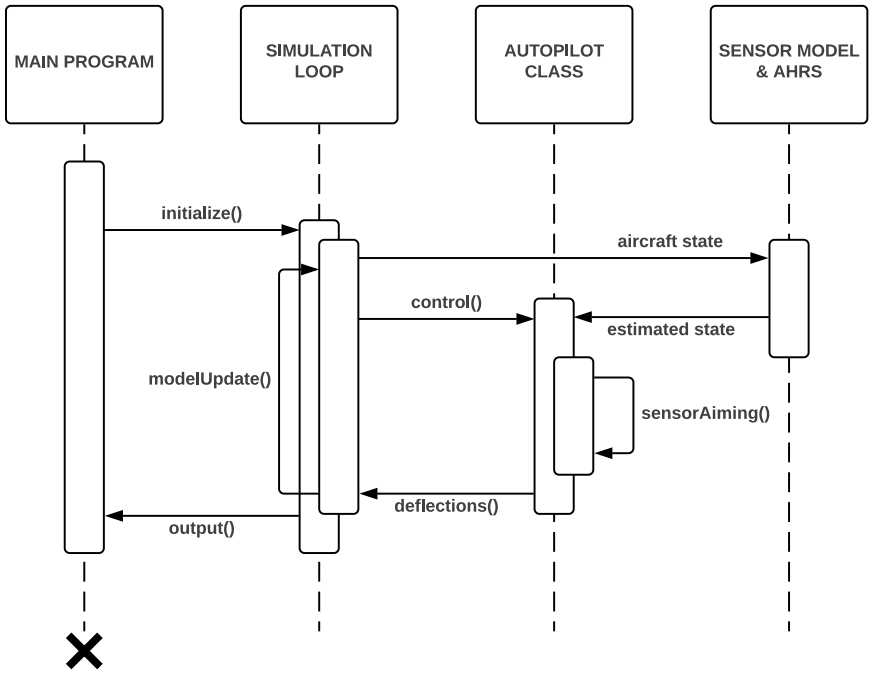

Figure 4.8: Software sequence diagram of the simulation.

4.4 Inertial Sensor Model

The inertial sensor model is defined by:

$$\begin{aligned} \tilde{s}(t) &= s(t) + \mu(t) + w_1(t) \\ \dot{\mu}(t) &= w_2(t) \end{aligned} \tag{4.4}$$

where $\tilde{s}(t)$ is the sensor output, $s(t)$ is the true quantity being measured, $\mu(t)$ is a sensor bias modeled as a random walk, and $w_1(t), w_2(t)$ are white Gaussian noise

processes. This model is used both for generating the simulated sensor signals and in the AHRS for estimating the bias.

4.5 AHRS

The MEKF-based AHRS presented in Chapter 3 is used in the simulation for attitude feedback in control. The introduced sensor model is used to generate random noise and bias upon the angular rate measurements. Two vector measurements, gravity and magnetic north, are used by the AHRS to determine absolute attitude. These measurements are also subject to random noise in the simulation.

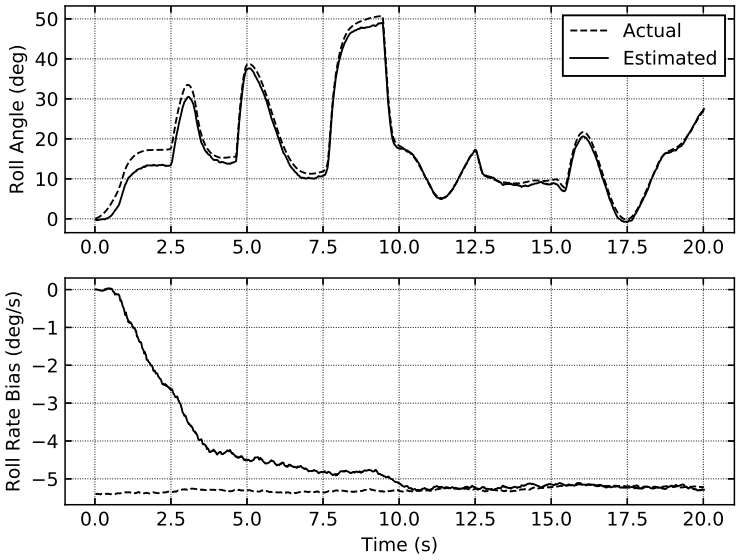

Figure 4.9: Roll angle and roll rate bias estimation segment.

Successful AHRS operation can be determined by observing the convergence on

the angular rate bias. While this is more difficult in reality, as the bias is not always known, in simulation the bias is generated and can be recorded for comparison. Figure 4.9 shows a segment of roll angle estimation with roll rate bias being estimated by the filter. The bias estimate converges on the actual bias at a rate dependent on the filter covariances. The initial filter values are set to zero for these tests, but in practice these would be bias estimates from previous testing, and as such the filter bias covariance would be selected based on the bias stability characteristics of the sensor.

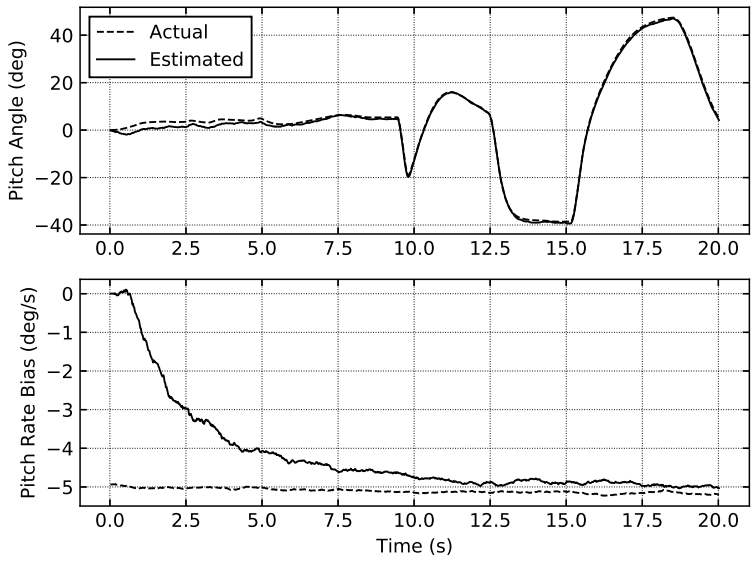

Figure 4.10: Pitch angle and pitch rate bias estimation segment.

Figures 4.10 and 4.11 contain pitch and yaw angle estimation segments, respectively, and their associated bias estimations. Again, the bias estimate converges on the actual bias and as it does so, the attitude estimate improves.

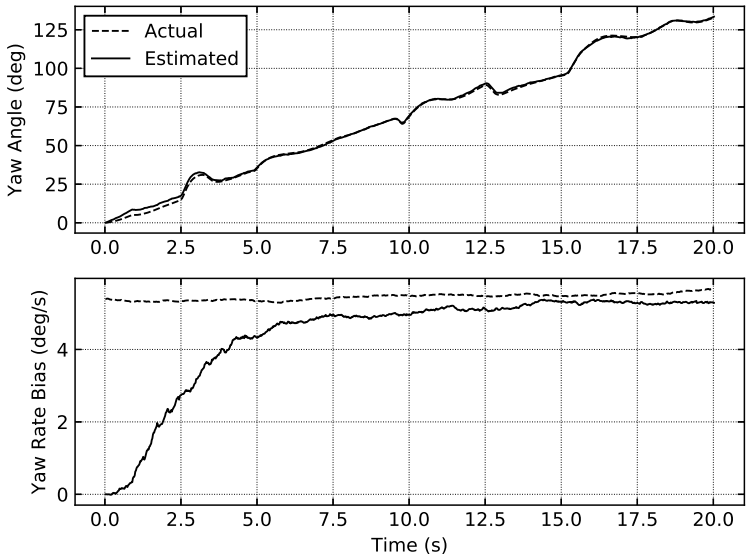

Figure 4.11: Yaw angle and yaw rate bias estimation segment.

4.6 Environmental Disturbance

The environmental disturbances modeled in simulation include gravity and wind, intended to demonstrate the controller capability against constant disturbances. Gravitational acceleration is applied versus lift in the simulation update step during the velocity and position calculation.

Wind is modeled as a constant velocity acting in the horizontal plane. The wind vector $\boldsymbol{w} = [w_x, w_y, 0]^\top$ is summed with the aircraft velocity during the position update.

In the simulation, the wind is randomly generated as:

$$\boldsymbol{w} = N(0, \sigma^2) \frac{\boldsymbol{w}_d}{\|\boldsymbol{w}_d\|} \tag{4.5}$$

where $N(\mu, \sigma^2)$ represents the standard normal distribution with mean μ and variance σ^2 and $\boldsymbol{w}_d = [N(0,1), N(0,1), 0]^\top$ is the randomized wind direction vector. A standard deviation of $\sigma = 5$ m/s is used in all simulation cases.

4.7 Results

Simulation results for various PF and SA scenarios are now presented. Each scenario is run in simulation 40 times[1] under various wind conditions.

4.7.1 Path Following

Three scenarios are considered for evaluative manoeuvres for PF in simulation:

1. A horizontal manoeuvre along a semi-circular path.

2. A vertical manoeuvre along a semi-circular path.

3. A helical manoeuvre consisting of several climbing horizontal circles.

The horizontal manoeuvre is a typical approach for UAV guidance and also typical of intruder or obstacle avoidance. It demonstrates a left and right turn as well as a period of sustained turning. Aircraft trajectory for the horizontal scenario is shown in Figure 4.12. Attitude of the aircraft throughout the trajectory is drawn periodically using symbols showing the banked turns.

[1]Number of simulations was chosen to be 40 based on execution time versus settling of mean error results, which were found to be within 50% in 20 runs.

