DESIGN OF AN INTELLIGENT FAULT-TOLERANT CONTROL SYSTEM FOR AN UNMANNED AERIAL VEHICLE

Duncan Higgs

A Dissertation submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, in fulfilment of the requirements of the degree of Master of Science in Engineering

Johannesburg 2014

Declaration

I, the undersigned, hereby declare that the work contained in this dissertation is my own original work, unless explicit reference is made. I also confirm that the dissertation in its entirety or in part has not previously been submitted to any university for a degree.

Duncan Higgs:.................................. Date:...

Acknowledgements

The construction of a dissertation is a challenging task that is not easily completed in isolation. Technical, strategic and emotional guidance was generously provided by many individuals to whom I would like to express my deep gratitude. In particular I would like to thank the following people for their contributions and support.

- Prof J.O Pedro for your guidance, consultation and review of the various stages of this project. Your input has been invaluable and much appreciated.
- Mr J.S Monk for the opportunity to complete this MSc dissertation and for the many discussions in the hallway and general support given. Without this support and encouragement this dissertation would not have been possible.
- Gen. (Ret.) D Barker for providing me with the opportunity to undertake this project, as well as your support and leadership through this time.
- Mr P. Skinner for the wind tunnel data provided.
- Mr N. Moore for your friendship, support and sanity checks. It is much appreciated.
- CSIR for providing the opportunity to undertake such a project and for providing the funding for this project.
- CSIR - Aeronautic Systems Competency for the vast supply of wind tunnel data used in this dissertation and for the general supportive attitude displayed by all members of ASC. The wealth of knowledge and experience of ASC was a huge asset that was so generously shared by all. The ASC family made me feel that I was a part of the team and were always available to assist and this is much appreciated.
- My parents, Neil and Debbie Higgs for their continued love and support and to my father for the time spent reviewing and proofing-reading this dissertation.
- My grandfather, Keith Higgs, for countless hours discussing this topic as well as many others and for showing great interest and enthusiasm in everything I did.
- My girlfriend, Jessica van Wouw, for her continued support and patience throughout the duration of this project

Abstract

An intelligent fault-tolerant control system for an unmanned aerial vehicle was developed that was designed to be capable of tolerating a number of different control actuator faults. The development of the control system focused on the simulation of the system using a nonlinear flight dynamic model with the aim to implement this control strategy in an operational UAS in the future. The nonlinear flight dynamic model was a high fidelity, six-degree-of-freedom model that made use of available wind tunnel data. The model considered the general equations of motion of an asymmetric rigid aircraft within the troposphere and also considered motor, and control actuator dynamics. The proposed control strategy consisted of a model reference fuzzy logic adaption algorithm combined with a daisy chain allocation algorithm. An equivalent desired first order behaviour was used to generate an ideal response to a control input and used as a reference for the adaption algorithm to follow. The allocation algorithm made use of secondary and tertiary control effectors that were used only after the primary control surface reached its physical limits of travel. A number of control actuator failures, of varying severity, were modelled that included elevator failures, aileron failures and combined aileron and elevator failures. The results showed the proposed control system was better able to tolerate the simulated failures when compared to the unmodified autopilot. For more severe failures it was found that the control allocation algorithm was a necessity and in some cases the adaption algorithm when used in isolation, induced control instability. Tuning of the adaption rates of the adaption algorithm was found to have a significant effect on the performance of the system. In some cases the incorrect adaption rate caused degraded control performance. It was, however, concluded that the proposed control strategy did provide a degree of fault-tolerance for the failure scenarios considered. It is recommended that research into the effects of adaption rates, auxiliary control functions (such as feedforward loops) and the use of health monitoring be considered for a more practical system. It is also recommended that extensive testing be conducted with hardware in the loop simulators before this system be implemented.

Scope and Contribution

The scope of this dissertation involves the initial research and design of a fault-tolerant control system for a UAS. A detailed high fidelity flight dynamic model is used to simulate and test the proposed control strategy under a number of control actuator failures. The scope is limited to control actuator failures and in particular to aileron and elevator failures. The system is to show fault tolerance in the longitudinal axis and lateral axis in isolation as well as show tolerance in both axes combined. Limitations of the flight dynamics models validity are recognized, these include:

- Subsonic flight
- Flight within ± 30° sideslip angles
- Flight within -10°,+ 20° angle of attack
- Flight within the troposphere with no turbulence

This research has contributions in the development of a high fidelity flight dynamic model for the simulation of the fault tolerance control system. Secondly, there is little research to the effect of using fuzzy logic as a method of control adaption for both the longitudinal and lateral axes. This research also has contributions in combined failures of control actuators and this research also contributes to the understanding of combining two fault-tolerance control techniques namely, control adaption and control allocation.

Published Work

As yet, this work remains unpublished. At the time of writing a journal article, titled *"Design of an intelligent fault-tolerant control system for an UAV"*, had been written and submitted to the Royal Aeronautical Society of South Africa for consideration.

Table of Contents

List of Figures

List of Tables

Nomenclature

Symbol	Description	Units
X	Force in the x-axis (subscript refers to system being used)	N
Y	Force in the y-axis (subscript refers to system being used)	N
Z	Force in the z-axis (subscript refers to system being used)	N
L	Moment about x-axis (subscript refers to system being used)	Nm
M	Moment about y-axis (subscript refers to system being used)	Nm
N	Moment about z-axis (subscript refers to system being used)	Nm
B	Subscript indicating body axis system	N/A
E	Subscript indicating earth axis system	N/A
W	Subscript indicating wind axis system	N/A
ρ_0	Air density at sea level defined by ISA	kg/m^3
ρ	Air density	kg/m^3
σ	Air density ratio	N/A
CL	Lift coefficient	N/A
CD	Drag coefficient	N/A
CM	Pitching moment coefficient	N/A
CY	Side force coefficient	N/A
CR	Rolling moment coefficient	N/A
CN	Yawing moment coefficient	N/A
CT	Thrust coefficient	N/A
CP	Power coefficient	N/A
CQ	Torque coefficient	N/A
J	Advance ratio	N/A
α	Angle of attack	°
β	Angle of sideslip	°
U	Translational velocity of aircraft in the x body axis direction	m/s
V	Translational velocity of aircraft in the y body axis direction	m/s
W	Translational velocity of aircraft in the z body axis direction	m/s
p	Rotational velocity of aircraft about the x body axis	°/s
q	Rotational velocity of aircraft about the y body axis	°/s
r	Rotational velocity of aircraft about the z body axis	°/s
m	Mass	kg
g	Acceleration due to gravity	m/s^2
V	Free stream velocity	m/s
I_x	Second moment of inertia about X body axis	kg.m^2
I_y	Second moment of inertia about Y body axis	kg.m^2
I_z	Second moment of inertia about Z body axis	kg.m^2
I_{xz}	Product of inertia about XZ body axis	kg.m^2
I_{xy}	Product of inertia about XY body axis	kg.m^2
I_{yz}	Product of inertia about YZ body axis	kg.m^2

T_0	ISA temperature at sea level	K
P_0	ISA pressure at sea level	Pa
$DALR$	Dry adiabatic lapse rate as defined by the ISA	K/m
R	Universal gas constant	J/mol
M	Molar mass	kg/mol
h	Altitude above sea level	m
T	Thrust	N
Q	Torque	Nm
κ	Down thrust line angle	°
κ_s	Side thrust line angle	°
h	Height of thrust line above CG	m
l	Distance of thrust line fore of CG	m
w	Distance of thrust line starboard of CG	m
θ	Pitch angle relative to earth axis	°
ϕ	Roll angle relative to earth axis	°
ψ	Yaw angle relative to earth axis	°
t	Time	s
h	Time increment	s
k	Runge Kutta estimator	N/A
Ψ	Latitude	rad
Θ	Longitude	rad

1 Chapter 1 - Introduction

1.1 Research Background

The first rudimentary, but functional, autopilot design was first developed as early as 1914 by Lawrence Sperry, the son of gyrocompass co-inventor, Elmer Sperry [1]. He demonstrated his autopilot to an air show crowd by flying his Curtis C-2 “hands free”. While basic, it provided a demonstration of what was to follow. Autopilots have evolved into complex control systems that have the ability to optimise aircraft performance, reduce pilot work load, reduce airspace congestion and improve flight safety. Autopilots range in complexity from simple wing levelling systems in small general aviation aircraft to complex systems capable of controlling unstable military fighter aircraft, such as the Euro fighter shown in Figure 1.1.

Figure 1.1: Euro fighter performing a high G manoeuvre [2].

Advances in autopilot design and an increasing need for systems capable of laborious or dangerous flight missions led to the development of Unmanned Aerial Systems (UAS). UAS aircraft are particularly useful in the military context as they offer a safer platform for reconnaissance and observation, and more recently, attack. Aircraft such as the Northrop Grumman Global Hawk and General Atomics Predator drone, shown in Figure 1.2 (a) and (b), have been actively used by the U.S. Air Force in recent engagements.

The capabilities of UAS in the civilian context are rapidly becoming apparent. The interest shown by law enforcement, border patrol, research groups and general interest groups has seen the rapid development of small UAS. These systems have been made possible through the increasing miniaturisation of electronic components and advances in energy storage capacity.

The research into control of an UAS is extensive and widely varying and, in particular, the field of fault-tolerant control is equally extensive. Various techniques have been developed to ensure a fault-tolerant system. These include the use of redundancy, robustness, adaptability and artificial intelligence, all of which will be described in more detail in the chapters that follow. All of these techniques have some merit and their suitability largely depends on the application.

Figure 1.2: Typical UAS: (a) Northrop Grumman Global Hawk [3], (b) General Atomics Predator Drone [4]

1.2 Types of Fault-Tolerant Control Systems

Fault-tolerant control techniques fall into a number of classes some of which are:

- Redundancy
- Fault detection, diagnosis and isolation
- Adaptive control (Indirect or Direct)
- Reconfigurable control

These categories are further subdivided into an ever increasing series of techniques that all have their advantage and disadvantages. A summary of the various techniques examined and how they relate to each other in the fault-tolerant control design problem is given in Figure 1.3. While the diagram does not hope to include all possible methodologies and approaches, it does show the basic ideas around fault-tolerant design techniques.

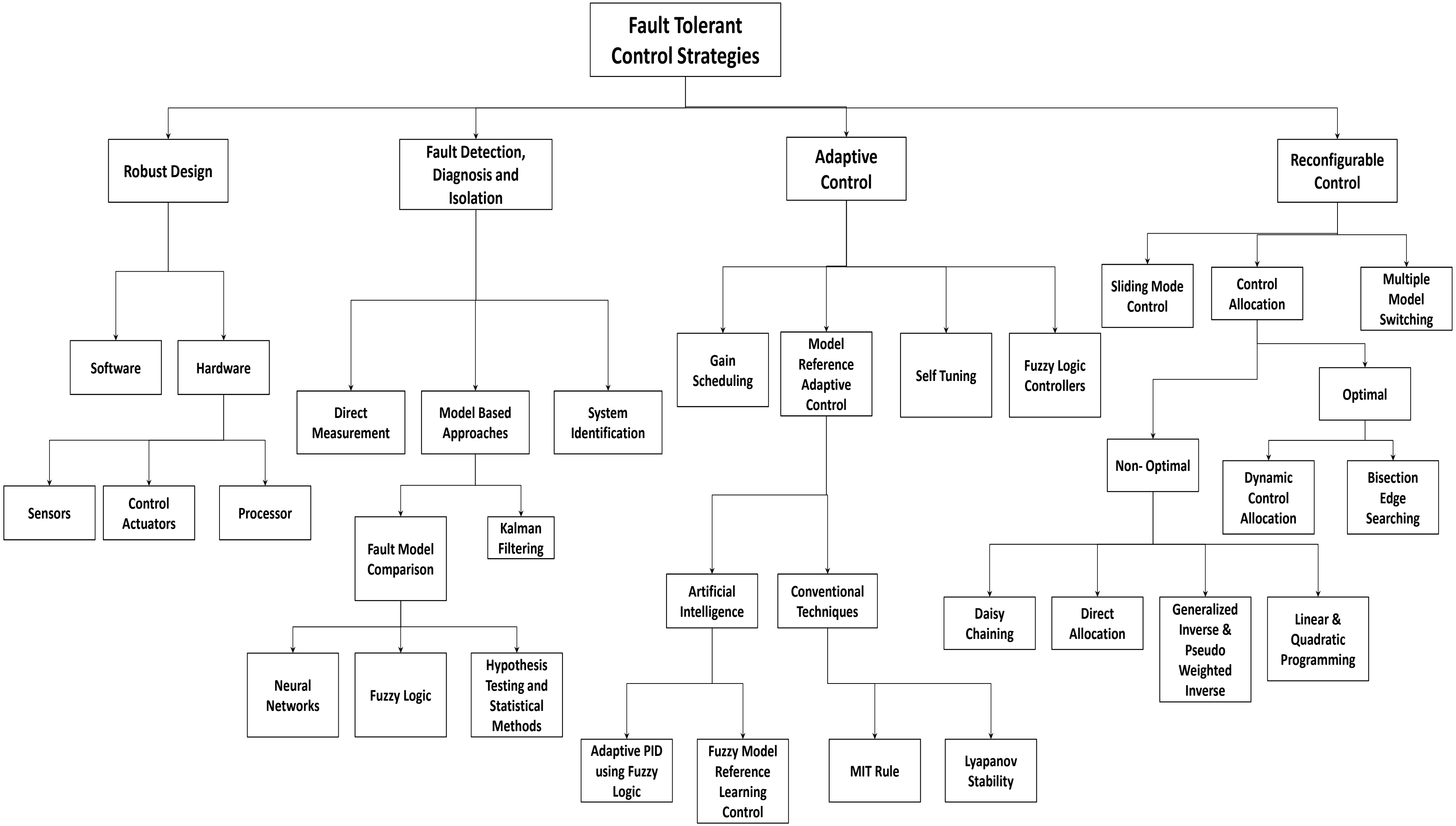


Figure 1.3: Diagram indicating an overview of fault-tolerant controller design

1.3 Challenges Associated with the Design of Fault-Tolerant Control Systems

There are a number of challenges associated with the design of a fault-tolerant control system. These include, but are not limited to[5]:

- Early and successful detection of faults (in the case of a FDD system)
- Uncertainty within the modelled plant and robustness to unknown disturbances
- Ensuring simplicity and integration into existing control system architectures
- Ensuring that the fault-tolerant system is able to asymptotically track the desired set point.
- Fault-tolerant systems may be controlling a nonlinear plant that behaves linearly at the operating point. In the event of a failure there may be a shift in plant dynamics to the extent that the system must control a highly nonlinear system. [6]
- Designing a system that is able to detect faults in real world environments where faults and uncertainties are mixed (defined as the "Robustness problem") [7]

The first challenge for any fault-tolerant system is the identification of a fault. This may be done actively, as is the case with detection systems or this may be done passively, where the system simply reacts to the disturbance as in model based systems. A robust system is also required, as a system that is too sensitive to uncertainties, in both the operating environment and discrepancies in modelled behaviour will result in a system that triggers false alarms or unnecessary changes in control behaviours that may lead to unstable system performance. Of great difficulty is the ability of the fault-tolerant system to maintain stability after a fault has occurred as well as ensure that instabilities are not created during normal operation of the system.