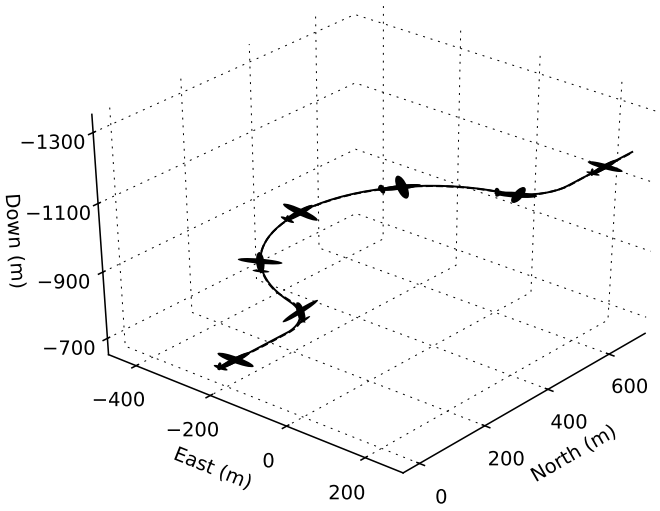

Figure 4.12: Simulated trajectory for horizontal manoeuvre.

Path error for the horizontal manoeuvre is shown in Figure 4.13. The path error is represented in the path frame with respect to the arc length. Therefore, it is the positional offset of the aircraft in the plane normal to the path versus distance along path. Controlling by feedback results in an inherently reactive response and this is clearly shown as a spike in position error during the two sharp turns. An inherent offset is also present during the overall sustained turn and the response is shown to settle on this value.

Vertical avoidance is of particular advantage to UAVs because the risk of ground collision is less urgent versus manned aircraft. Figure 4.14 shows a downward vertical manoeuvre. Due to the strategy of the PF control, the aircraft is pitched to achieve the descent and re-ascent, which demonstrates the applicability to high-performance manoeuvres.

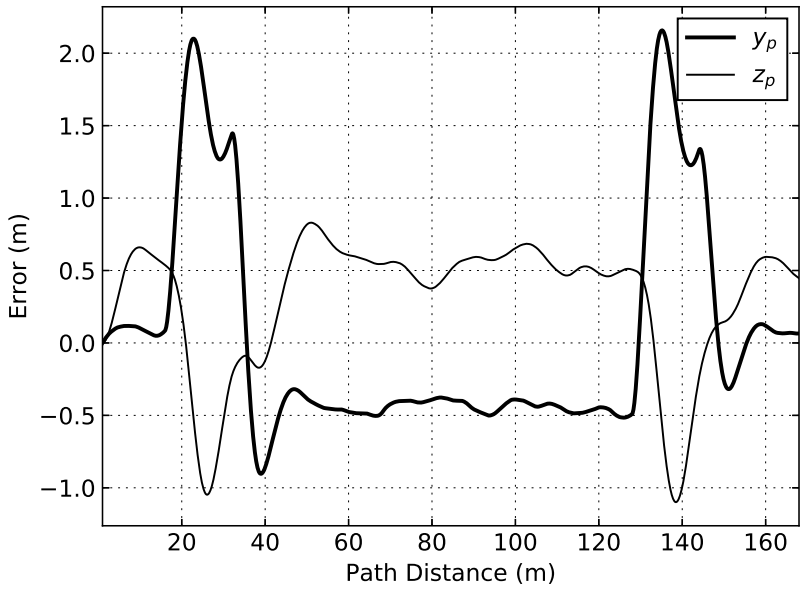

Figure 4.13: Path error for horizontal manoeuvre.

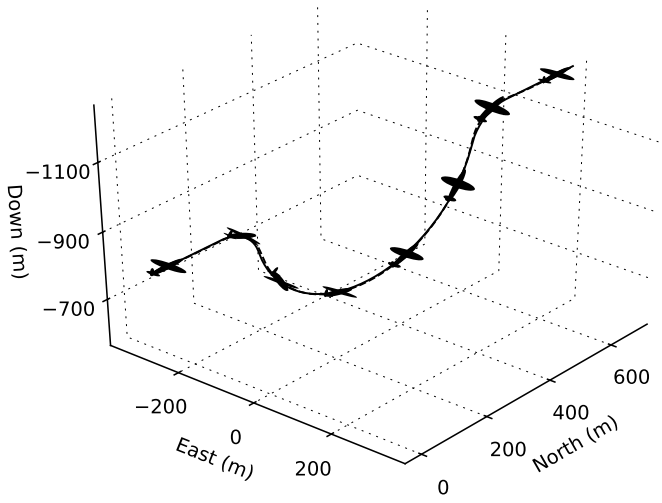

Figure 4.14: Simulated trajectory for vertical manoeuvre.

Path error for the vertical manoeuvre is shown in Figure 4.15. Similarly to the horizontal case, reactive errors are seen during the two sharp turns. It should be noted that the success of using an aggressive pitch angle to climb depends on the availability of excess thrust, as is typical in small UAVs.

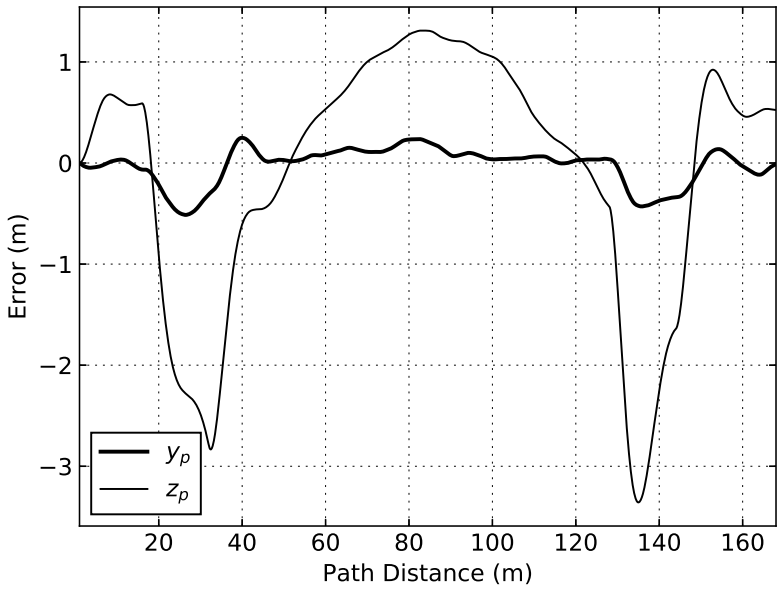

Figure 4.15: Path error for vertical manoeuvre.

Simulation of a helical manoeuvre is used to evaluate the ability of the PF control method to accommodate a changing 3D path. The resulting trajectory is shown in Figure 4.16.

Path error for the helical manoeuvre is shown in Figure 4.17. Due to the constantly changing path and absence of feed-forward compensation, the controller settles on an offset. Similarly to the vertical manoeuvre, sufficient thrust is required.

The deficiency of feedback control for PF is notable in each scenario. Due to the reactive nature of the controller, it is impossible to track a changing path without

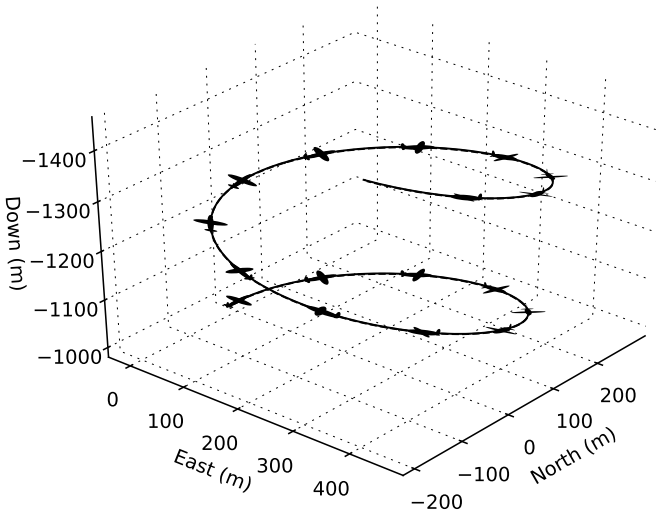

Figure 4.16: Simulated trajectory for helical manoeuvre.

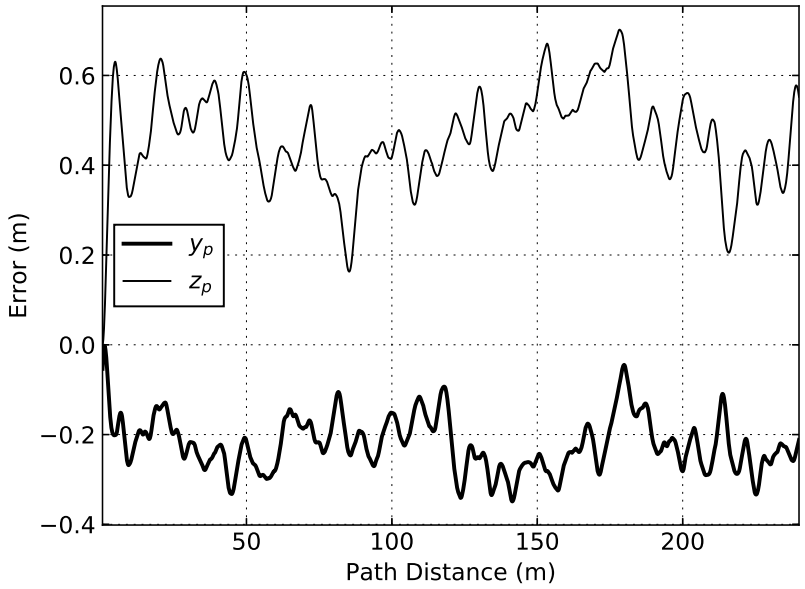

Figure 4.17: Path error for helical manoeuvre.

error. Despite this, the controller maintains a relatively small error throughout the manoeuvre. Path error statistics for each PF scenario are summarized in Table 4.3.

Table 4.3: Path error statistics for PF scenarios over 40 runs.