1.4 Literature Review

1.4.1 Artificial Intelligence Techniques

Fuzzy Logic

Fuzzy logic was first theorised by Lotfi Zadeh [8]. The idea of fuzzy logic is to allow a computer, which operates on 1's and 0's to have an ability to reason and think based on data which is flawed and inherently ill defined. Fuzzy logic is used to give a computer the ability to reason by making a judgement of how true a condition is. The analogy often given to explain fuzzy logic, as outlined in [9], is that of temperature. One would generally describe temperature as cold, cool, mild, warm or hot and a person's decision is based on a number of factors such as temperature, wind chill and humidity, but importantly, it is not a definite answer of hot or cold as there are subdivisions in-between. If the temperature was 23°C, one may say that the temperature is warm but on the colder side of warm. Fuzzy logic tries to emulate this process of reasoning by creating a set of rules that can be followed to obtain an answer of how hot the temperature is and then determine an action to perform based on this information. Fuzzy logic follows five basic steps as outlined in [10], which are the following.

- *Pre-processing*

This is the manipulation of the measured data such as filtering, averaging, differentiation, integration or normalisation. It is simply used to obtain the data desired for use in the fuzzy logic controller. While not unique to fuzzy logic controllers, it is nonetheless important, particularly in fault-tolerant controller design.

- *Fuzzification*

Fuzzification is the process of taking the crisp data from the pre-processing, for example, a temperature, and converting it into degrees of membership. The process starts with the creation of a membership set made up of linguistic variables. This is a purely subjective process based on expert opinion from a knowledgeable person in the field being controlled. In keeping with the temperature example, a membership set could be the definition of "Cold", "Cool", "Mild", "Warm", and "Hot". The membership sets typically have the form shown in Figure 1.4; however, their shape may vary. The shape may be triangular, trapezoidal, or Gaussian. However, it is common to use triangular membership functions due to the reduced computational requirement. Typically, there would be some overlap of the membership set to avoid a measured variable being in no membership function. The fuzzification process is illustrated in Figure 1.4.

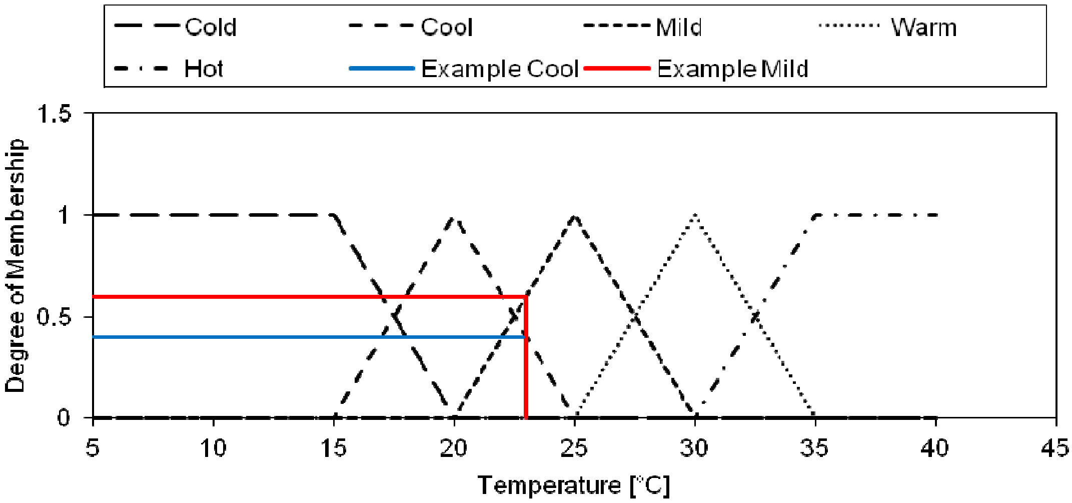


Figure 1.4: Diagram showing an example of membership functions for the fuzzy logic process with an example (23°C) of how to derive the degree of membership for a crisp number

The graphical example depicted in Figure 1.4 shows the determination of the degree of membership for an arbitrary input. Table 1.1 shows the complete degree of membership for each membership set for the chosen example temperature of 23°C. While it has occurred in this case, it is not a requirement for the degree of membership to sum to 1.

Table 1.1: Degree of membership for each membership set for a temperature of 23°C

Membership set	**Degree of membership**
Cold	0
Cool	0.4
Mild	0.6
Warm	0
Hot	0

- *Rule Base*

Once the data has been fuzzified, it can be applied to a set of rules. These rules, the rule base, allow the controller to determine the correct response to apply. The rules may have a single input as in this example or they may have multiple inputs thus allowing fuzzy logic controllers to be used with multi input and single input systems. Typically in fuzzy logic controllers, the error from a set point and the rate of change of error are used to control the output of the system. However, an accumulated error can also be used, in essence emulating a Proportional-Integral-Derivative (PID) controller.

The rule base is typically formulated through the use of linguistic rules. In other words, “If the temperature is COLD THEN the heater is turned onto maximum” or “If the temperature is HOT

THEN the cooling is turned onto maximum”. These rules are then tested based on the degrees of membership described above. Again for completeness a typical rule set for the temperature example is given in Table 1.2. It is noted that the rules are deliberately "Fuzzy", that is, the use of words such as "slightly" are encouraged.

Table 1.2: Rule base for the temperature example (aim to keep temperature "Warm")

Rule number	**If Temperature**	**Then**
1	Cold	Heating on maximum and cooling off
2	Cool	Heating on slightly and cooling off
3	Mild	Heating off and cooling off
4	Warm	Cooling on slightly and heating off
5	Hot	Cooling on maximum and heating off

A rule is said to fire when its condition is met. Thus following with the above example, rule number 2 and 3 are met to a certain degree. Thus rule 2 and 3 are said to fire with a firing strength of 0.4 and 0.6 respectively.

- *Inference Engine*

In the case of a single input system, there is no need for this step as a rule has fired with a certain firing strength. At this point a single input system would proceed to defuzzification, described below. For a multi input system where, for example, error and rate of change in error are used, an additional step before defuzzification is required. If, for example, the rule is, “*If the temperature is hot AND the temperature is decreasing THEN apply slight cooling*”, one needs to determine the degree to which each part of the rule contributes to the output. Thus, if the temperature was “Hot” to a degree of 0.9 and temperature was “Decreasing” to a degree of 0.1, it can be seen that the “Hot” is more dominant and should have more prevalence. In practical fuzzy logic algorithms, this is typically done using the AND and OR operators which correspond to minimum and maximum operators respectively. Thus, the firing strength of the example would be 0.9. and not 0.1. This would force more cooling to occur until the correct temperature is attained.

- *Defuzzification*

Defuzzification is the process of converting the results of the rules into a crisp output to be applied by the control actuator. In the example used, this would be the air conditioner/heater. Several methods exist for calculating the required output, these are:

1. Centre of gravity method

This method is the one of the simplest and most commonly used and looks at the degree of membership of each membership set and converts this to an area as shown in Figure 1.5. This is then converted to a single output value using a form of centre of gravity of the areas.

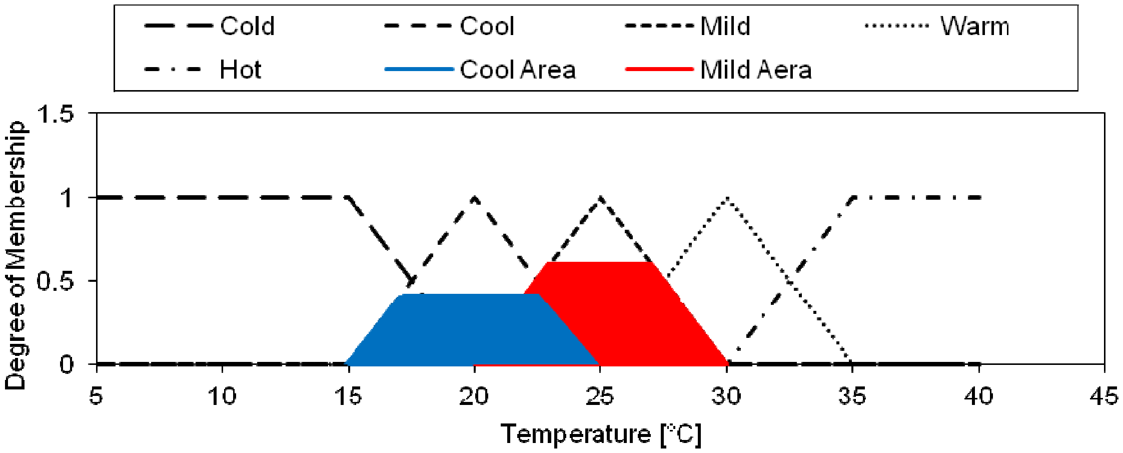


Figure 1.5: Diagram showing the centre of gravity method

2. Bisector of area method

This method finds the midpoint of the relative areas under the two graphs and uses this to find the output required. The method is computationally intense and can be ambiguous in the case of discrete values (singletons) [10].

3. Mean of maxima method

This method looks at the rules which fire with the highest strength and uses these to influence the output; however, this can lead to ambiguities. Instead the mean of the maxima is used. This method ignores the shape of the membership functions but has reduced computational complexity [10].

4. Left most maximum and right most maximum

This method chooses the left most maximum or the right most maximum but is more suited to robotics where a simple left or right decision must be made.

Neural Networks

Neural networks were inspired by the structures and processes found in the brain. The brain consists of billions of neurons which respond to inputs. These inputs are processed and passed along through synapses to the next neuron. Artificial Neural Networks (ANN) have a comparable structure and work in a similar manner to the brain. Inputs are passed to nodes and given weightings. From this, they are generally passed onto another level of nodes. Each node sums the inputs it receives until a certain threshold value is reached. Once the threshold value has been reached, the node will then output a predefined value, commonly “1”. The basic structure of a neural network is given in Figure 1.6.

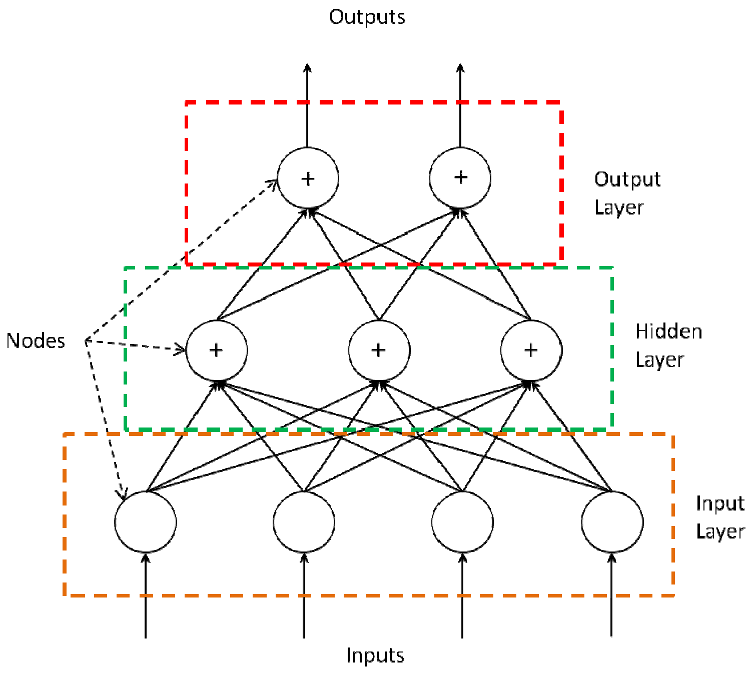


Figure 1.6: Basic structure of an artificial neural network

The weightings and threshold values give the ANN its characteristics and determine the output values from a set of inputs. Thus in order to obtain an ANN that is useful, these weights and threshold values need to be “learned” by the ANN; for simple problems this can be done manually. In general, ANNs will have far more nodes and layers than depicted in Figure 1.6. This means that the “teaching” of an ANN needs to be automated. This is achieved through various techniques such as back propagation, the use of genetic algorithms, gradient descent, amongst others[11]. With these techniques, there are two ways to teach an ANN. The first is incremental teaching, where some of the data is presented to the ANN and the weights adjusted, and the second is batch learning, where all of the data is presented to the ANN and the weights adjusted. There are problems in the training of ANNs in that the process is generally quite slow. This can be influenced by the type of learning mechanism and the rate at which the weights are adjusted. If the adjustment rate is low, the learning process will be slow, and if the learning rate is high, the values of the weightings may diverge. This presents a problem with regard to aircraft and fault-tolerant control, as a quick and reliable change in control parameters is required.

- *Back Propagation*

Back propagation is a learning mechanism that allows the ANN to tune its threshold and weighting values based on the inputs and required outputs. Back propagation starts with a random set of weights assigned to each of the hidden layers. The inputs are then inserted into the ANN and the outputs compared to the desired outputs. The weights and thresholds are then adjusted by looking at the effect of each node on the outcome. A learning rate is also specified that determines how quickly the ANN will adapt its weights. The higher this is, the faster the ANN will learn; however, if it is too large the weights will diverge. This process is propagated backward and a number of iterations run until the desired state is reached [11]. There are various back propagation algorithms such as gradient descent, conjugate gradient descent, Broyden-Fletcher-Gloldfarb-Shanno method, quasi-Newton, one-step secant, Levenberg-Marquardt and Bayesian regularization as described in [11].