Manoeuvre	y_p **(RMS) (m)**	z_p **(RMS) (m)**
Horizontal	0.848	0.588
Vertical	0.562	1.567
Helical	0.479	0.489

4.7.2 Sensor Aiming

Two scenarios were considered for SA simulations: radar aiming and ground target aiming. These scenarios demonstrate the practical application of manoeuvre-based SA to provide a control-based solution to an otherwise physically challenging problem.

4.7.2.1 Radar aiming

In the field of non-cooperative detect-and-avoid (DAA) for UAVs, air-to-air radar is a key sensor for detecting intruder aircraft. For this purpose, modified airborne weather radar are a possible solution. These are small and light enough to be flown on a UAV, and could be reduced further if a tilt gimbal was not required.

A typical airborne weather radar has a plate antenna on a horizontal panning mechanism mounted at the front of the aircraft in an electromagnetically transparent nose cone. The radar scans forward of the aircraft across a swept angle as illustrated in Figure 4.18. A manual tilt control allows the radar FOV to be directed upward or downward, which is used for weather detection. Additionally, some radar units can stabilize roll and pitch if connected to the aircraft AHRS output.

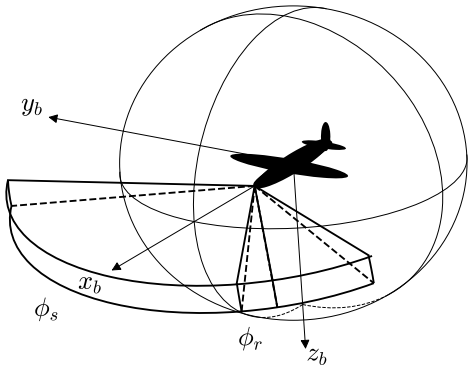

Figure 4.18: Illustration of weather radar beam geometry.

Assume that an airborne weather radar is being used for DAA, with manual tilt set level with the aircraft body axes, and stabilization disabled. The radar beam dimensions are $\phi_r = 7.8°$ both vertical and horizontal. Sweep angle is $\phi_s = 120°$ or $\pm 60°$ with respect to the x_b axis. If an intruder is detected at a range of 10 km, the heading angle to target can be determined relatively precisely from the radar sweep. However, the vertical uncertainty of the target is ± 685 m. By periodically rotating (or tilting) the aircraft about an axis perpendicular to the bearing to target and horizontal to the ground, the vertical uncertainty can be reduced.

Algorithm 3 describes how the SA controller can be used to improve the vertical target resolution of the radar. The set operators $\cap$ and $\setminus$ refer to set intersection and set difference operations, respectively. Target size is a virtual concept representing the apparent region containing the target. That is, the region containing the estimated target and current radar return are compared to reduce the uncertainty in target position. As information about the target is gained, the apparent target size can be

Algorithm 3 Radar-based vertical target resolution.

Require: initial target detection
  $target\_size \leftarrow current\_fov$
  while $target\_alt\_range > vertical\_tol$ **do**
    radarScan()
    if targetInCurrentFOV() **then**
      $target\_size \leftarrow target\_size \cap current\_fov$
    else
      $target\_size \leftarrow target\_size \setminus current\_fov$
    end if
    if midpoint($target\_size$) > midpoint($current\_fov$) **then**
      $tilt\_angle \leftarrow 0.5 * \text{range}(target\_size)$
    else
      $tilt\_angle \leftarrow -0.5 * \text{range}(target\_size)$
    end if
    tiltAircraft($tilt\_angle$)
    $target\_alt\_range \leftarrow$ angle2Altitude($target\_size$)
  end while

refined and compared to the maximum allowable uncertainty.

Considering the radar detection scenario, a test case can be defined for the SA algorithm. Let the radar horizontal scan rate be $48°s^{-1}$, which gives a mean time to complete a scan of 2.5 s. This interval defines the periodicity of the SA events, with a tilt angle of $\pm 7.8/2°$ at an arbitrary heading to target within the radar horizontal sweep. These tilt angles represent the worst-case attitude change required by Algorithm 3. The aircraft is assumed to be flying along a straight line path.

Figure 4.19 shows the aircraft attitude and SA weighting during a simulation of the scenario. As each SA event approaches, the weighting variable (Section 3.4.1) increases towards 1 which represents full SA attitude setpoint. As such, the aircraft attitude is controlled towards the setpoint composed of the required tilt angle. Therefore, the attitudes at times 7.5, 10.0, and 12.5 s represent the SA event objectives. According to Algorithm 3, the tilt angle will alternate as observed in the figure.

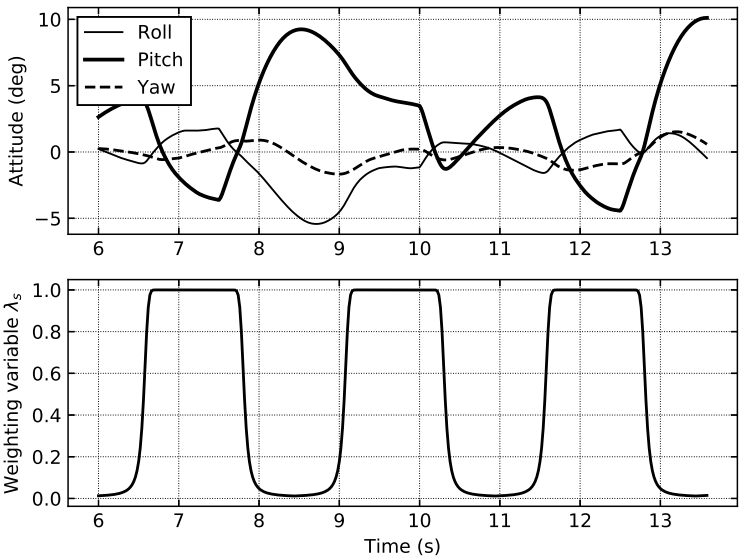

Figure 4.19: Attitude and SA weighting during a segment of the radar aiming manoeuvre.

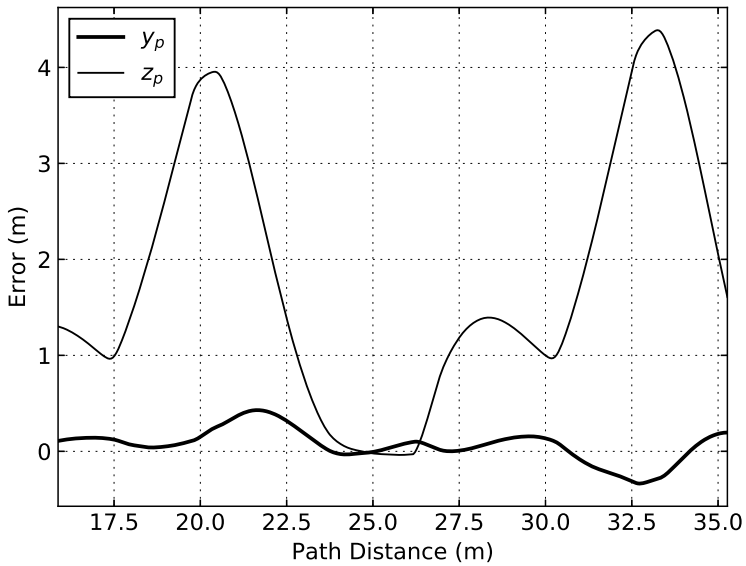

Figure 4.20: Path error segment for radar aiming manoeuvre.

Figure 4.20 contains the path error during this segment. Once each SA event has passed, the weighting variable tends toward 0, representing full PF attitude, and returns to the path. It is observed that pitch changes result in a much larger path error versus roll, which is expected. Therefore, it is more costly from a PF perspective to require SA pitch changes. Figure 4.21 contains the 3D trajectory for the manoeuvre segment.

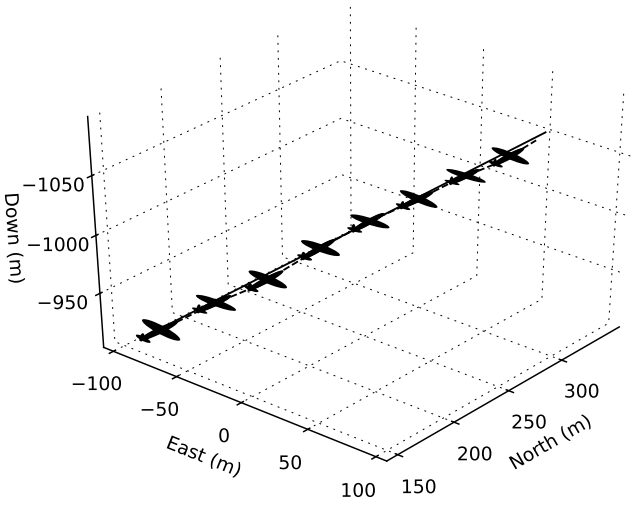

Figure 4.21: Simulated trajectory segment for radar aiming manoeuvre.

The radar aiming scenario was run 40 times in simulation with random target headings and winds. The SA and PF error statistics over these simulations are shown in Table 4.4. The SA attitude error refers to the absolute angular difference between the actual attitude and the SA event setpoint, at the time required. 47.4% of time was dedicated to SA in this scenario during the execution of the aiming manoeuvre.

The overall time spent on SA for the entire mission is less, as it is only used during DAA events.

Table 4.4: Error statistics for radar aiming scenario over 40 runs.

Metric	**Mean**	$\boldsymbol{\sigma}$
SA attitude error (deg)	0.792	0.369
y_p (RMS) (m)	1.497	-
z_p (RMS) (m)	1.518	-

4.7.2.2 Ground target aiming

The second SA scenario is the application of the ground target aiming discussed in Section 3.4. This scenario is intended to demonstrate how a fixed orientation sensor could be pointed towards a ground target without a gimbal.