1.4.2 Fault-Tolerant Design

Fault-tolerant design has seen much attention in recent times [12] due to the drive for safer and more reliable systems. Fault tolerance is particularly important where people’s safety is concerned, being particularly true in the aviation, nuclear and petrochemical industries [12]. The use of a more advanced control system would have undoubtedly resulted in a better outcome in many of the engineering failures of recent history. Perhaps the most famous example of where a fault-tolerant control system would have changed the outcome of a critical situation is the case of United Airlines 232, a McDonnell Douglas DC-10 flying from Denver, Colorado to Philadelphia, Pennsylvania. The aircraft suffered a catastrophic failure of the number 2 engine, mounted at the rear of the aircraft. The failure of the stage 1 fan disk resulted in the loss of hydraulic fluid in all three of the hydraulic systems, due to the engine fragments leaving the engine cowling. This resulted in the loss of the actuation of all control surfaces and subsequent loss of traditional control [13]. An off-duty training pilot provided assistance to the on-duty pilots and using differential thrust with the remaining engines was able to navigate to Sioux City Airport and perform a landing. Unfortunately, the right wing dipped moments before landing resulting in the aircraft tumbling on impact. 111 people were fatally injured but 185 people survived [13]. An almost identical situation developed on the 22 November 2003. An Airbus 300 cargo aircraft was departing Baghdad airport and was struck by a surface to air missile. Using a similar technique to the UA232 case, the pilot was able to land the aircraft without casualty, however the aircraft was severely damaged [14] . Due to accidents similar to this, a great deal of attention has been given to fault-tolerant systems that could prevent a situation similar to this occurring in the future.

Fault-tolerant design can be considered as a form of Systems Engineering, where a failure analysis of the system, coupled with good engineering practice, will result in a design that is tolerant to faults to a certain degree. This may include a system that is completely fault-tolerant, or simply a system that is failsafe. Thus the subject of fault tolerance is very broad and encompasses a variety of disciplines in order to achieve a truly fault-tolerant design.

1.4.3 Redundancy

Redundancy is simply the duplication of critical components and systems to ensure that a single point of failure cannot occur. Redundancy can, in general, be split into hardware redundancy and analytical redundancy.

Hardware Redundancy

In the areas of human safety, hardware redundancy may include redundancy of actuators, sensors and systems. For example, on many commercial aircraft such as the McDonnell Douglas DC-10, there are multiple hydraulic systems to ensure that should one system be compromised the aircraft is still controllable and safe [13]. Many aircraft also make use of multiple control actuators and various backup instruments and sensors to ensure that the failure of one component cannot cause the unsafe operation of the aircraft. Unfortunately, not all possible faults can be accounted for and many aircraft accidents have been caused when all of the redundancies have failed.

The main drawback of hardware redundancy is; however, the increase in hardware complexity, weight and in general a large financial implication. However, there are many advantages to hardware redundancy. This is commonly seen in modern fighter aircraft to increase manoeuvrability by using redundant control actuators in combination with the classical control actuators. In less safety critical industries, hardware redundancy is generally created through the use of redundant sensors which are less costly rather than more expensive systems and actuators, with the aim of minimising production losses [12], in essence creating a system that is fail safe, rather than fault-tolerant. Hardware redundancy, in some form, will always be required in order to achieve fault-tolerant design for systems that interact with a real world environment. Thus the use of hardware redundancy will be a vital aspect of the fault-tolerant UAS design.

Analytical Redundancy

Analytical redundancy is the use of software and computational techniques that check the system or make adjustments to the control laws to achieve either a fault-tolerant system or a failsafe system. Analytical redundancy can be as simple as sensor fusion, where two separate sensors are used in combination to act as a check of sensor accuracy or functionality, or it can be more complex.

Analytical redundancy can be achieved either passively, by having a set of pre-programmed actions to perform should a failure be detected, or actively using artificial intelligence techniques to identify and make decisions about what actions to make [12] to maintain the system performance. Active methods are generally considered superior to passive methods as they can potentially accommodate any fault of the system, whereas a passive system is limited to the number of faults that have been designed for and programmed.

1.4.4 Fault Detection, Diagnosis and Isolation

Fault detection, diagnosis and isolation is a strategy to give the controller some information about the status of the system as a whole. The strategy involves three steps as indicated by the title. Firstly, the controller needs to determine that a change in the characteristics of the controlled device has occurred or a component failure has occurred (Detection). Once this has been established, the fault or change needs to be identified (Diagnosis). And finally, the action to take needs to be enacted (Isolation).

There are multiple methodologies that can be used to achieve fault detection diagnosis and isolation. In the simplest case, it may be possible to measure directly the area of interest. For example, a car's engine temperature can be easily measured and a fault determined if that temperature is too high. The control actuator of the plant can directly be measured and compared to the output of the controller and simple thresholds used to determine that the part is functioning correctly. In the case where a direct measurement cannot be easily made, for example the output of a sensor, other techniques need to be used. These include model-based approaches, shown in Figure 1.7, where the output of the controlled device is compared to a set of fault models. Should the output match one of these models, a fault can be deemed to have occurred and preset action taken to mitigate the fault. Hypothesis testing can also be conducted to determine whether a fault has occurred. Difficulties arise in successfully detecting a fault as uncertainties within the analytical model and faults are difficult to differentiate in real world environments. [7]

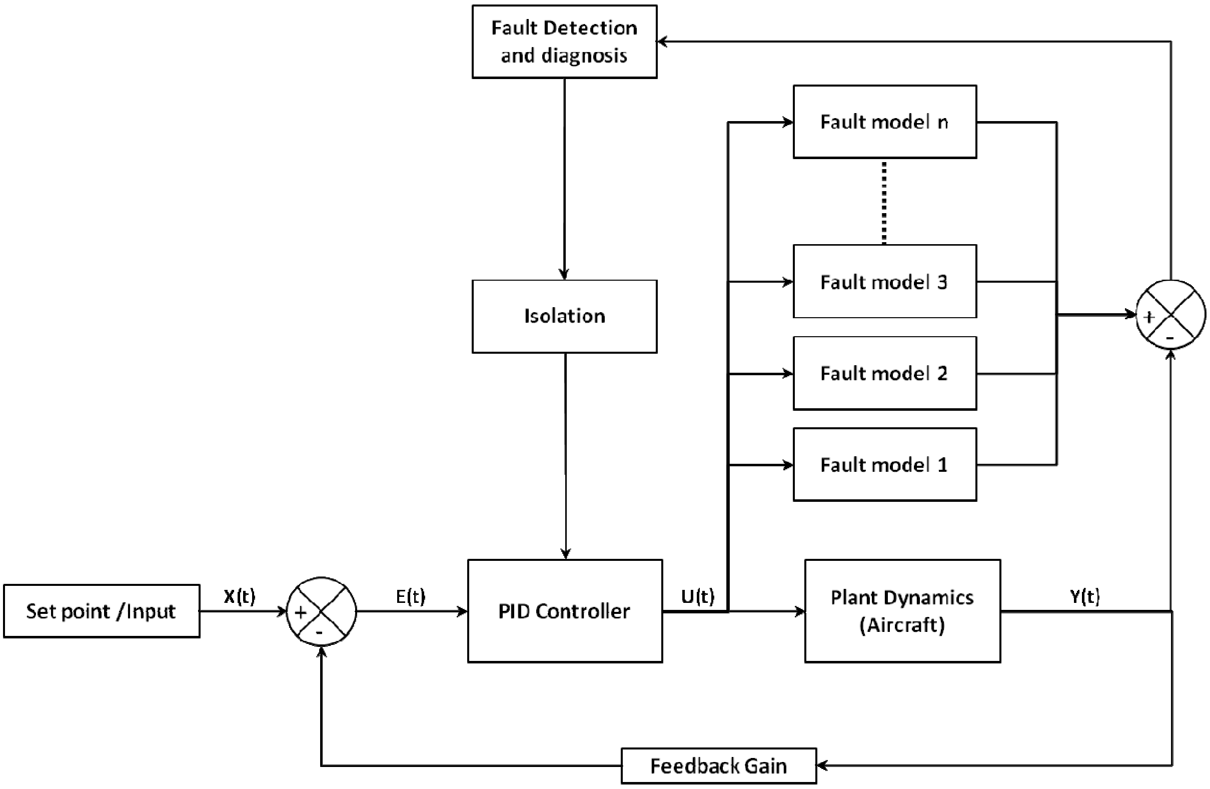


Figure 1.7: Example of a model-based approach to fault detection, diagnosis and isolation

Applications of Fault Detection, Diagnosis and Isolation

In [15], a fuzzy logic system was developed to detect, diagnose and then apply corrective action to a petrochemical process. The authors of this research suggest that because of the inherent characteristics of fuzzy logic, it can be applied to fault detection and diagnosis. It also has the advantage of being able to capture expert knowledge and thus it has a clear decision making process that can be followed by the user. A model-based approach was used in [15] to determine the state of the plant and whether a fault had occurred. Classical methods to do this exist and are used; however, they generally rely on very accurate models of the process to create residuals that allow the control actuator response to be modified. In [15], it is proposed that in order to overcome the need for a highly accurate model, that the algorithms used be robust against disturbances and uncertainties but sensitive to faults. The authors propose that a model-based technique be used, where the model is a fuzzy model for the process in normal operation. In other words, the predicted behaviour of the system is formulated using the “if-then” rules of fuzzy logic. The research was successful in that a fuzzy model of the process as created using real plant data that may or may not have had any faults and then applied to the system. Two faults were introduced and detected by the fuzzy logic algorithms.

1.4.5 Adaptive Control Techniques

Model Reference Adaptive Control (MRAC)

- *Overview*

Model reference adaptive control is a control technique that is closely based on system identification techniques. MRAC attempts to adjust the output of the controller in such a way as to force the plant to have certain dynamic characteristics. This is done by estimating the dynamic characteristics of the system based on the inputs supplied by the controller to the system and the measured response of the system. There are two primary techniques used in MRAC. The first is indirect MRAC, where the model parameters are explicitly determined and from these results the control laws are adapted to obtain the desired plant transfer function. The second method is the direct method, where the parameters of the plant are not explicitly determined but, rather, the control law is adapted to force the output of the plant to match that of a reference model. The difference is subtle; however, the indirect method is more closely related to systems identification techniques, whereas the direct method makes use of numerous minimisation techniques. Model reference adaptive control consists of three fundamental components. The first is the reference model which describes the ideal behaviour of the plant to an input. The second is the controller itself and the final component is the adjustment mechanism [16]. The adjustment mechanism changes the parameters of the controller using numerous numerical techniques such as the "MIT" rule, Lyapunov theory, and augmented error theory [16]. The basic idea of MRAC is shown in Figure 1.8.

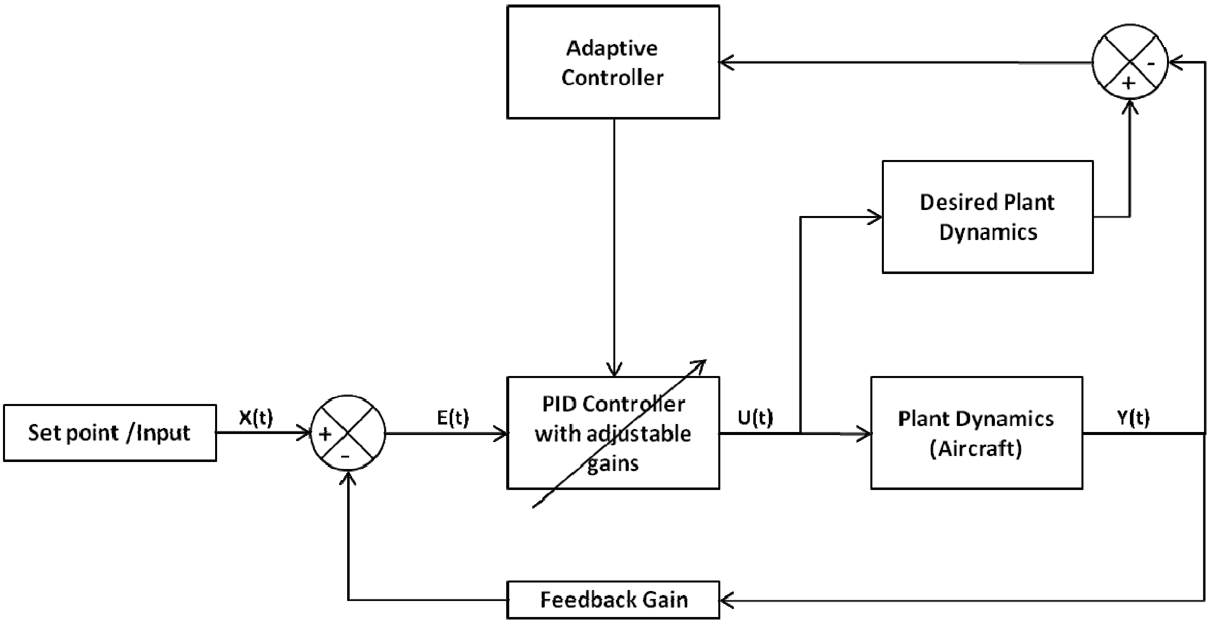


Figure 1.8: Control structure of a typical model reference adaptive control system

Model reference adaptive control has two distinct advantages. Firstly, MRAC can be used as a method of self tuning. The controller tunes the PID gains to drive the plant to a desired reference model. This property also gives MRAC another advantage over a traditional controller in that should the plant change, due to wear, actuator failure or structural changes, the MRAC will adapt the PID gains to achieve the desired performance. This has been successfully demonstrated in the adaption of an autopilot to changes in airframe. In particular, experiments with quad-rotor aircraft losing a fraction of one propeller blade have proved the ability of the model reference adaptive control technique to adjust the control laws to accommodate changes in airframes [17].

- *Applications*

Model reference adaptive control has successfully been used in numerous instances including UAS.

In [16] a model reference adaptive controller was applied to a second order system using the MIT rule and Modified MIT rule. The study varied the adaption rates of the controller and found that the selection of the adaption gain was important in creating a system that performed adequately. They found that by incrementing the adaption rates the performance of the system could be improved to a point, where after the performance degraded.