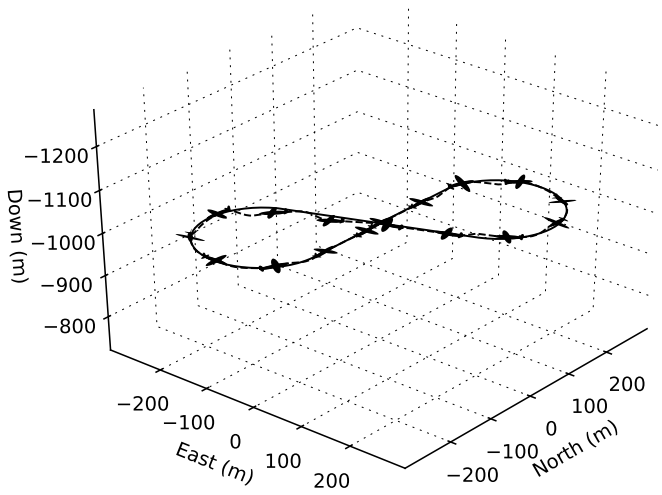

Figure 4.22: Simulated trajectory for loitering figure eight manoeuvre.

A loitering figure eight path at an altitude of 1000 m is used to keep the aircraft above the target at the origin while providing a variety of attitudes to demonstrate the SA controller. The SA events occur periodically with an interval of 5 s. Equation 3.35 gives the roll and pitch angles used to define the SA attitude at each event, assuming the sensor orientation is along the z_b axis of the aircraft.

Figure 4.22 illustrates the simulated trajectory for the figure eight manoeuvre. Small deviations from the flight path are visible during each SA event. Path error is shown in Figure 4.23 for this segment. It can be observed that SA events requiring pitch result in the greatest deviation from the path.

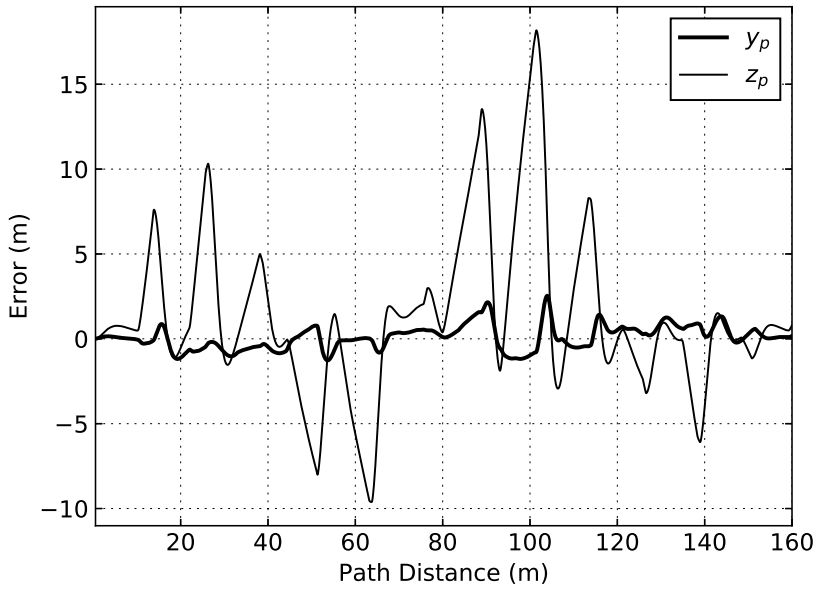

Figure 4.23: Path error for loitering figure eight manoeuvre.

To visualize the efficacy of the SA controller to achieve ground target aiming, the intersection of the z_b axis with the $z = 0$ plane at each SA event was recorded. This was done for each SA event over 40 simulation runs with random wind. The

intersection points are plotted in Figure 4.24. The performance of the SA controller is consistent with the attitude tolerance, sensor noise, and wind disturbance. At an altitude of 1000 m, 1° of attitude error results in 17 m of target error in the $z = 0$ plane.

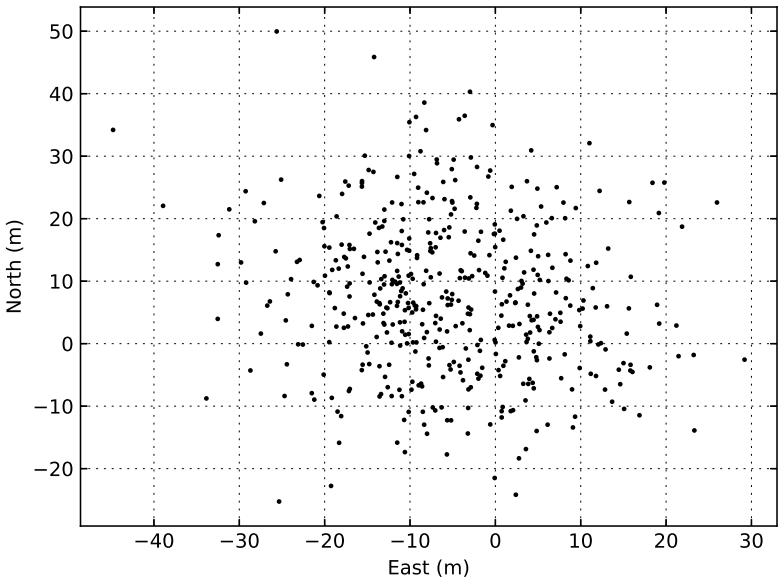

Figure 4.24: SA targets in $z = 0$ plane for loitering figure eight manoeuvre.

Statistics from this scenario over 40 runs are shown in Table 4.5. The $z = 0$ plane means show a bias which is due to the figure eight pattern and asymmetry of SA events that is consistent between runs. The RMS path error is also shown, which is greater than the other PF cases due to the combination of a complex path and SA demands. The time spent on SA for this dedicated mission is 48.2% of the total time, which is approaching the maximum time that can be spent performing SA while adhering to the PF requirement. This maximum time depends on a variety of factors, including path and SA attitude requirements, aircraft performance, and frequency of

SA events. Overall, it is shown that the UAV is able to achieve both PF and SA requirements using the proposed controller.

Table 4.5: Error statistics for ground target aiming scenario over 40 runs.

Metric	**Mean**	$\boldsymbol{\sigma}$
$z = 0$ plane x (m)	-5.34	11.46
$z = 0$ plane y (m)	7.51	12.27
$z = 0$ plane $\sqrt{x^2 + y^2}$ (m)	16.95	8.91
SA attitude error (deg)	2.42	1.29
y_p (RMS) (m)	3.15	-
z_p (RMS) (m)	5.38	-

Chapter 5

Flight Testing

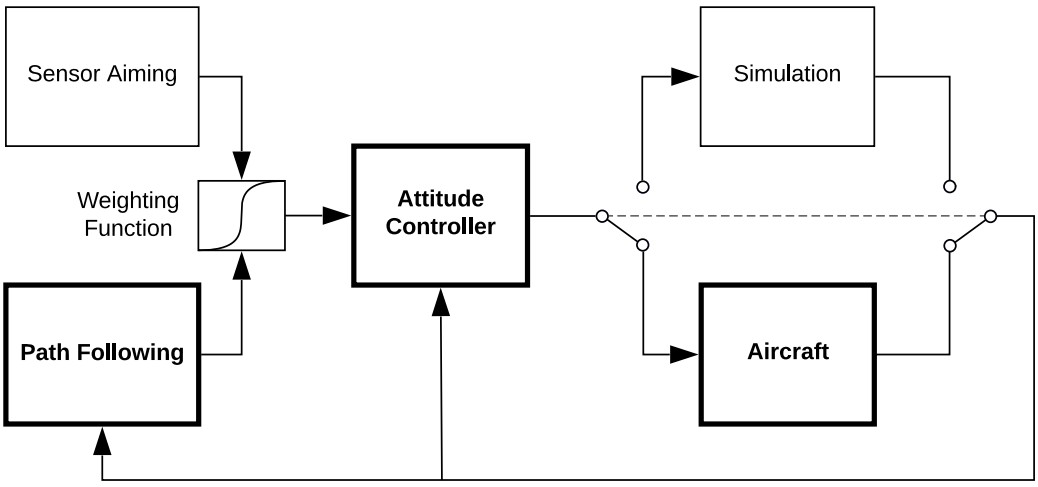

Figure 5.1: Aircraft component of flight control system block diagram.

This chapter describes the development of the UAV test platform, autopilot hardware, and flight test results. Figure 5.1 contains the flight control system block diagram with the components relevant to flight testing highlighted. Aircraft construction, outfitting, autopilot modification, and piloting was performed by the author.

5.1 Aircraft

The UAV platform is based on a Giant Big Stik 1:60 scale model aircraft with electric propulsion. The advantages of this airframe include:

- highly manoeuvrable.
- able to accommodate a payload of 4.5 kg.
- easily repairable for minor damage due to wood construction.
- large enough to possess inertia to resist wind disturbances.
- relatively low cost.

Figure 5.2: Giant Big Stik model aircraft configured as UAV.

The disadvantages of the airframe include: a pulling thrust configuration which restricts the use of forward facing sensors, and a fuselage compartment that requires

wing removal for access. This airframe has been used extensively by Memorial University's UAV research team.

Figure 5.2 shows a photograph of the UAV on manual landing approach. A conventional control surface arrangement and excess thrust availability allow the aircraft to be operated with a relatively short runway and under cross-wind conditions. Maximum take-off weight is approximately 14 kg including 4.5 kg of additional payload capacity beyond frame and batteries. Dimensions of the aircraft are 2 m in wingspan by 1.4 m in length.

The electric motor can supply up to 2,700 W of power, which is in excess of the airframe requirement. Propulsion batteries are four 5-cell lithium polymer with 5,000 mAh capacity. They are connected series-parallel to provide 10-cell 10,000 mAh capacity. At half throttle, the aircraft has an approximate airspeed of 25 ms^{-1} and draws approximately 75 A of current. This gives a practical maximum range of 24 km with a flying time of 16 minutes.