In [18], a model reference adaptive control was developed that made use of adaptive algorithms in both the feed forward and feedback paths. The aim was to control a not explicitly known system and force it to behave to a certain reference model. A PID controller, with adaptable gains, was placed in the feedback path of the system. The PID gains were adjusted online to minimise the tracking error between the ideal behaviour and the actual behaviour. It was concluded that the MRAC control used was able to track the reference model and was asymptotically stable. It was thus concluded that the same controller could be successfully used in multiple systems as the system tracks to a desired performance. Thus in theory this could also be used to adapt should the system change, such as in the case of a fault occurring.

In [19], the issue of stability in the adaptive gains was considered. In this research, the authors highlight the fact that, in a PID system with uncertainties, simple adaptation techniques may not result in the convergence of the PID gains to their ideal values, but instead the values may converge to an unknown and unpredicted value due to the inherent noise and time variance of the system. This may also result in a system where the PID gains do not converge at all but rather the adaptive controller may become unstable. The work presented in [19] aimed to investigate this phenomenon. The research concluded that the adaptive gains "...do converge to those bounded steady-state values that fit the particular input commands used in the specific application to guarantee perfect tracking." They found that generally the PID gains were lower than in the linear time invariant case and that this warranted future research.

Fuzzy logic controllers

Fuzzy Logic Control (FLC) can be used as a means of direct control. Commonly, this is simply another form of the classic PID controller; however, instead of using gains, a fuzzy inference system is used as shown in Figure 1.9. In other words, the input to the FLC is the error, rate of change of error and sum of error, from a desired set point. Fuzzy logic is then used to make a decision about the size and direction of the output, as previously described in Chapter 1.4.1. In many cases this has proved to be highly successful. While a direct adaptation mechanism is not explicitly present, the FLC is more robust than PID control and is more proficient in handling nonlinear dynamic systems.

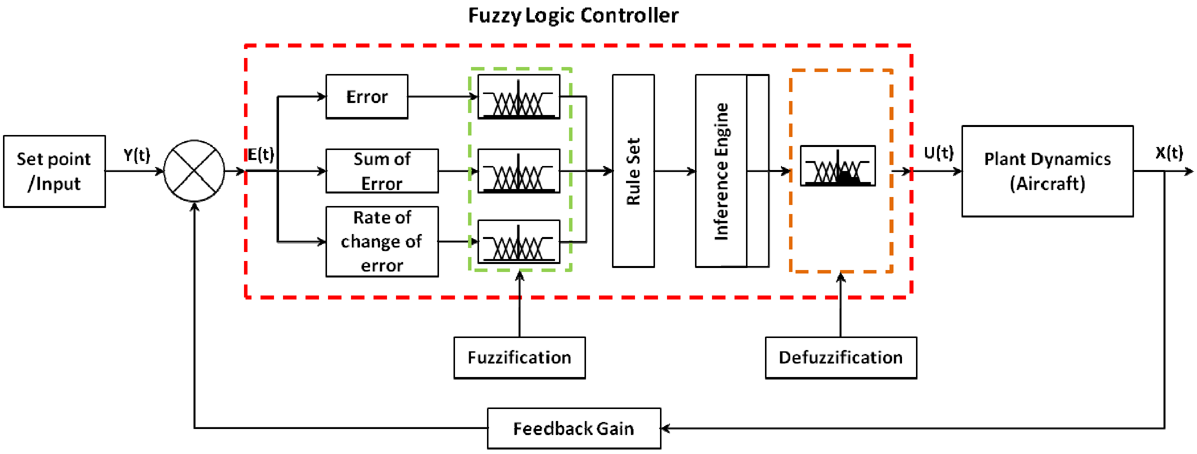


Figure 1.9: Block diagram showing the basic process of a fuzzy logic controller

In [20], a fuzzy logic controller was designed, simulated and compared to a Linear Quadratic Regulator (LQR) controller. It was found that the FLC had marginally better performance; however, the main advantage was that an explicit mathematical model of the system was not required as the actions of a human operator were modelled. This means that, in the presence of uncertainties or changes in the system, a FLC would perform better than a LQR controller.

In [21], a flight model, based on the Aerosonde UAS, was created and used in the development of a FLC for the UAS. The aim was to fly accurately a Standard Instrument Departure (SID) and a Tactical Air Navigation (TACAN) approach. In [22], the work was extended to include the autonomous landing of the Aerosonde UAS. The simulated results were highly successful and show that these manoeuvres could be successfully flown using a FLC.

In [23], a FLC was used to control the orientation and navigation of a simulated UAS. This proved successful; however, it was found that oscillations in altitude were present. It was thought that this was as a result of the membership functions defined in the simulations and that they were based on pilot experience rather than flight performance predictions. Their work continues into methodologies for improving these membership functions through the use of genetic algorithms.

In [24], a controller was required to control the longitudinal motions of two different general aviation aircraft, over their entire flight envelope. Typically classic PID controllers use gain scheduling to achieve this; however, when moved from one aircraft to another, a re-tuning of the control gains would be required. The aim of this research was to reduce/eliminate the need for gain tuning between aircraft models and reduce the need for gain scheduling through the application of fuzzy logic. A fuzzy logic controller was developed and successfully simulated on a Beech Bonanza type aircraft and on a small business jet type aircraft. It was found that the controller performed better on the piston driven aircraft; however, the performance of both simulations was satisfactory over the entire flight envelope and in different aircraft configurations. A note by the author was made regarding the process followed in the design of the FLC. It was found that the procedure to develop the FLC closely resembled the manner in which pilots are taught to fly. This shows the purpose of fuzzy logic in trying to create a system that mimics the way in which a human operator would think. The work in [24] was then expanded to also include lateral motions in [25] and was also highly successful.

Adaptive PID controllers using Fuzzy Logic

Fuzzy logic has also been implemented as a means of fault-tolerant control. In a similar methodology to model reference adaptive control, the gains of a classic PID controller are adapted in real time by a fuzzy logic control system as shown in Figure 1.10. The fuzzy logic system allows for a rule-based adaptation which is more intuitive and more transparent than model reference adaptive control. The use of fuzzy logic in this system also allows for the inclusion of other inputs that can aid in adapting the PID gains of the controller.

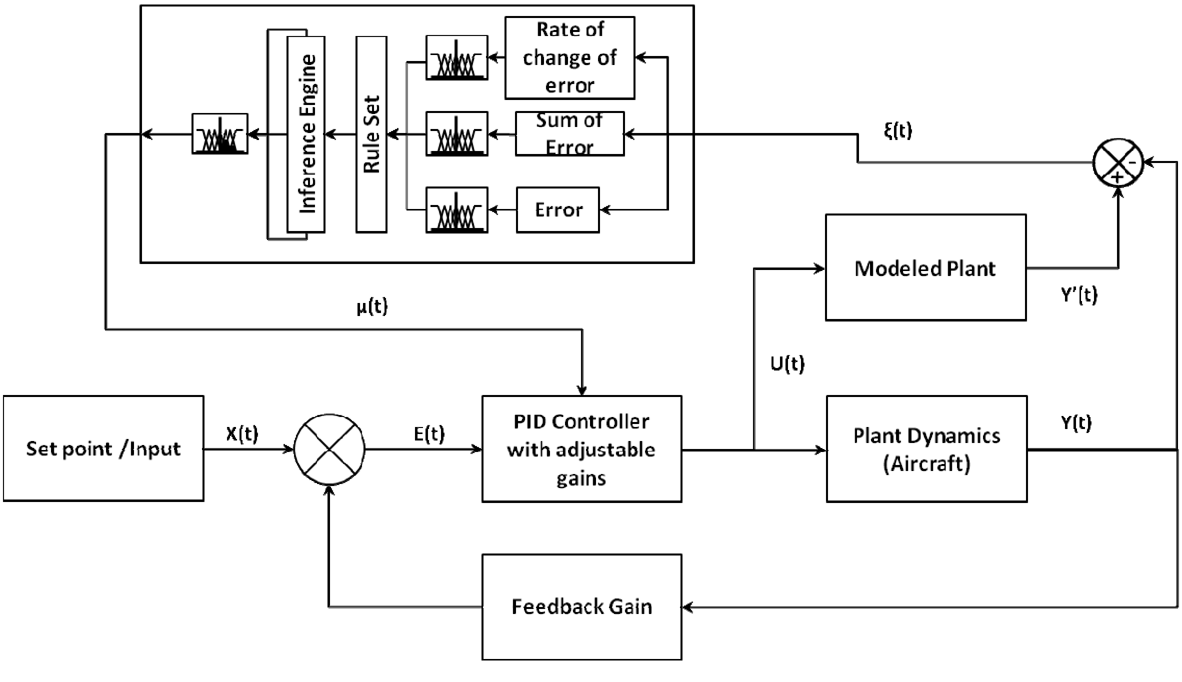


Figure 1.10: Model reference control using fuzzy logic

In [26], a fault-tolerant fuzzy logic controller was designed for a quad-rotor UAS. The UAS made use of classical PID control to maintain position, orientation and altitude. A fuzzy logic system was then used to adapt the PID gains to achieve a required control performance in a similar manner to model reference adaptive control. The quad-rotor UAS had two faults to adapt to, namely, the loss of effectiveness of all actuators (rotors) and the loss of control effectiveness in one actuator. The UAS was tested using a Gumstix microcontroller at 200Hz. It was observed that the fuzzy adaptive PID controller reacted faster than the PID controller to return the UAS to the desired set point after the fault was introduced. The fuzzy PID controller also required less tuning as the gains were adapted real time by the fuzzy logic system.

In [27], the development of a fuzzy logic system similar to the type in [26] was developed. However, this system focused more on a reduction in gains based on sensor measurements that indicate a fault. Thus the system was designed to protect the airframe should a fault occur by ensuring that the controller's outputs are more conservative. They were successful in their design and noted that the algorithms were not computationally intense and as such favoured miniaturisation. They also noted that a hybrid neuro-fuzzy control system would be ideal; however, because of the uncertainty as to how neural systems obtain their decisions and the fact that the training of neural networks can be too slow for online applications, they concentrated only on the fuzzy logic system.

Fuzzy Model Reference Learning Control (FMRLC)

The next advancement in fault-tolerant control is a mechanism where the control algorithm can "learn" based on the outputs it sends and inputs it receives. A typical block diagram of this is given in Figure 1.11. This is in essence the same as the previous model reference controllers; however, the adaption controller changes the membership sets of the FLC in real time.

In the research conducted in [28], a fuzzy model reference learning controller was developed for an F-16 fighter aircraft. This controller was used to reconfigure the F-16 flight controller after a fault had occurred. The research was extended to include fault detection and diagnosis which improved the overall performance of the system by reducing the performance capabilities of the aircraft when a fault was detected. This ensures that while parts of the mission are compromised, the aircraft is still safe to fly and can return to a safe airfield. It follows a similar methodology to that of model reference fuzzy control, where a reference model is used to adjust the controller performance; however, the difference is that the controller is a fuzzy logic controller. Thus the controller is learning and adapting to achieve a desired, preset performance. While the simulations proved successful, the authors noted that more work needs to be conducted on non-linear models, that a comparison between these techniques and conventional model reference techniques should be conducted and that the stability of the system should be tested.

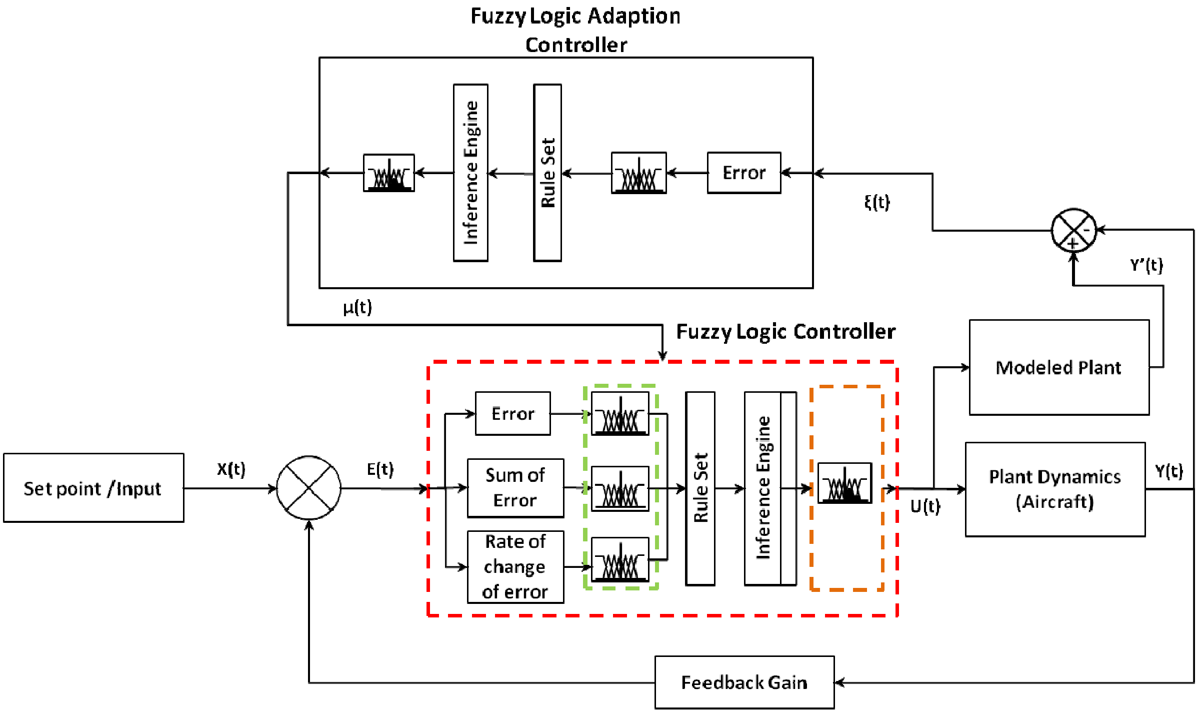


Figure 1.11: Control structure of a fuzzy model reference learning controller

In [29], a fuzzy model reference learning control (FMRLC) technique was developed that changed the values and ranges of the fuzzy logic controller membership functions based in the input-output relationship of the controller. The controller then tried to minimise the difference between the output of the system and a predetermined reference model. The aim was to create a controller capable of adapting to a change in the system properties. This controller was then applied to a time varying rocket velocity control problem and a multi input multi output two degree-of-freedom robotic arm. It was found that the advantage of the FMRLC technique was that a detailed analytical model of the plant was not a requirement and it provides the advantage of having an expert knowledge base while ensuring the required performance of the system.

Neural Networks in Adaptive Control

In [30], research was conducted into the control of a formation of quad rotor UAS aircraft. The methodology used is that of a leader–follower, where one UAS is deemed the leader and the others maintain their position relative to the leader UAS. A neural network, consisting of two hidden layers, was used to control this simulated formation. The first hidden layer consisted of randomly assigned weights, while the second layer consisted of tuned weights. Unfortunately, no mention of the tuning procedure was made as only the framework was presented.

In [31], the design of a fault-tolerant UAS making use of neural networks was investigated. In particular, it was desired to have a system that could accommodate a loss in propulsion and by using the remaining propulsive thrust maintain safe operation. The research was based on previous work in the field of fuzzy logic, where fuzzy logic was used in the simulation of formation flight of a pair of UAS. The research in [31] proposed a neuro-adaptive control system that could accommodate "...fading jet-engine power, while still maintaining desirable performance and stability properties." This would be conducted using a neural network where the weights are adjusted in real time based on the faults detected. The research showed promising results in that the simulations were reasonably successful. It was noted by the author that the effectiveness of the controller was improved when more than 15 input nodes were used and increasing this resulted in better performance. However, when more than 25 nodes were used, chattering of the control outputs started to occur. This research demonstrated the ability of a neural network to learn in real time and adapt to a fault in a UAS. However, no mention was made about the computational requirements for such a system.

1.4.6 Reconfigurable Control - Control Allocation

Overview

Reconfigurable control differs from adaptive control in that control for one objective is apportioned to several control effectors. In the case of an aircraft, for example, ailerons are traditionally used for roll control; however, the use of differential flaps, rudder and differential elevator can be combined to create a more effective roll command. Research into reconfigurable control began as a method of increasing manoeuvrability of advanced fighter aircraft and has developed into an extensive area of research with many differing opinions and techniques. The technique identified as being the most promising with regard to actuator faults is that of control allocation. The control allocation problem, as it is known, essentially involves the determination of control deflections to create a desired moment about a particular axis. This is generally accomplished by placing the control allocation algorithm between the controller and the actuators as shown in Figure 1.12. Thus no change to the control law is made. Instead the output of the control law is distributed through the control allocation mechanism to all of the control effectors.

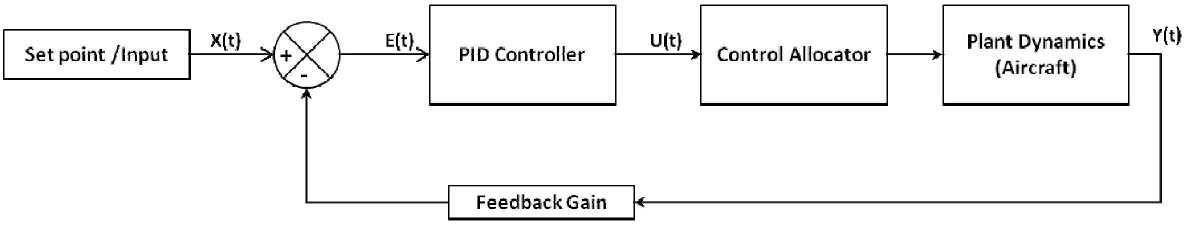


Figure 1.12: Block diagram showing control allocation and its relation to aircraft control

Several control allocation techniques exist. These are classified into non-optimal and optimal techniques.

The non-optimal techniques include:

- Generalised inverse
- Daisy Chaining
- Cascaded Generalised Inverse
- Multi-pass Inverse Methods

The optimal techniques include:

- Facet Searching
- Edge Bisector
- Optimisation techniques

Non- Optimal Techniques

- *Generalised Inverse*

The generalized inverse technique is the simplest mathematical approach used and is essentially the solution to the state space problem

$$y_d = Bu \qquad (1.1)$$

where y_d is the output of the control law, B is the control effectiveness matrix and u is a matrix of individual control actuators [32]. Thus the generalized inverse method simply solves for u using

$$u = B^{-1}y_d \qquad (1.2)$$

This assumes that there are no bounds on the control actuators and that the relationship is linear. While this method is relatively straight forward, it requires that the control effectiveness matrix B be known to a reasonable degree of accuracy. It also has the disadvantage that no allowance is made for control actuator saturation in a practical system.

- *Daisy Chaining*

Daisy chaining is a priority driven control allocation technique where the control effort is assigned to all of the control actuators through a prioritised order. The control effect receives a desired control effort which is distributed to the primary control actuators for that command until either the control effort is obtained, or until the control actuator saturates [32]. Should the primary controls saturate, the secondary control actuators will be activated to achieve the balance of the control effort required. This continues until either the control effort is obtained, or all of the control actuators are saturated.

- *Cascaded Generalised Inverse*

The cascaded generalised inverse is similar to the generalised inverse; however, it takes the idea of daisy chaining and assigns the desired control effort to the primary actuators and then assigns the residual control effort, given in (1.3) [32], to the next set of control actuators.

$$y_{residual} = y_{desired} - B_1 u_1 \tag{1.3}$$

This continues until the residual is driven to zero. In essence this technique is a more formalised approach to daisy chaining. However, again, the control effectiveness matrix is required in order to distribute the control effort effectively.

- *Multi-pass Inverse*

The multi-pass inverse technique is a combination of those techniques described above. The control effort is distributed to the controls through a generalized inverse method; however, should the control surface saturate the next set of control actuators are incremented until the control effort required is achieved. It was suggested in [32] that the use of a direct measurement of control deflections would aid this method as the deflection of the control surface would be known and would not need to be inferred.

Optimal Techniques

The optimal techniques are based on the determination of a solution within the control actuator deflection space. The idea is to map the possible solution space of all control effectors to an m-dimensional volume, where m is the number of control actuators. The optimal solution then occurs on the surface of this volume [33]. The optimal techniques attempt to determine this point for any required control effort.

- *Facet Searching*

This is a direct allocation technique that attempts to find the optimal solution to the control allocation by brute force [33]. The technique develops edges defined by two controls and then attempts to determine whether the edge intersects a line within the solution volume that is in the direction of the desired moment [33]. This method then iterates through numerous combinations until a solution is obtained. Thus this method is very computationally intense [32].

- *Bisecting edge search*

This method was developed in an attempt to reduce the computational requirements of the facet searching method. Essentially it attempts to find a solution that converges onto the edge of the solution volume but over a fixed number of iterations [32]. It does this in a similar way to that of facet searching but instead of examining the entire solution space, the direction generated by the developed edge is noted, thus allowing for a quicker convergence to the optimal solution. In [32], it was noted that this method was up to four times faster than the facet searching methods.

- *Optimisation Techniques*

Other optimisation techniques such as quadratic programming and linear programming have also been used to minimise a secondary objective related to control the deflections such as structural loadings, as seen in [34]. Normal optimisation techniques are utilised to solve the generalized inverse problem with the added constraints imposed, thus creating a solution that is simpler to understand and more optimal than the non–optimal techniques [32].

1.5 Identified Gaps in the Literature

There is a fair amount of research that exists regarding the adaption of PID gains using either conventional techniques or artificial intelligent techniques. However, there are relatively few cases where these have been applied to an operational UAS. Furthermore, in many circumstances where aircraft are used, only one axis is considered. Thus longitudinal and lateral motions are viewed separately. Another gap identified is in the use of a combined fault-tolerant strategy, where multiple fault-tolerant techniques are used in combination.

1.6 Rationale and Motivation

The loss of an UAS, while not as critical as the loss of a manned aircraft, is highly undesirable. This may be due to the high cost of the UAS itself, the cost of the payload it may carry, the loss of confidentiality of the UAS or the information it contained, or, in the civilian context, the safety risks for people on the ground. UAS aircraft are often subjected to harsh and hostile operating environments which put the airframe at risk. These environments may cause a variety of faults that include sensor failures, actuator failures and, in more extreme cases, combat damage. A current limitation of UAS comes from the inability of the autopilot systems to accommodate for these faults or uncertainties in the system performance. Significant effort is made in the design of a robust control system that is insensitive to a certain degree of failure, but there is still a desire to have a system that can tolerate more significant faults. It would also be desirable to have a system capable of operating outside the current limited flight envelope and to have a system capable of recovering from unusual attitudes.

In many commercial autopilots used in UAS, flight control is achieved through the use of classical control methods such as Proportional-Integral-Derivative (PID) control. This control system, also found in many other commercial and industrial systems, is highly effective in basic UAS control.

However, two main drawbacks are evident with PID control systems, the first being the tuning of the PID gains used in the control loop and the second being the inability of PID control to cope with large changes in the system performance. These changes may stem from differences in UAS geometry or from a failure of any number of systems on board the UAS, and, if large enough, may result in the inability of the PID controller to control the UAS adequately.

The proposed research is to explore various methodologies of fault-tolerant control design, including the use of artificial intelligence techniques for a UAS aircraft. The aim is to develop a control strategy that makes use of artificial intelligence techniques and apply this strategy to a known UAS. In the first instance, the control strategy will be tested through simulations with the intention of implementing this research into an UAS in the future.

Fault-tolerant systems are not entirely unique and several systems been successfully demonstrated. A noteworthy example of this is the work conducted by Rockwell Collins. They have successfully flown a model of a Boeing F-18 UAS and ejected up to 80% of one wing in one test and 60% of the wing, 30% of the horizontal tail and 30% of the vertical tail in another. The UAS recovered from the flight surface loss in both cases and performed a safe landing [35]. Other successful demonstrations of fault tolerance have been conducted on smaller UAS by a number of universities and research groups [17], [36].

1.7 Research Objectives

The problem statement for this research topic is as follows.

Research, develop, simulate and test a fault-tolerant UAS autopilot system for use on the CSIR's Modular UAS.

Due to the nature of the work at the CSIR, the most common fault expected is that of control actuator failure. This may be due to a malfunction with the servo motor, electromagnetic interference, an oversight in assembly or the disconnection of the servo in flight. Another area of concern is in the loss of a control surface, due to flutter, or other structural damage. Thus this will be the primary focus of the research. However, as a secondary objective, it is desirable to have a system that is capable of tolerating changes within the aircraft as in many instances the UAS is modified for a particular mission. Finally, sensor failure is of concern; however, this is not considered the primary focus of the project, but will be borne in mind during the research.

The objective of the proposed project is thus to research, design, simulate and then, should the results be successful and time permitting, apply a fault-tolerant system into a commercial autopilot system. The detailed objectives of the proposed research are to develop an autopilot that is tolerant to the following.

- Actuator failure
- The loss of a control surface
- Uncertainties within the aircraft model

It is desirable to focus on artificial intelligence techniques as; in general, they are more suited to non-linearity and uncertainties within the system being controlled. However, more conventional techniques will not be ignored where they are considered to be superior.

1.8 Research Scope, Strategy and Methodology

The methodologies proposed in this dissertation were chosen based on the advantages and disadvantages of the various techniques described in the aforementioned sections, while being aware of the limitations of the chosen autopilot and airframe. The strategy proposed for this research is to combine several of the ideas of previous work in to a hybrid of all of the main techniques. However, because of the limited processing power of the autopilot, many of the techniques proposed are not in their most advanced form, as in general these have been found to be highly complex and computationally expensive. Thus a more practical methodology is proposed.

The methodology proposed is a three pronged approach to fault-tolerant UAS controller design. The start of this is simple robust design. This will include both software and hardware redundancy. This is already catered for in the design of most airframes in the form of multiple control actuators and in the case of the airborne platform modelled in this research; this will include 7 independent control surfaces and 2 independent engine controls.

The second part of the strategy involves the idea of reconfigurable control and, in particular, in control allocation techniques. At this stage, a non-optimal technique, such as daisy chaining or a weighted inverse approach, will be used. Because of the uncertainties in the aircraft, a robust technique that can be easily interfaced with the current controller will be chosen. This robustness is also required to accommodate unknown actuator failures as well as uncertainties in the resulting control effectiveness matrix. The non-optimal techniques have been compared in [32] and found to have a good performance when compared to the optimal methods. In [32], it was stated that no noticeable difference was seen in the non-optimal and optimal techniques. A non-optimal solution can also be motivated for because the project is based on fault-tolerant UAS design rather than the optimisation of control deflections for manoeuvrability purposes. The computational requirement must also be considered to ensure a rapid response to a failure.

The third part of the control strategy is the more complex of the strategies and includes a fuzzy logic adaptation mechanism of the PID gains of the controller. This will increase the reaction time should a fault occur and help to maintain the desired performance. This will be based on an ideal reference model for the aircraft to follow. This also improves on the current system in that the PID gains will be adjusted to optimise the controller performance during normal flight. It is expected that this system will be the most computationally expensive as a fuzzy logic inference system will need to be developed for roll, pitch and yaw axes as well as heading, airspeed and altitude control. The high level diagram indicating the proposed control strategy for this dissertation is shown in Figure 1.13.

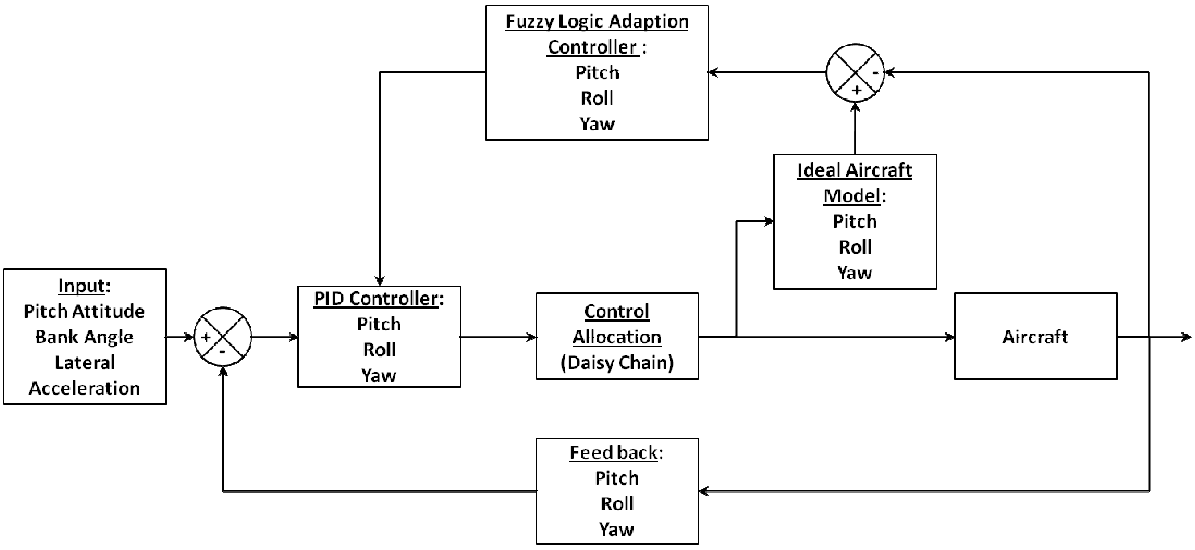

Fuzzy Logic Adaption Controller : Pitch Roll Yaw
Ideal Aircraft Model: Pitch Roll Yaw
Input: Pitch Attitude Bank Angle Lateral Acceleration
PID Controller: Pitch Roll Yaw
Control Allocation (Daisy Chain)
Aircraft
Feed back: Pitch Roll Yaw
+
-
+
-

Figure 1.13: Proposed solution to fault-tolerant UAS controller.