Figure 5.3 shows a photograph of the fuselage compartment with avionics components labelled. A Pixhawk autopilot module [36] forms the basis for the flight control hardware. The autopilot uses an external global positioning system (GPS) and magnetic compass module with an integrated antenna. A 915 MHz two-way radio link is used for ground station telemetry and a 2.4 GHz remote control (RC) radio receiver is used for manual control input. The UAV is also equipped with an Analog Devices ADIS16488 IMU with dedicated power supply and data logging. Autopilot power is from a 3-cell lithium polymer with 1,300 mAh capacity and a nickel-metal hydride battery pack for servo power. The avionics system is shown as a block diagram in Figure 5.4.

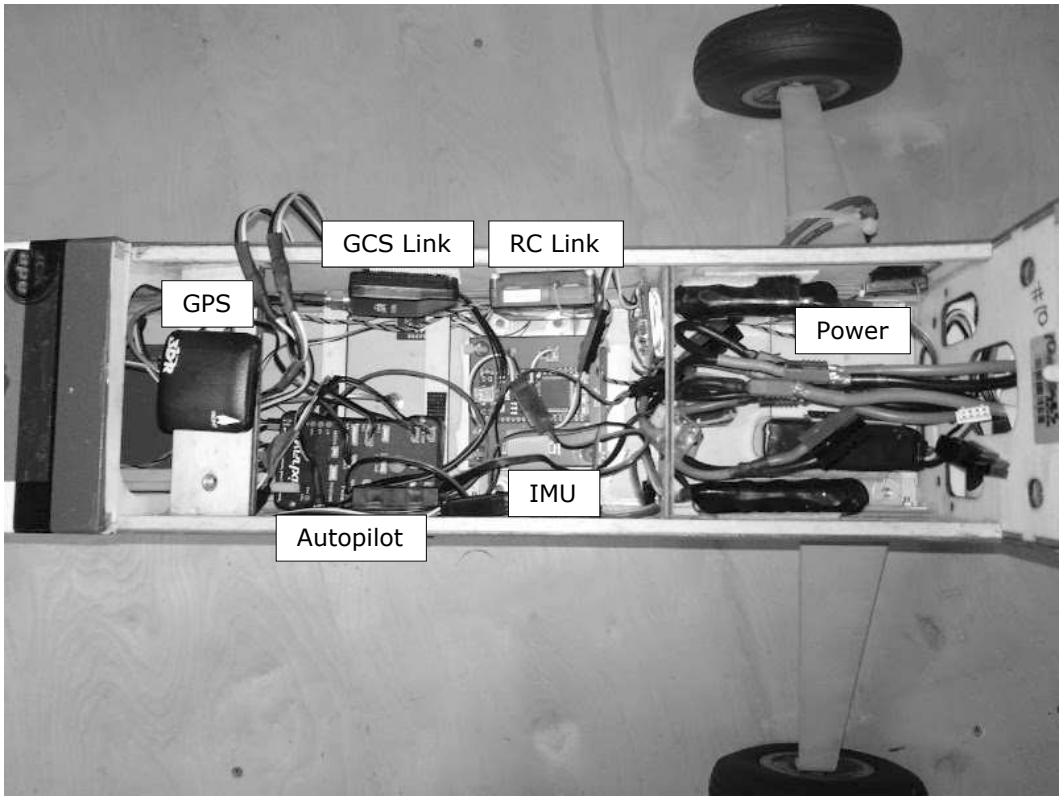

Figure 5.3: Labelled photograph of fuselage compartment showing avionics components.

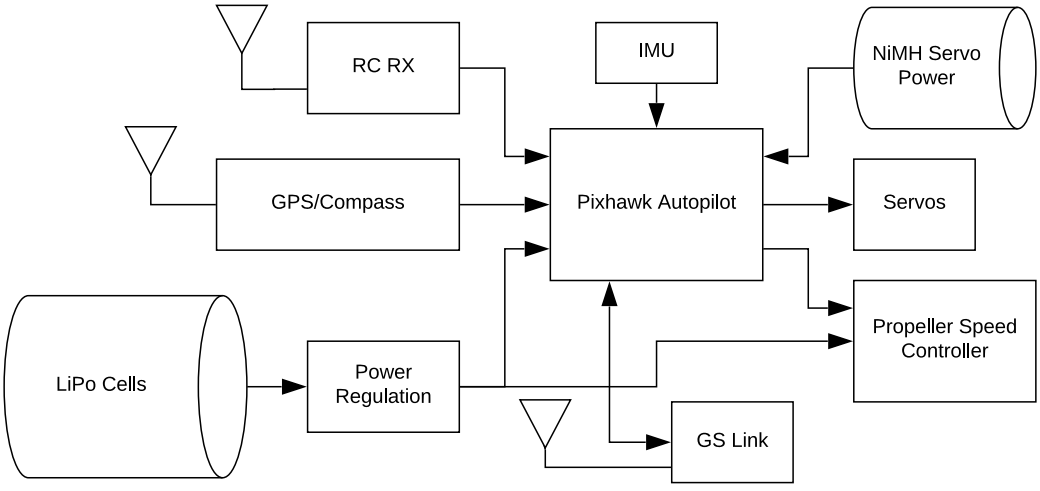

Figure 5.4: Block diagram of avionics system.

5.1.1 Autopilot

The Pixhawk autopilot module is used for this UAV. The open source ArduPilot project software can be installed on this hardware platform. ArduPlane is the fixed-wing version of the generic ArduPilot autopilot software. This software provides a variety of functions, but for controls research and development there are several key features:

- All hardware interfacing is handled by the autopilot software. Variables are accessible via a hardware abstraction layer (HAL), allowing the code to be applicable to multiple hardware configurations.
- Telemetry is available and can be customized to provide signals of interest to the ground control station (GCS).
- Data logging is modular with multiple sample rates for many variables. Data can be converted in post-processing to a number of formats for analysis.
- The RC interface is handled by the software and the ability to pass through RC inputs during automatic flight is implemented. This allows the autopilot to operate at the trim condition and to provide manual disturbance input when required.
- Autopilot mode switching is achieved by assigning an RC channel controlled by a switch to various modes. Individual modes can then be modified in the source code to provide custom functionality.

The ArduPilot is open source and being actively maintained as of this writing. A disadvantage of the project is a lack of detailed technical documentation on the

implementation of the low-level autopilot functions. However, implementation of advanced control algorithms is possible with included mathematics libraries. The availability of matrix and quaternion calculations via the ArduPilot libraries allow rapid development of advanced control algorithms on an embedded platform. For the implementation of the PF algorithm, both the low-level attitude controller and guidance loops were replaced. The code was modified inside the attitude stabilization loop to override the normal control function and implement the PF and attitude controllers.

Control functions for the PF controller had to be "hard coded" in the firmware because there is no interface to configure the controller in this experimental state. Switching to PF mode is achieved by assigning one of the auxiliary RC channels to a mode switch on the transmitter and associating this channel with the autopilot mode in the software. Once the signal is received to begin PF mode, the current altitude and heading are used as a reference to begin the PF segment. During initial testing, software errors resulted in erratic flight and the mode switch was used to quickly return to manual flight to recover the aircraft. Fortunately, no collision occurred during any of the flight tests.

5.2 Flight tests

The flight tests were conducted at a recreational flying field located at the Witless Bay Line on the Avalon Peninsula, Newfoundland, Canada (47° 20′ 26.16″ N, 53° 0′ 48.96″ W).

Table 5.1 provides a summary of the main flight tests conducted. Each test is

Table 5.1: Flight test summary.

Date	Airframe	Purpose
Nov 26, 2016	EPP-FPV	AHRS Validation
Sep 25, 2017	Giant Big Stik UAV	System identification data collection
Nov 26, 2017	Giant Big Stik UAV	Attitude controller tuning
Dec 6, 2017	Giant Big Stik UAV	PF controller testing

described in more detail in the following sections.

5.2.1 AHRS Validation

The first flight test was conducted with a low-cost first person view aircraft to validate the ArduPlane AHRS with an absolute measurement of attitude to determine whether the on-board AHRS was suitable to use for PF controller implementation and data collection. The flight track is shown in Figure 5.5.

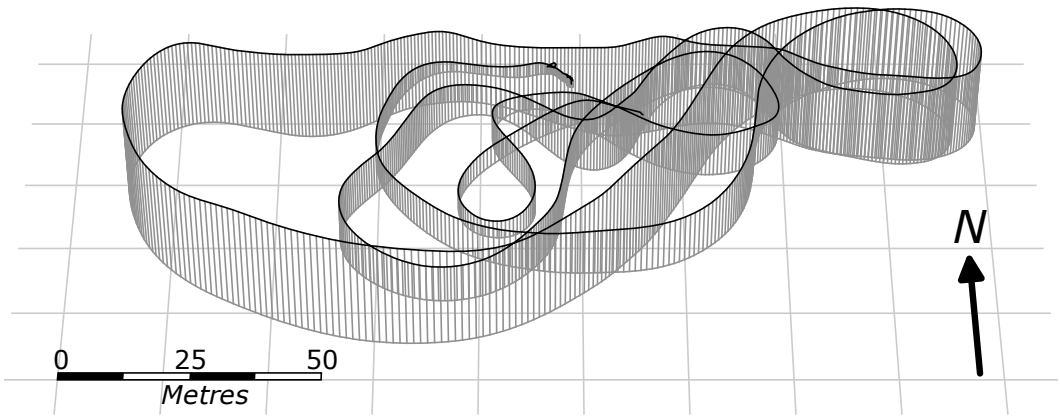

Figure 5.5: Flight track on Nov 26, 2016 with EPP-FPV.

Horizon detection was used to estimate the absolute roll angle of the aircraft. Therefore, a front facing camera was installed in the nose of the aircraft to record video during flight. Figure 5.6 shows a frame from the video collected during the

flight.

Figure 5.6: Video frame showing horizon detection.

A machine vision-based approach to horizon detection using a Hough transform in OpenCV [6], [13] was used with the appropriate parameters. Detection of straight line segments with a minimum length more than half the width of the image gave consistent results for the horizon. An overlay of the automatically detected line is visible in the video frame. Review of the video with overlay shows accurate results except when the horizon was not visible due to excessive pitching of the aircraft.