This particular approach, although using the basic algorithms of each fault-tolerant area, is unique in that it will apply a variety of different techniques to create a system that is tolerant against actuator faults, and possibly to a certain extent structural damage. The advantage of this approach is the modularity that it gives. This will allow for the improvement of each individual module as the autopilot hardware develops and an understanding of the computational requirements of each technique is gained.

1.9 Research Contributions

Novelty for this research exists in:

1. The high fidelity of the simulation model to be used. Wind tunnel data of the UAS will be used with a full 6 Degree-of-Freedom, model that accounts for asymmetries in the aircraft size and mass.
2. The fuzzy logic adaption algorithm will simultaneously be applied to both longitudinal and lateral aircraft motion.
3. A combined system of control allocation and control adaption will be used
4. Faults in all aircraft axes will be generated, both singularly and combined. Thus the control system will be required to not only tolerate faults in the longitudinal axis and lateral axis singularly, but for all axes collectively.

1.10 Layout of Dissertation

This dissertation has the following structure:

- Chapter 2 - Development of a high fidelity, 6DOF, nonlinear flight dynamic model.
- Chapter 3 - Development of a Autopilot model
- Chapter 4 - Development of fault-tolerant control model
- Chapter 5 - Simulation of Control actuator faults
- Chapter 6 - Discussion of Results
- Chapter 7- Conclusions and recommendations

This dissertation covers the initial design and simulation of a fault-tolerant control system of a UAS. It begins with the development of a high fidelity, 6 degree-of-freedom (6 DoF), nonlinear flight dynamic model of the CSIR's Modular UAS. This model is a time domain based model that makes use of the wind tunnel data of a full scale model of the UAS. This is followed by the development of an autopilot model, based on the Ardupilot Mega architecture but simplified to extract the necessary control laws. The development of the fault-tolerant system follows. The use of fuzzy logic in PID adaption as well as the reconfiguration of control surfaces is outlined. A detailed simulation of the system is conducted and the results and presented in Chapter 5. The results will be further discussed in Chapter 6 and finally a conclusion as to the success of the fault-tolerant control system is presented in Chapter 7. Additional flight dynamic details and models are described in Appendices A and B and the full simulation code, fault-tolerant control code and a full set of results can be found in the digital Appendix C.

2 Chapter 2 – System Description and Mathematical Modelling

2.1 Chapter Outline

Typically, when performing research into control, a plant is identified and the development of a physical model representing the system is developed. This model is created by first finding a point at which the system is in equilibrium and then examining the changes in these forces due to the dynamics of the plant. This results in a set of equations that, when linearised, provide a relatively easy way of obtaining the response of the system to a disturbance or change in set point about the equilibrium condition. This works well for many systems, as their behaviours are often close to linear for small accelerations, angles and control inputs. However, for this research the assumption of small accelerations, small control inputs and small angles cannot be guaranteed due to the very nature of the problem under examination. Thus, the development of a model that considers non-linearity is vital to determine the ability of the control algorithms developed to handle various faults. This chapter outlines the development of such a model. The chapter will start with a brief overview of the intended aircraft to be modelled, the CSIR's Modular UAS, and will then follow with the simulation strategy and detail the various techniques and methods used in the flight dynamic model. The relevant wind tunnel data will also be presented.

2.2 CSIR's Modular UAS Airframe

The airborne platform to be used in this research is the CSIR's Modular UAS, shown in Figure 2.1. The airframe was developed with the intention of being used for control research and, in particular, fault-tolerant control research. Its configuration was thus tailored to provide multiple redundancies. The airframe consists of two fuselages joined with a centre wing section and a payload pod suspended beneath the centre wing. There are two elevators, ailerons and two rudders that are all driven independently. Flaps inboard of the ailerons but outboard of the fuselages are also present.

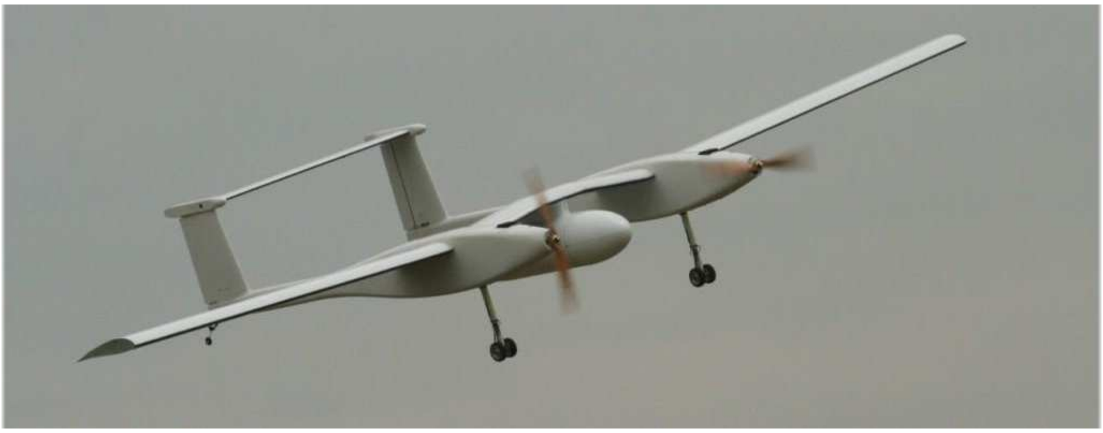


Figure 2.1: The CSIR's Modular UAS on final approach

A significant advantage of using this airframe not only lies with the multiple control surface redundancies, but also in the set of wind tunnel data available. This allows for the development of a high fidelity flight dynamic model that accounts for the non-linear effects seen at high angles of attack and high sideslip angles. Unfortunately, the dynamic characteristics of the aircraft were not obtained in the wind tunnel, and as such will be derived in the sections that follow. The geometry and key specifications of the Modular UAS are provided in Table 2.1, for reference.

Table 2.1: Table detailing the geometry of the CSIR's Modular UAV

Parameter	Symbol	Value	Unit
Mass and Inertia			
Mass	$MTOW$	26.382	kg
Moment of inertia about x-axis	I_x	11.106	kg.m^2
Moment of inertia about y-axis	I_y	7.900	kg.m^2
Moment of inertia about z-axis	I_z	18.455	kg.m^2
Product of inertia about x and y axes	I_{xy}	0.000	kg.m^2
Product of inertia about x and z axes	I_{xz}	0.836	kg.m^2
Product of inertia about y and z axes	I_{yx}	0.000	kg.m^2
Wing			
Area	S_{Ref}	1.482	m^2
Root chord	c_{W_R}	0.359	m
Tip chord	C_{W_T}	0.359	m
MAC	$\bar{c}_W$	0.359	m
Span	b_W	4.130	m
Aspect ratio	AR_W	11.509	N/A
Oswald’s factor	e_W	0.85	N/A
Horizontal Tail			
Area	S_{HT}	0.200	m^2
Root chord	c_{HT_R}	0.200	m
Tip chord	C_{HT_T}	0.200	m
MAC	$\bar{c}_{HT}$	0.200	m
Span	b_{HT}	1.000	m
Aspect ratio	AR_{HT}	5.000	N/A
Oswald’s factor	e_{HT}	0.9	N/A
Vertical Tail			
Area	S_{VT}	0.1689 x2	m^2
Root chord	c_{VT_R}	0.2735	m
Tip chord	C_{VT_T}	0.210	m
MAC	$\bar{c}_{VT}$	0.243	m
Span	b_{VT}	0.348	m
Aspect ratio	AR_{VT}	0.717	N/A
Oswald’s factor	e_{VT}	0.9	N/A

2.3 Flight Dynamic Simulation Model Strategy

In essence, the approach for this model is relatively simple as is shown in Figure 2.2; however, the complexity presents itself in the detailed development of the model. The high level strategy is to determine all of the forces acting on the aircraft at any given state. Based on these forces, the acceleration of the aircraft about all three axes will be determined. The accelerations are then integrated over a small time increment and the state of the aircraft is updated. This process is then repeated, storing the results of the state, until the time period of interest has lapsed. This methodology, similar to that used by real time flight simulators, has several advantages over traditional control theory methods. This method allows one to model the non-linear behaviour of the aircraft and controls, as well as incorporate a more realistic representation of autopilot control strategies and additional control strategies, such as the control strategy to be developed for this dissertation.

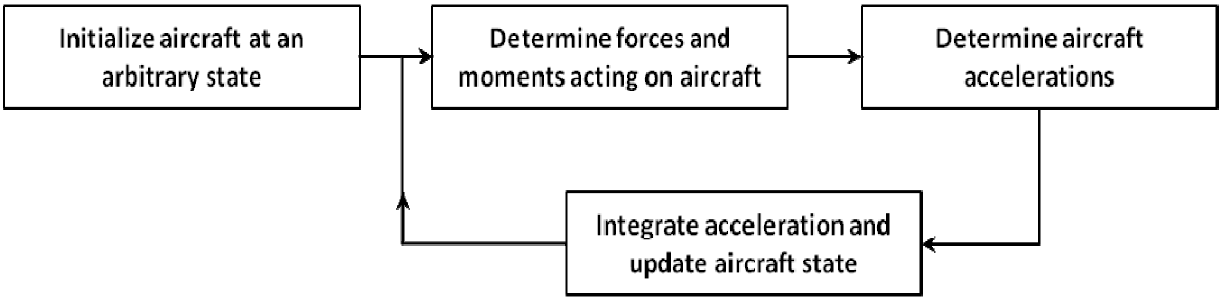

Initialize aircraft at an arbitrary state
Determine forces and moments acting on aircraft
Determine aircraft accelerations
Integrate acceleration and update aircraft state

Figure 2.2: High-level flight dynamic model strategy

2.4 Coordinate Systems

Traditionally in flight dynamic analyses, there are three main coordinate systems used. The use of one over another is largely dependent on the aspect under examination. For this simulation, three coordinate systems will be required. Firstly, an Earth-fixed coordinate system will be used, as shown in Figure 2.3. This coordinate system will assume a flat earth, and will be centred about an arbitrary location on the earth surface, with its x-axis positive to the North, y-axis positive to the East and z-axis pointing downwards. This system will be used to describe the orientation of the aircraft relative to the earth, and will provide a reference for the navigational algorithms of the autopilot.

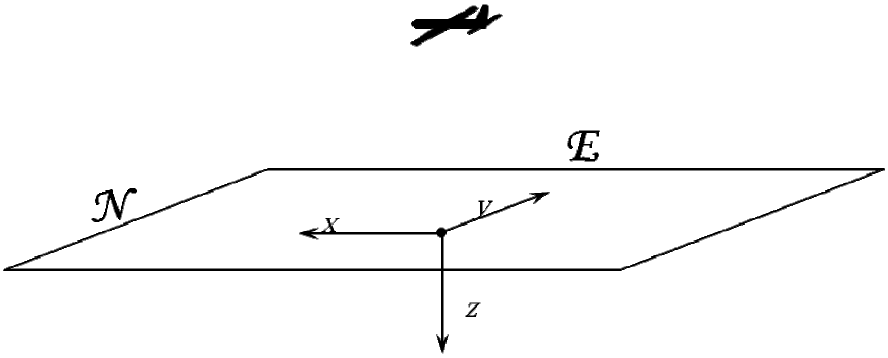


Figure 2.3: Diagram showing the conventions used for the earth-fixed coordinate system

The second axis system to be used is the body-centred axis system, shown in Figure 2.4. This set of coordinates is fixed to the aircraft. The origin of this axis system is located at the aircraft centre of gravity, with the x-axis being positive towards the fore of the aircraft. The y-axis is positive to the starboard side of the aircraft and, to maintain consistency, the z-axis is positive towards the floor of the aircraft. This is the primary axis through which the accelerations are determined and upon which the equations of motion are based.

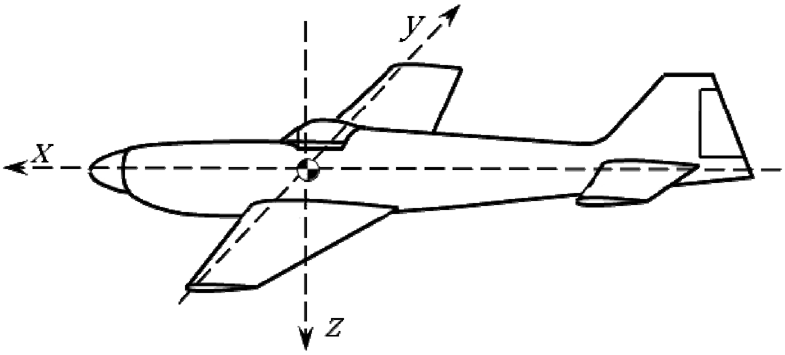


Figure 2.4: Diagram showing the conventions used for the body-fixed coordinate system

Lastly, a wind axis system is also used throughout the flight model. This is needed as the wind tunnel data available is referenced to the wind tunnel axis. The wind axis has a similar sign convention to that of the body-centred axis and is centred about the aircraft centre of gravity, but is rotated away from the body axis by the angle of attack and sideslip angle, resulting in a system that is aligned with the local flow direction.

During the flight simulation, transformation from one axis system to the other will be required. These transformations are well understood and are formulated by rotations about each of the axes in turn. The order in which this occurs is crucial to the transformation. The transformations used, obtained from [37], are presented as Eqs. (2.1) and (2.2).

-Transformation from wind axis system to body-fixed axis system:

$$\begin{bmatrix} X_b \\ Y_b \\ Z_b \end{bmatrix} = \begin{bmatrix} \cos\alpha\cos\beta & -\cos\alpha\sin\beta & -\sin\alpha \\ \sin\beta & \cos\beta & 0 \\ \sin\alpha\cos\beta & -\sin\alpha\sin\beta & \cos\alpha \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} \tag{2.1}$$

-Transformation from body-fixed axis system to Earth-fixed axis system:

$$\begin{bmatrix} X_E \\ Y_E \\ Z_E \end{bmatrix} = \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\phi - \sin\psi\cos\phi & \cos\psi\sin\theta\cos\psi + \sin\psi\sin\phi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\phi + \cos\psi\cos\phi & \sin\phi\sin\theta\cos\phi - \sin\psi\cos\phi \\ \sin\theta & -\cos\theta\sin\phi & \cos\theta\cos\phi \end{bmatrix} \begin{bmatrix} X_b \\ Y_b \\ Z_b \end{bmatrix} \tag{2.2}$$

2.5 Aircraft Equations of Motion

The equations of motion for any rigid body about its centre of gravity and with no planes of symmetry are given in Eqs. (2.3) to (2.8). These equations were developed from first principles in [37], and form the backbone upon which the flight dynamic model was created.

$$m(\dot{U} - rV + qW) = X \tag{2.3}$$

$$m(\dot{V} - pW + r) = Y \tag{2.4}$$

$$m(\dot{W} - qU + pV) = Z \tag{2.5}$$

$$I_x\dot{p} - (I_y - I_z)qr + I_{xy}(pr - \dot{q}) - I_{xz}(pq + \dot{r}) + I_{yz}(r^2 - q^2) = L \tag{2.6}$$

$$I_y\dot{q} + (I_x - I_z)pr + I_{yz}(pq - \dot{r}) + I_{xz}(p^2 - r^2) - I_{xy}(qr + \dot{p}) = M \tag{2.7}$$

$$I_z\dot{r} - (I_x - I_y)pq - I_{yz}(pr + \dot{q}) + I_{xz}(qr - \dot{p}) + I_{xy}(q^2 - p^2) = N \tag{2.8}$$

where, U, V and W represent the linear velocities in the x, y and z axes, respectively and p, q and r are the rotational velocities of roll rate, pitch rate, and yaw rate, respectively. The positive direction of each is indicated in Figure 2.5.

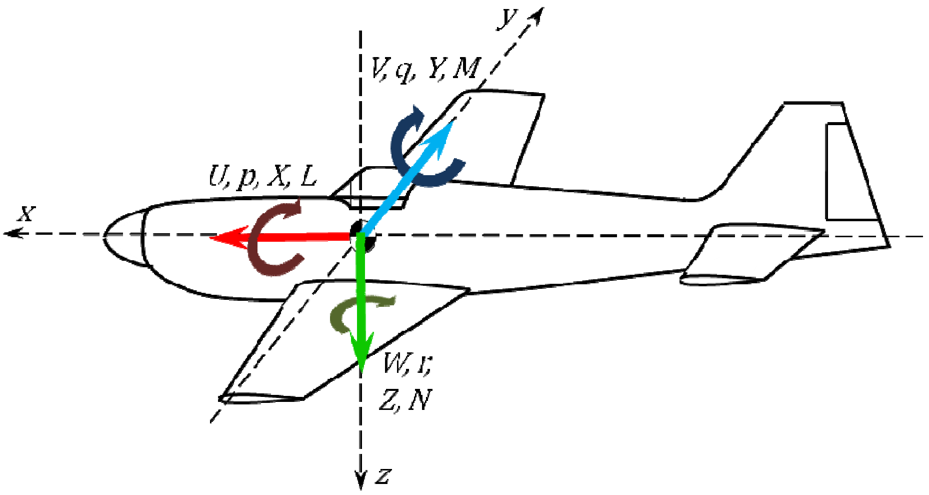


Figure 2.5: Diagram indication the positive sign convention used in the flight dynamic model

In the case of an aircraft, the forces X, Y and Z, and moments L, M and N can be broken down into Eqs. (2.9) to (2.14), as indicated in [37]; again the sign convention shown in Figure 2.5 applies.

$$X = X_{Aero} + X_{Thrust} + X_{Gravity} + X_{Disturbances} \tag{2.9}$$

$$Y = Y_{Aero} + Y_{Thrust} + Y_{Gravity} + Y_{Disturbances} \tag{2.10}$$

$$Z = Z_{Aero} + Z_{Thrust} + Z_{Gravity} + Z_{Disturbances} \tag{2.11}$$

$$L = L_{Aero} + L_{Thrust} + L_{Gravity} + L_{Disturbances} \tag{2.12}$$

$$M = M_{Aero} + M_{Thrust} + M_{Gravity} + M_{Disturbances} \tag{2.13}$$

$$N = N_{Aero} + N_{Thrust} + N_{Gravity} + N_{Disturbances} \tag{2.14}$$

Typically in aircraft control analyses, an assumption is made that the aircraft is symmetrical about the x-z plane, which aids in reducing the coupling between these equations. However, to allow for the possibility of asymmetry of an aircraft due to a fault or failure, this assumption has not been made here, and the further development will continue without this simplification. From these equations the accelerations of the aircraft in all directions can be determined by rearranging Eqs. (2.3) - (2.8) into the following form and then solving the differential equations numerically.

$$\dot{\boldsymbol{x}} = f(U, V, W, p, q, r)$$

$\dot{\boldsymbol{x}}$ in the above equation represents the time rate of change of the state vector of the aircraft and is given by:

$$\dot{\boldsymbol{x}} = [\dot{U}\ \dot{V}\ \dot{W}\ \dot{p}\ \dot{q}\ \dot{r}]^T$$

Equations (2.3), (2.4) and (2.5) can be easily transformed into this form by inspection and are thus presented as Eqs. (2.15) - (2.17) respectively.

$$\dot{U} = \frac{X}{m} + rV - qW \tag{2.15}$$

$$\dot{V} = \frac{Y}{m} + pW - rU \tag{2.16}$$

$$\dot{W} = \frac{Z}{m} + qU - pV \tag{2.17}$$

However, Eqs. (2.6) - (2.8) require a fair degree of manipulation to achieve this form. The development of these equations is presented in Appendix A, with the results of the manipulation presented as Eqs. (2.18) - (2.20):

$$\dot{p} = \frac{\left(I_yI_z - I_{yz}^2\right)\left\{I_z\left[L + \left(I_y - I_z\right)qr - I_{xy}pr + I_{xz}pq - I_{yz}(r^2 - q^2) + I_{xz}A\right] + B\left(I_{xy}I_z + I_{xz}I_{yz}\right)\right\}}{\left(I_xI_z - I_{xz}^2\right)\left(I_y - I_{yz}^2\right) - \left(I_{xy}I_z + I_{xz}I_{yz}\right)\left(I_{xz}I_{yz} + I_{xy}I_z\right)} \tag{2.18}$$

$$\dot{q} = B + \left(\frac{I_{xz}I_{yz} + I_{xy}I_z}{I_yI_z - I_{yz}^2}\right)\dot{p} \tag{2.19}$$

$$\dot{r} = A + \frac{I_{yz}\dot{q}}{I_z} + \frac{I_{xz}\dot{p}}{I_z} \tag{2.20}$$

where A and B are given by Eqs. (2.21) and (2.22) respectively:

$$A = \frac{N + \left(I_x - I_y\right)pq + I_{yz}(pr) - I_{xz}(qr) - I_{xy}(q^2 - p^2)}{I_z} \tag{2.21}$$

$$B = \frac{I_z\left(M - (I_x - I_z)pr - I_{yz}(pq - A) - I_{xz}(p^2 - r^2) + I_{xy}qr\right)}{I_yI_z - I_{yz}^2} \tag{2.22}$$

The equations of motion presented describe the acceleration of the aircraft in all directions and rotations about the body axes. From these equations, the state of the aircraft can be estimated by solving the equations simultaneously, over a very small time interval. A fourth-order Runge-Kutta method was used to achieve this. It is used to solve first order differential equations of the form [38] :

$$\dot{x} = f(x, t)$$

where the initial conditions are known. The method is a multistep method, i.e. the function is evaluated multiple times for a single time step. While this is computationally more intense, the accuracy of the solution is improved over single step methods, such as the Euler method, for the same time step size [38]. Assuming that a process follows the form given above with initial conditions X_n at t_n, the Runge-Kutta method then estimates the state of X at the next interval t_{n+1}, using Eq. (2.27), from [38], after the Runge Kutta estimators, $k_{1\rightarrow4}$, are determined using Eqs. (2.23) to (2.26):

$$k_1 = hf(t_n, X_n) \tag{2.23}$$

$$k_2 = hf\left(t_n + \frac{1}{2}h, X_n + \frac{1}{2}k_1\right) \tag{2.24}$$

$$k_3 = hf\left(t_n + \frac{1}{2}h, X_n + \frac{1}{2}k_2\right) \tag{2.25}$$

$$k_4 = hf(t_n + h, X_n + k_3) \tag{2.26}$$

$$X_{n+1} = X_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{2.27}$$

This process is continued until the desired time has been reached. This process can be extended to a set of first-order simultaneous equations by calculating a set of estimators $k_{1\rightarrow4}$ for each set of equations in sequence [38]. This is done for this flight dynamic model where a set of 9 simultaneous equations are solved. $U, V, W, p, q, r, \emptyset, \theta$ and Altitude are solved simultaneously using the fourth order Runge Kutta method. Heading and geodetic position are simply determined using numerical integration techniques.

2.6 Atmospheric Modelling

The atmospheric model used in this fight dynamic simulation is the International Standard Atmosphere (ISA) model. Turbulence has not yet been modelled; however, allowance has been made for constant translational winds. The primary requirement for this model is to provide the air density variation with altitude for the simulation. For this simulation, the atmosphere will only be modelled to the top of the Troposphere. The properties of the atmosphere used in this simulation are given in Table 2.2.

Table 2.2: Atmospheric Properties used in the flight dynamic model

Description	Symbol	Value	Units
ISA Temperature at S.L	T_0	288.16	K
Environmental Lapse Rate	L	-0.0065	K/m
ISA Pressure at Sea Level	P_0	101325	Pa
Universal Gas Constant	R	8.31447	J/mol
Molar Mass	M	0.0289644	Kg/mol
Air Density at Sea Level	ρ_0	1.225	Kg/m^3

The temperature at any altitude based on the ISA can be found using Eq. (2.28):

$$T = T_0 + hL \tag{2.28}$$

where h is the height above sea level in m.

This is then used to determine the pressure of air at that temperature using Eq. (2.29):

$$P = P_0 \left(1 + \frac{Lh}{T_0}\right)^{\frac{gM}{RL}} \tag{2.29}$$

And finally, the air density ratio σ can be found using Eq. (2.30):

$$\sigma = \frac{PM}{RT\rho_0} \tag{2.30}$$

The air density ratio is used to calculate the dynamic pressure of the air and hence affects all aerodynamic force calculations.

2.7 Force and Moment Modelling

2.7.1 Gravity

Gravitational forces are relatively easy to understand and model. For the case of a UAS, which primarily operates near the earth's surface, one can make the assumption that the gravity force vector acts towards the centre of the Earth, or in a positive sense in the earth fixed z-axis. This force is given by the well known Eq.(2.31):

$$F_G = mg \tag{2.31}$$

where m is the mass of the aircraft and g is the gravitational acceleration due to gravity with a value of 9.81m/s^2.The equations of motion are derived in the body-fixed axis coordinate system, which is free to rotate and translate relative to the Earth-fixed axis system. Thus all that remains is to transform the gravitational force vector from the Earth-fixed axis to the body-fixed axis. This is accomplished using Eqs. (2.32) - (2.37):

$$X_b = F_G \sin(\theta) \tag{2.32}$$

$$Y_b = F_G \sin(\emptyset)\cos(\theta) \tag{2.33}$$

$$Z_b = F_G \cos(\theta)\cos(\emptyset) \tag{2.34}$$

$$L_b = 0 \tag{2.35}$$

$$M_b = 0 \tag{2.36}$$

$$N_b = 0 \tag{2.37}$$

2.7.2 Aerodynamic

The aerodynamic modelling for the flight dynamic simulation will be done through a coefficient build up method, based on experimental wind tunnel data gathered at the CSIR's 7m wind tunnel. In order to use this data, the translational and rotational velocities determined through the equations of motion need to be converted into the wind axis system. Thus the velocities U, V and W are converted into $V_{wa},\ \alpha$ and β using Eqs. (2.38) to (2.40), these can be differentiated to get $\dot{\alpha}\ and\ \dot{\beta}$.