Figure 5.7 shows the roll angle comparison during a segment of flight data. The horizon-based roll signal has been shifted to align time and orientation offsets. Gaps in the horizon based roll estimate represent failed horizon detection during image processing. AHRS roll estimate very closely matches the absolute roll angle measurement. It is assumed that accuracy is similar in pitch[1] and therefore the AHRS results

[1]Previous experience with AHRS development indicates that roll angle is usually more problematic due to the centripetal acceleration term during banked turns. Aside from this, AHRS pitch and roll calculations are structurally similar.

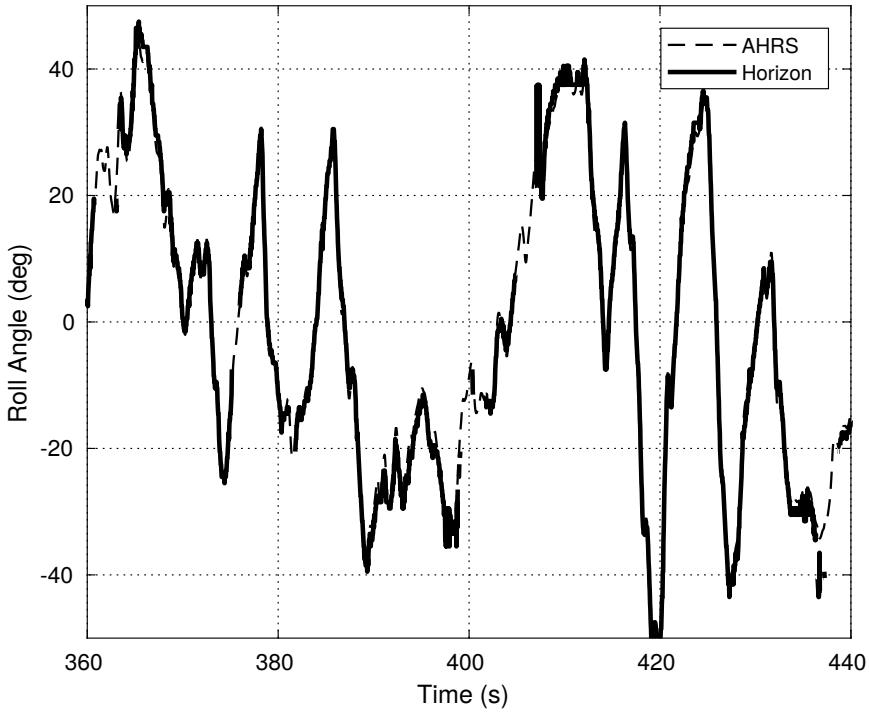

Figure 5.7: Comparison of roll angles from AHRS and horizon detection.

are deemed to be reliable.

5.2.2 System Identification Flight

A manual flight test was conducted to collect open-loop flight data for system identification on September 25, 2017. The flight track is shown in Figure 5.8.

The test manoeuvres included several 400 m length circuits at approximately 100 m altitude. A constant throttle was applied to maintain an average airspeed of 25 ms^{-1}. Oscillatory inputs were applied with each control surface (aileron, elevator, and rudder) to evoke a system response suitable for system identification. Sufficient data was collected to allow both identification and validation segments of data which was detailed in Section 4.2.1.

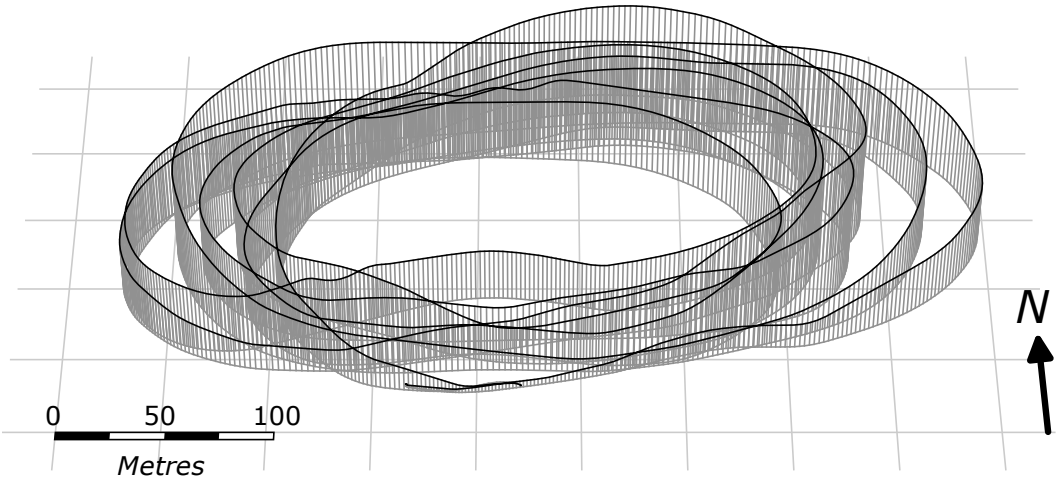

Figure 5.8: Flight track from Sep 25, 2017 with GBS UAV.

5.2.3 Attitude Controller Tuning

Several validation flight tests were conducted at the Witless Bay Line field to ensure the function of the airframe, radio control link, safety override, and inner stabilization control loop. Figure 5.9 shows the flight track during one of the stabilization tuning tests with the segment under automatic control in bold.

Attitude controller tuning giving the desired response was within an order of magnitude from the gains chosen in simulation. From practical experience, a model that can be used to estimate tunings within this range is desirable for control system development. Therefore, the model is deemed accurate for control simulation.

5.2.4 Path Following

The PF controller was implemented in the ArduPilot software and downloaded to the Pixhawk autopilot. A straight line path was used to demonstrate the controller due to the difficulty involved in programming a more complex path. Manual control is

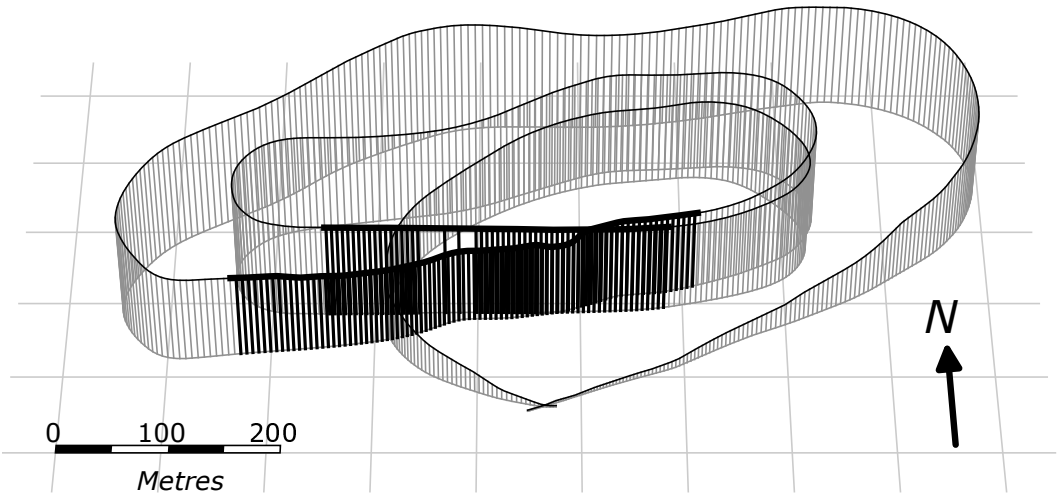

Figure 5.9: Flight track from Nov 26, 2017 with GBS UAV.

used for takeoff and landing, with a controller switch programmed to toggle between manual and autopilot modes. When the switch is activated, PF along a straight line begins at the current altitude and heading.

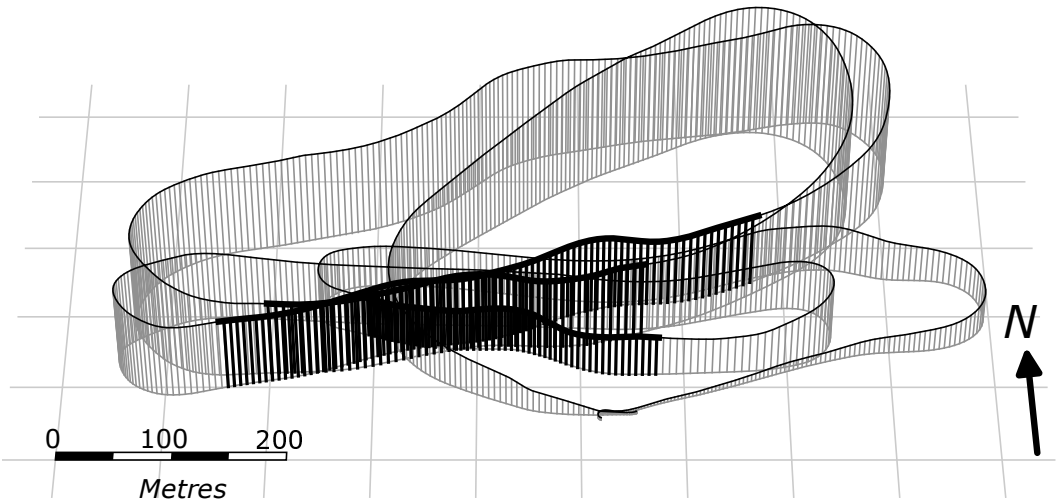

Figure 5.10: Flight track from Dec 6, 2017 with GBS UAV.

Figure 5.10 shows the PF algorithm being demonstrated several times. The bold

segments represent automatic flight. Figure 5.11 shows the attitude and trajectory of the aircraft during one of the PF segments.

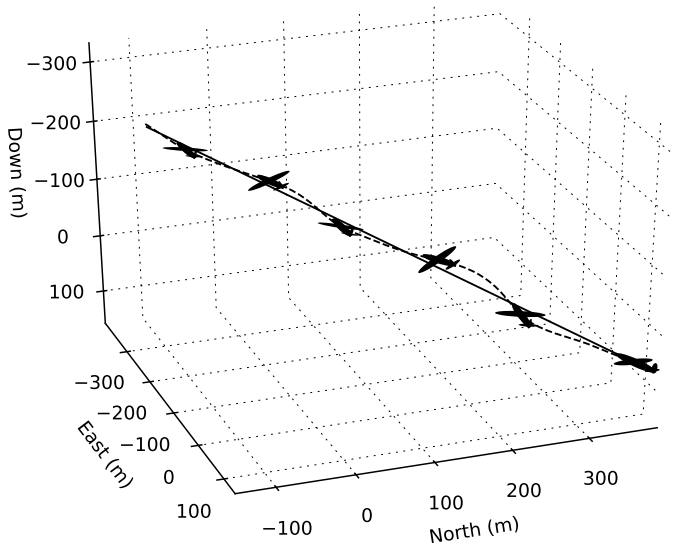

Figure 5.11: Trajectory for straight path flight test.

Path error for the straight line segment is shown in Figure 5.12. It is observed that the controller is under-damped which results in oscillation about the position set-point, yet is stable. Aircraft motion during the PF segment shows controller response dominating versus disturbances and therefore, with adequate tuning, the response can be improved.

The straight line PF was tested three times during the flight. The path error statistics are summarized in Table 5.2. Given the level of effort required to conduct a single flight, it was difficult to fully tune the controller and demonstrate more complex PF segments. However, the path errors for the three segments conducted demonstrate

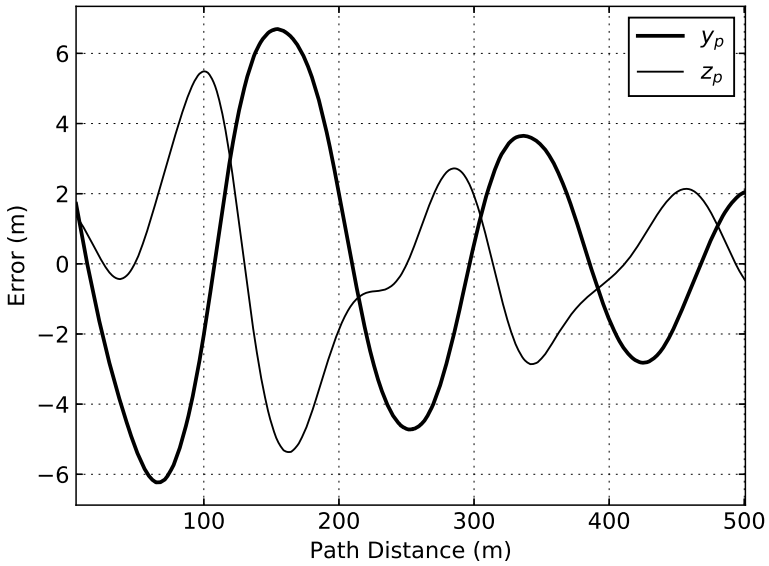

Figure 5.12: Path error for straight path flight test.

a successful controller developed with a fit-for-purpose aircraft model in simulation.

Table 5.2: Path error statistics for PF test segments.

Segment	y_p **(RMS) (m)**	z_p **(RMS) (m)**
Test 1	4.33	2.73
Test 2	3.57	2.45
Test 3	4.44	2.65
Average	4.11	2.65

Chapter 6

Conclusions

This thesis describes a solution to SA for UAVs from theory to experimentally verified simulation. A 3D PF algorithm is developed which uses aircraft attitude as an intermediate control variable. A method for merging SA requirements with PF control is proposed, explicitly using aircraft attitude to perform gimbal-less SA for the first time.

A unique aircraft modelling approach is taken to create a fit-for-purpose model from recorded flight data suitable for use in simulation and control system design. The simulation results demonstrate the success of the PF controller to guide the aircraft along a prescribed path using a generalized approach. SA simulations show the applicability of the controller to achieve sensor aiming without the use of a gimbal.

The modelling and simulating approach is verified with a flight test demonstration of the PF controller. The controller tuning from simulation gave a stable response in actual flight, which justifies the aircraft modelling approach.

The main contribution of this thesis is the SA flight controller. This controller is

shown to be capable of eliminating the need for electronic gimbals for cases where an intermittent and limited duration time-on-target is acceptable. By fixing the sensor to the airframe, a more reliable, efficient, and cost-effective sensing platform can be deployed. Depending on the PF objectives, the path deviations required for SA can be accepted to justify these advantages.

The development of the SA flight controller in this thesis covers the complete design cycle from initial concept, theoretical design, modelling, simulation, and flight testing. Several iterations were performed from flight testing to model identification to simulation, resulting in a refined overall solution.

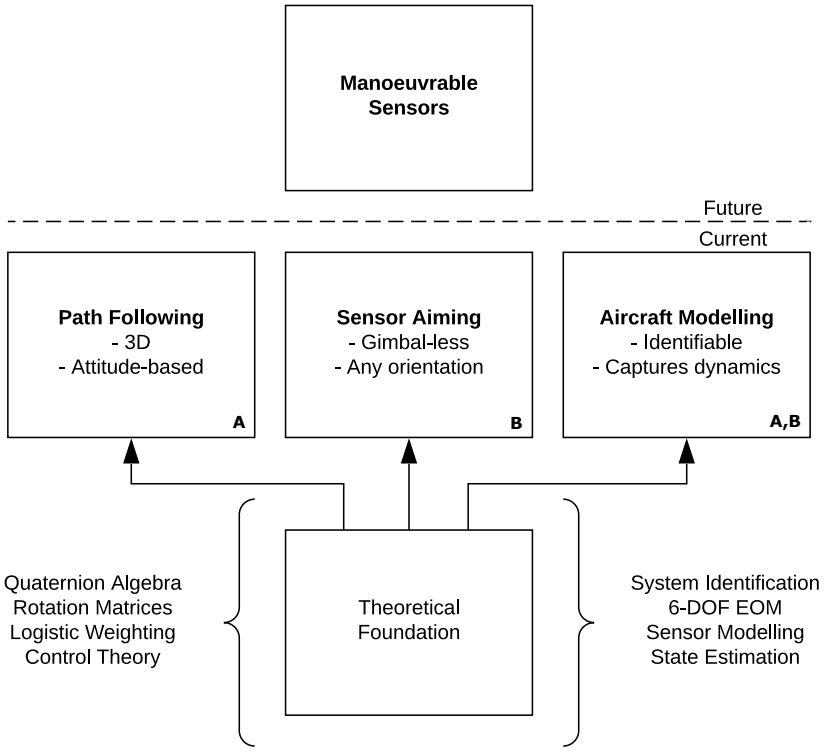

Figure 6.1: Contributions towards manoeuvrable sensor concept.

Figure 6.1 contains a diagram illustrating the major contributions of the current thesis beneath the future possibility of manoeuvrable sensors. The subscripts A and B refer to the publications as discussed in Section 1.3. The theoretical basis serves as the root of the development of applications leading to new concepts regarding unmanned vehicles.

Together, the PF and SA controllers form a type of dynamic path planning method that incorporates both vehicle and sensor tracking requirements. By allocating these requirements to path planning, the design of the vehicle and sensor can be simplified. Path planning has the capacity to incorporate additional requirements such as DAA, terrain tracking, obstacle avoidance, and dynamic seeking.

The concept of a simplified vehicle/sensor combination can be used to make smaller, more robust systems. For example, a small fixed-wing UAV could be designed with a fixed camera embedded inside the airframe with a lens flush to the fuselage. By manoeuvring the aircraft appropriately, the camera could capture images from any conceivable angle or position, despite having no moving parts in the sensor assembly. The SA concept is also applicable to both manned and unmanned aircraft.

Considering the design paradigm of blending vehicle and sensor, the underwater environment is an obvious pursuit. Due to the harsh nature of the environment, eliminating the requirement for a gimbal mechanism or reducing sensor protrusion is beneficial to the design of a vehicle. In addition, isolation in the underwater environment encourages simplification of the vehicle design to reduce the likelihood of malfunction.

All of the concepts presented in this thesis are directly applicable to AUVs with

minimal changes. A similar dynamic modeling approach can be taken with AUVs using system identification data from manoeuvring data. The PF controller can be modified in the control law relating positional offset error to desired attitude, to suit the AUV. Due to the differences in control surface arrangement in AUVs versus aircraft, attitude control must be allocated appropriately from the body torques specified by the attitude controller. Another difference between AUVs and aircraft involves buoyancy, which for AUVs allows depth control without reliance on lift. This makes AUVs more flexible to perform SA attitude changes without significant path departure.

Despite the adequacy of the developed SA flight controller, the ideal method involves an MPC-like approach wherein the future path is predicted and optimized with constraints. Such a predictive method could be used to combine the PF and SA requirements without a weighting function but by imposing state constraints on the future prediction. However, due to the multivariate nature of the 6-DOF path planning problem, MPC implementation is formidable. The non-linearity of the dynamics to be predicted and optimized results in a complex mathematical formulation difficult to solve.

Future work involves incorporating such a predictive method, perhaps in a reduced dimensionality, to improve the intelligence of PF and SA combination. For example, the desired sensor attitude may coincide with a PF attitude that tracks a desirable path, or vice versa. Currently, the proposed controller is not intelligent in this way, and such an outcome would be due to chance. Another aspect of future work involves using the manoeuvrable sensor concept to design a vehicle specifically intended for SA applications, such as a small fixed-wing UAV with a fixed camera.

Bibliography

[1] S. A. Al-Hiddabi and N. H. McClamroch. Tracking and maneuver regulation control for nonlinear nonminimum phase systems: Application to flight control. *IEEE Transactions on Control Systems Technology*, 10(6):780–792, 2002.

[2] R. W. Beard, J. Ferrin, and J. Humpherys. Fixed wing uav path following in wind with input constraints. *IEEE Transactions on Control Systems Technology*, 22(6):2103–2117, 2014.

[3] R. W. Beard, D. B. Kingston, M. Quigley, D. Snyder, R. Christiansen, W. Johnson, T. W. McLain, and M. A. Goodrich. Autonomous vehicle technologies for small fixed-wing uavs. *JACIC*, 2(1):92–108, 2005.

[4] P. Bechtle, T. Gehrmann, C. Sieg, and U. Zillmann. AWEsome: An open-source test platform for airborne wind energy systems. *arXiv preprint arXiv:1704.08695*, 2017.

[5] G. Bishop, G. Welch, et al. An introduction to the kalman filter. *Proc of SIGGRAPH, Course*, 8(27599-23175):41, 2001.

[6] G. Bradski and A. Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc., 2008.

[7] M. Breivik and T. I. Fossen. Principles of guidance-based path following in 2d and 3d. In *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*, pages 627–634. IEEE, 2005.

[8] P. Cheng, J. Keller, and V. Kumar. Time-optimal uav trajectory planning for 3d urban structure coverage. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 2750–2757. IEEE, 2008.

[9] J. C. Chou. Quaternion kinematic and dynamic differential equations. *IEEE Transactions on Robotics and Automation*, 8(1):53–64, 1992.

[10] J. Cortes, S. Martinez, T. Karatas, and F. Bullo. Coverage control for mobile sensing networks. *IEEE Transactions on Robotics and Automation*, 20(2):243–255, 2004.

[11] J. Craighead, R. Murphy, J. Burke, and B. Goldiez. A survey of commercial & open source unmanned vehicle simulators. In *Robotics and Automation, 2007 IEEE International Conference on*, pages 852–857. IEEE, 2007.

[12] J. L. Crassidis, F. L. Markley, and Y. Cheng. Survey of nonlinear attitude estimation methods. *Journal of Guidance, Control, and Dynamics*, 30(1):12–28, 2007.

[13] E. R. Davies. *Machine vision: theory, algorithms, practicalities*. Elsevier, 2004.

[14] A. Dorobantu, A. A. Ozdemir, K. Turkoglu, P. Freeman, A. Murch, B. Mettler, and G. Balas. Frequency domain system identification for a small, low-cost,

fixed-wing uav. In *AIAA Guidance, Navigation, and Control Conference*, page 6719, 2011.

[15] J. C. Doyle, K. Glover, P. P. Khargonekar, and B. A. Francis. State-space solutions to standard h-2 and h-infinity control problems. *IEEE Transactions on Automatic Control*, 34(8):831–847, 1989.

[16] M. Euston, P. Coote, R. Mahony, J. Kim, and T. Hamel. A complementary filter for attitude estimation of a fixed-wing uav. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 340–345. IEEE, 2008.

[17] J. Evans, G. Inalhan, J. S. Jang, R. Teo, and C. J. Tomlin. Dragonfly: A versatile uav platform for the advancement of aircraft navigation and control. In *Digital Avionics Systems, 2001. DASC. 20th Conference*, volume 1, pages 1C3–1. IEEE, 2001.

[18] T. I. Fossen. *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.

[19] E. Frew, T. McGee, Z. Kim, X. Xiao, S. Jackson, M. Morimoto, S. Rathinam, J. Padial, and R. Sengupta. Vision-based road-following using a small autonomous aircraft. In *Aerospace Conference, 2004. Proceedings. 2004 IEEE*, volume 5, pages 3006–3015. IEEE, 2004.

[20] F. Gavilan, R. Vazquez, and E. F. Camacho. An iterative model predictive control algorithm for uav guidance. *IEEE Transactions on Aerospace and Electronic Systems*, 51(3):2406–2419, 2015.

[21] W. R. Hamilton. On a new species of imaginary quantities connected with a theory of quaternions. In *Proceedings of the Royal Irish Academy*, volume 2, pages 4–1, 1844.

[22] D. Henderson. Euler angles, quaternions, and transformation matrices for space shuttle analysis. 1977.

[23] N. V. Hoffer, C. Coopmans, A. M. Jensen, and Y. Chen. A survey and categorization of small low-cost unmanned aerial vehicle system identification. *Journal of Intelligent & Robotic Systems*, 74(1-2):129, 2014.

[24] I. Horowitz. Invited paper survey of quantitative feedback theory (QFT). *International Journal of Control*, 53(2):255–291, 1991.

[25] N. Hovakimyan and C. Cao. *L1 adaptive control theory: guaranteed robustness with fast adaptation*, volume 21. SIAM-Society for Industrial and Applied Mathematics, 2010.

[26] E. N. Johnson and D. P. Schrage. The georgia tech unmanned aerial research vehicle: Gtmax. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, pages 11–14, 2003.

[27] T. L. Jordan, J. V. Foster, R. M. Bailey, and C. M. Belcastro. Airstar: A uav platform for flight dynamics and control system testing. In *25th AIAA Aerodynamic Measurement Technology and Ground Testing Conference*, pages 2006–3307, 2006.

[28] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[29] I. Kaminer, A. Pascoal, E. Xargay, N. Hovakimyan, C. Cao, and V. Dobrokhodov. Path following for unmanned aerial vehicles using l1 adaptive augmentation of commercial autopilots. *Journal of Guidance, Control, and Dynamics*, 33(2):550, 2010.

[30] Y. Kang and J. K. Hedrick. Linear tracking for a fixed-wing uav using nonlinear model predictive control. *IEEE Transactions on Control Systems Technology*, 17(5):1202–1210, Sept 2009.

[31] Y. A. Kapitanyuk, A. V. Proskurnikov, and M. Cao. A guiding vector-field algorithm for path-following control of nonholonomic mobile robots. *IEEE Transactions on Control Systems Technology*, 2017.

[32] J. B. Kuipers et al. *Quaternions and rotation sequences*, volume 66. Princeton university press, 1999.

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

[34] U. Ly, A. E. Bryson, and R. H. Cannon. Design of low-order compensators using parameter optimization. *Automatica*, 21(3):315–318, 1985.

[35] F. L. Markley. Attitude error representations for kalman filtering. *Journal of Guidance, Control, and Dynamics*, 26(2):311–317, 2003.

[36] L. Meier, P. Tanskanen, F. Fraundorfer, and M. Pollefeys. Pixhawk: A system for autonomous flight using onboard computer vision. In *Robotics and Automation (ICRA), 2011 IEEE international conference on*, pages 2992–2997. IEEE, 2011.

[37] A. Micaelli and C. Samson. *Trajectory tracking for unicycle-type and two-steering-wheels mobile robots*. PhD thesis, INRIA, 1993.

[38] K. Murrant. *Attitude and heading reference system for small unmanned aircraft collision avoidance maneuvers*. 2012.

[39] R. Naldi, M. Furci, R. G. Sanfelice, and L. Marconi. Robust global trajectory tracking for underactuated vtol aerial vehicles using inner-outer loop control paradigms. *IEEE Transactions on Automatic Control*, 62(1):97–112, 2017.

[40] J. Nelder. The fitting of a generalization of the logistic curve. *Biometrics*, 17(1):89–110, 1961.

[41] A. Oosedo, S. Abiko, S. Narasaki, A. Kuno, A. Konno, and M. Uchiyama. Flight control systems of a quad tilt rotor unmanned aerial vehicle for a large attitude change. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 2326–2331. IEEE, 2015.

[42] W. F. Phillips. *Mechanics of flight*. John Wiley & Sons, 2004.

[43] M. Ryll, H. H. Bulthoff, and P. R. Giordano. First flight tests for a quadrotor uav with tilting propellers. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 295–302. IEEE, 2013.

[44] C. Samson. Control of chained systems application to path following and time-varying point-stabilization of mobile robots. *IEEE Transactions on Automatic Control*, 40(1):64–77, 1995.

[45] W. Siwakosit, S. Snell, and R. A. Hess. Robust flight control design with handling qualities constraints using scheduled linear dynamic inversion and loop-shaping. *IEEE Transactions on Control Systems Technology*, 8(3):483–494, 2000.

[46] S. A. Snell, D. F. Nns, and W. L. Arrard. Nonlinear inversion flight control for a supermaneuverable aircraft. *Journal of Guidance, Control, and Dynamics*, 15(4):976–984, 1992.

[47] B. L. Stevens, F. L. Lewis, and E. N. Johnson. *Aircraft control and simulation: dynamics, controls design, and autonomous systems*. John Wiley & Sons, 2015.

[48] P. Sujit, S. Saripalli, and J. Borges Sousa. Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicless. *Control Systems, IEEE*, 34(1):42–59, 2014.

[49] A. Tayebi and S. McGilvray. Attitude stabilization of a vtol quadrotor aircraft. *IEEE Transactions on Control Systems Technology*, 14(3):562–571, 2006.

[50] J. Tisdale, A. Ryan, M. Zennaro, X. Xiao, D. Caveney, S. Rathinam, J. K. Hedrick, and R. Sengupta. The software architecture of the berkeley uav platform. In *Computer Aided Control System Design, 2006 IEEE International Conference on Control Applications, 2006 IEEE International Symposium on Intelligent Control, 2006 IEEE*, pages 1420–1425. IEEE, 2006.

[51] D. Titterton and J. L. Weston. *Strapdown inertial navigation technology*, volume 17. IET, 2004.

[52] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. L. Grixa, F. Ruess, M. Suppa, and D. Burschka. Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue. *IEEE Robotics & Automation Magazine*, 19(3):46–56, 2012.

[53] J.-Y. Wen and K. Kreutz-Delgado. The attitude control problem. *IEEE Transactions on Automatic Control*, 36(10):1148–1162, 1991.

[54] T. R. Yechout. *Introduction to aircraft flight mechanics*. Aiaa, 2003.

[55] H. Zhou, H. Kong, L. Wei, D. Creighton, and S. Nahavandi. Efficient road detection and tracking for unmanned aerial vehicle. *IEEE Transactions on Intelligent Transportation Systems (TITS)*, 16(1):297–309, 2015.

[56] J.-H. Zhu. A survey of advanced flight control theory and application. In *Computational Engineering in Systems Applications, IMACS Multiconference on*, volume 1, pages 655–658. IEEE, 2006.