$$V_{wa} = \sqrt{U^2 + V^2 + W^2} \tag{2.38}$$

$$\alpha = \arctan\left(\frac{W}{V_{wa}}\right) \tag{2.39}$$

$$\beta = \arctan\left(\frac{V}{V_{wa}}\right) \tag{2.40}$$

Using these equations, as well as the rate of change in angle of attack ($\dot{\alpha}$), roll rate, pitch rate and yaw rate (p, q and r) and the deflections of the control surfaces(δ), the force and moment coefficients are built up using (2.41) to (2.46):

$$\begin{aligned} C_L = \mathrm{C}_{\mathrm{L}_{0(\alpha)(\beta)}} + \Delta\mathrm{C}_{L_{\dot{\alpha}(\alpha\dot{)}(\beta)}}\dot{\alpha} + \Delta\mathrm{C}_{L_{p(\alpha)(\beta)}}p + \Delta\mathrm{C}_{L_{q(\alpha)(\beta)}}q + \Delta\mathrm{C}_{L_{r(\alpha)(\beta)}}r \\ + \Delta\mathrm{C}_{L_{Control(\alpha)(\beta)(\delta_{Control})}}\delta \end{aligned} \tag{2.41}$$

$$\begin{aligned} C_D = \mathrm{C}_{D_{0(\alpha)(\beta)}} + \Delta\mathrm{C}_{D_{\dot{\alpha}(\alpha\dot{)}(\beta)}}\dot{\alpha} + \Delta\mathrm{C}_{D_{p(\alpha)(\beta)}}p + \Delta\mathrm{C}_{D_{q(\alpha)(\beta)}}q + \Delta\mathrm{C}_{D_{r(\alpha)(\beta)}}r \\ + \Delta\mathrm{C}_{D_{Control(\alpha)(\beta)(\delta_{Control})}}\delta \end{aligned} \tag{2.42}$$

$$\begin{aligned} C_M = \mathrm{C}_{M_{0(\alpha)(\beta)}} + \Delta\mathrm{C}_{M_{\dot{\alpha}(\alpha\dot{)}(\beta)}}\dot{\alpha} + \Delta\mathrm{C}_{M_{p(\alpha)(\beta)}}p + \Delta\mathrm{C}_{M_{q(\alpha)(\beta)}}q + \Delta\mathrm{C}_{M_{r(\alpha)(\beta)}}r \\ + \Delta\mathrm{C}_{M_{Control(\alpha)(\beta)(\delta_{Control})}}\delta \end{aligned} \tag{2.43}$$

$$\begin{aligned} C_l = \Delta\mathrm{C}_{l_{0(\beta)(\alpha)}} + \Delta\mathrm{C}_{l_{\dot{\alpha}(\beta\dot{)}(\alpha)}}\dot{\alpha} + \Delta\mathrm{C}_{l_{p(\beta)(\alpha)}}p + \Delta\mathrm{C}_{l_{q(\beta)(\alpha)}}q + \Delta\mathrm{C}_{l_{r(\beta)(\alpha)}}r \\ + \Delta\mathrm{C}_{l_{Control(\beta)(\alpha)(\delta_{Control})}}\delta \end{aligned} \tag{2.44}$$

$$\begin{aligned} C_Y = \mathrm{C}_{Y_{0(\beta)(\alpha)}} + \Delta\mathrm{C}_{Y_{\dot{\alpha}(\beta\dot{)}(\alpha)}}\dot{\alpha} + \Delta\mathrm{C}_{Y_{p(\beta)(\alpha)}}p + \Delta\mathrm{C}_{Y_{q(\beta)(\alpha)}}q + \Delta\mathrm{C}_{Y_{r(\beta)(\alpha)}}r \\ + \Delta\mathrm{C}_{Y_{Control(\beta)(\alpha)(\delta_{Control})}}\delta \end{aligned} \tag{2.45}$$

$$\begin{aligned} C_N = \mathrm{C}_{N_{0(\beta)(\alpha)}} + \Delta\mathrm{C}_{N_{\dot{\alpha}(\beta\dot{)}(\alpha)}}\dot{\alpha} + \Delta\mathrm{C}_{N_{p(\beta)(\alpha)}}p + \Delta\mathrm{C}_{N_{q(\beta)(\alpha)}}q + \Delta\mathrm{C}_{N_{r(\beta)(\alpha)}}r \\ + \Delta\mathrm{C}_{N_{Control(\beta)(\alpha)(\delta_{Control})}}\delta \end{aligned} \tag{2.46}$$

where the first subscript refers to the parameter causing the change in coefficient and the subscripts in brackets refer to value of the change in coefficient at a particular angle of attack and sideslip angle and control deflection.

The coefficients are determined from wind tunnel data. The wind tunnel data for each aerodynamic coefficient has been characterised by a set of polynomials which are selected based on the angle of attack and sideslip angle for that aircraft state and interpolated. The details of the wind tunnel data and interpolation process is described in more detail in Chapter 2.8.5. The forces in the wind axis system can thus be found using Eq. (2.47) to (2.52).

$$L = \frac{1}{2}\rho_0\sigma V_{wa}^2 S_{Ref} C_L \tag{2.47}$$

$$D = \frac{1}{2}\rho_0\sigma V_{wa}^2 S_{Ref} C_D \tag{2.48}$$

$$M = \frac{1}{2}\rho_0\sigma V_{wa}^2 S_{Ref} \bar{c} C_M \tag{2.49}$$

$$Y = \frac{1}{2}\rho_0\sigma V_{wa}^2 S_{Ref} C_Y \tag{2.50}$$

$$l = \frac{1}{2}\rho_0 \sigma V_{wa}^2 S_{Ref} b C_l \tag{2.51}$$

$$N = \frac{1}{2}\rho_0 \sigma V_{wa}^2 S_{Ref} b C_N \tag{2.52}$$

where, ρ_0 is the air density at sea level, σ is the air density ratio, which varies with altitude, as already defined, V_{wa} is the velocity of the UAS in the wind axis and S_{Ref} is the wing area of the UAS. These forces are then transformed into the body-axis system using the wind axis to body axis transformation described in Chapter 2.4.

2.7.3 Power Plant Modelling

Propeller Modelling

The characterisation of the propeller is achieved through the use of a non-dimensional parameter which relates the velocity of the propeller through the air with its rotational velocity. This parameter, known as the advance ratio, allows a propeller to be characterised independently of forward velocity and is given by [39]:

$$J = \frac{V}{nD} \tag{2.53}$$

Typically, propeller data is in the form of a set of coefficients that vary with advance ratio. These coefficients, the thrust coefficient, torque coefficient and power coefficient, allow the determination of the thrust and torque generated or absorbed by the propeller.

The thrust coefficient is used to determine the thrust generated, T, by the propeller and is given by:

$$T = \rho \sigma n^2 D^4 C_T \tag{2.54}$$

The torque, Q, can be determined using (2.55).

$$Q = \frac{\rho \sigma n^2 D^5 C_P}{2\pi} \tag{2.55}$$

where C_T and C_P are the thrust and power coefficients respectively, D is the diameter of the propeller, and n is the rotational speed of the propeller.

To model the thrust and torque, all that remains is to obtain an expression to describe C_T and C_P variation with the advance ratio of the propeller. This can be achieved in a number of ways that include wind tunnel testing and analytical analysis. For the propellers used on the CSIR's Modular

UAS, no propeller data was available and thus a blade element analysis was performed. This blade element analysis model, used for small propellers can be found in Appendix B.

The model described in Appendix B was run for a number of model propellers for which data was available in order to validate the model and ensure an accurate result. The results of the validation showed that, while the thrust coefficient agreed closely with wind tunnel data, the power coefficient was approximately 35% lower than measured in the wind tunnel. This may be due, in part, to the errors associated with wind tunnel testing and errors and simplifications made in the blade element code or the aerofoil data used in the analysis. It was decided to use the blade element code, as is, as the thrust model dominates in the flight dynamics of the aircraft and the error associated with the different power coefficient will only affect the motor dynamics, and current drawn. In the context of aircraft motion, these differences are considered small.

The blade element analysis was conducted for the propellers used on the Modular UAV, namely the Xoar 23x10 propeller. The physical geometry of the propeller was determined by Mr N Moore at the CSIR and the blade angle was estimated by assuming that the propeller had an ideal pitch distribution. The results of the blade element analysis on the Xoar 23x10 propeller and the propeller used in the flight dynamic model are shown in Figure 2.6.

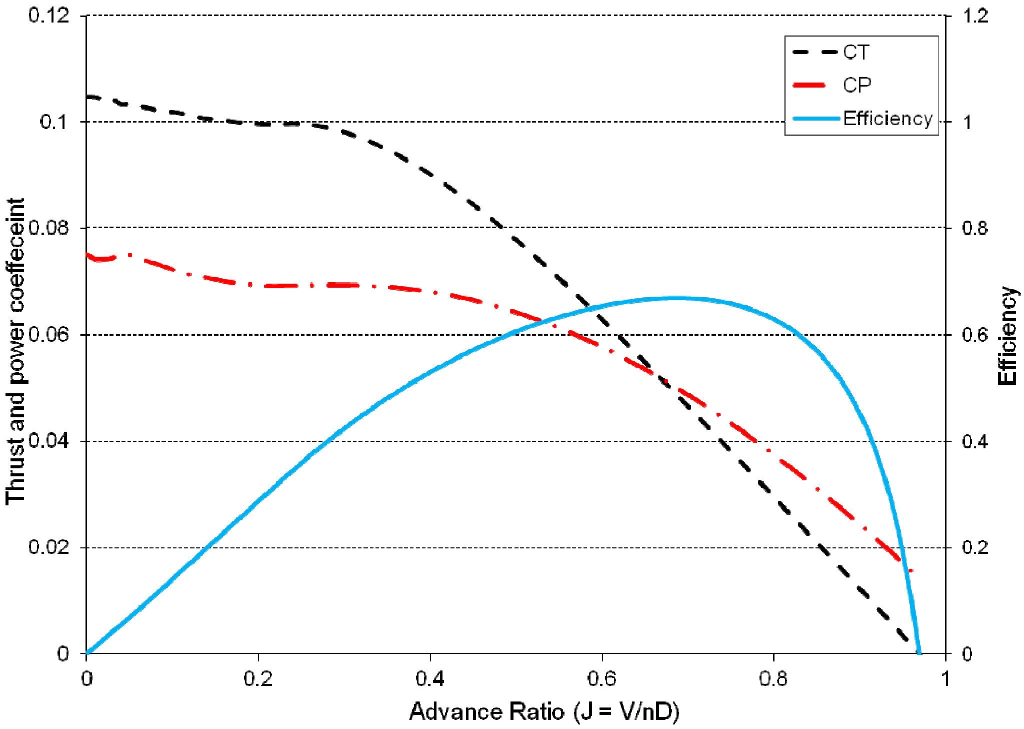


Figure 2.6: Estimated performance characteristics of the Xoar 23x10 propeller

The flight dynamic model made use of a polynomial model of these curves to determine the thrust and power coefficients. The coefficients of the polynomials used can be found in Table 2.3.

Table 2.3: Table indicating the polynomial coefficients used to describe the thrust and power coefficient curves of Figure 2.6

Polynomial coefficient	**Thrust Coefficient**	**Power Coefficient**
a0	0.1058065026	0.0762913734
a1	-0.0885243513	-0.0686730689
a2	0.6233706854	0.2752244756
a3	-1.9470978824	-0.4741954619
a4	2.0325856787	0.2018584708
a5	-0.7329001868	0

Motor Modelling

The motor model used for this study is the well defined “Three Constant” model, shown in Figure 2.7, which makes use of three main parameters to describe a brushless electric motor. These are the motor constant (kv), the no load current (I_o) and the internal resistance of the motor windings (R_m).

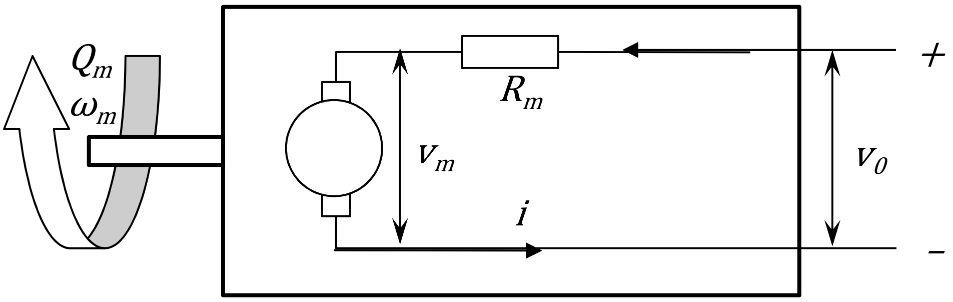


Figure 2.7: Diagram showing the main parameters of the three constant model adapted from [40]

Using these parameters and drawing the simplified circuit, the following can be derived from [40].

The torque produced by the motor is given by:

$$Q_m = \frac{(i - i_0)}{kv} \tag{2.56}$$

and the rotational velocity is thus:

$$\omega_m = (v_0 - iR_m)kv \tag{2.57}$$

The output power is then:

$$P_{Shaft} = Q_m \omega_m \tag{2.58}$$

and the input power is:

$$P_{Input} = v_0 i \tag{2.59}$$

Through some manipulation, the torque generated by the motor for any RPM and input voltage can be expressed in a more useful format given as:

$$Q_m = \frac{\left(\frac{\left(v_0 - \frac{\omega_m}{kv}\right)}{R_m} - i_0\right)}{kv} \tag{2.60}$$

This equation allows the determination of motor torque for any given voltage input and thus is useful in the flight dynamic model.

The control of an electric motor is accomplished through the use of an electronic speed controller (ESC). The speed controller synchronises the switching of the motor windings and, through the use of PWM control, the effective voltage (v_0) seen by the motor. The autopilot will provide the aircraft model with a % throttle setting which will then be converted to a voltage based on an assumption that the voltage is proportional to the throttle setting.

Combined model

The purpose of the power plant modelling is to obtain the forces and moments in the body axis system for a given power setting. However, there are dynamic effects of the motor and propeller combination that must be accounted for. Thus, the motor and propeller were modelled together.

This was accomplished by setting the throttle to an arbitrary starting point and crudely estimating the RPM at that throttle setting using:

$$RPM = v_0 kv \tag{2.61}$$

The accuracy of this is unimportant, as all that is required is a starting point. With this starting point, the advance ratio of the propeller in this state is determined. This allows the determination of the thrust and power coefficients and hence the thrust produced and the torque absorbed by the propeller.

Again, using the starting RPM, the torque provided by the motor is determined. Because the RPM is unlikely to be at equilibrium, there will be a difference in torque, resulting in an angular acceleration of the motor propeller combination. This angular acceleration is determined in the main solver block using (2.62).

$$a_{Motor} = \frac{\Delta Q}{J_{Motor}} \tag{2.62}$$

where, ΔQ is the difference in torque produced by the motor and that absorbed by the propeller and J_{Motor} is the second moment of inertia of the motor and propeller combination.

Finally, all that remains is to resolve the thrust and torque into components along the body axis coordinate system. Thus the forces and moments are given by Eqs. (2.63) to (2.68).

$$F_x = T\cos k\cos k_s \tag{2.63}$$

$$F_y = T\cos k\sin k_s \tag{2.64}$$

$$F_z = T\sin k \tag{2.65}$$

$$M_x = -F_y h - F_z w + Q_{Motor}\cos k\cos k_s \tag{2.66}$$

$$M_y = F_x h - F_z l + Q_{Motor}\cos k\sin k_s \tag{2.67}$$

$$M_z = -F_x w + F_y l \tag{2.68}$$

where, k is defined as the thrust line angle in the X-Z plane, k_s is the thrust line angle in the X-Y plane, h is the distance of the thrust line from the centre of gravity along the Z-Axis, w is the thrust line offset from the fuselage centreline and l is the distance of the motor from the lateral axis. These terms and their positive directions are defined in Figure 2.8. It is noted that the propellers in this case rotate in the same direction with is clockwise when viewed from the cockpit.

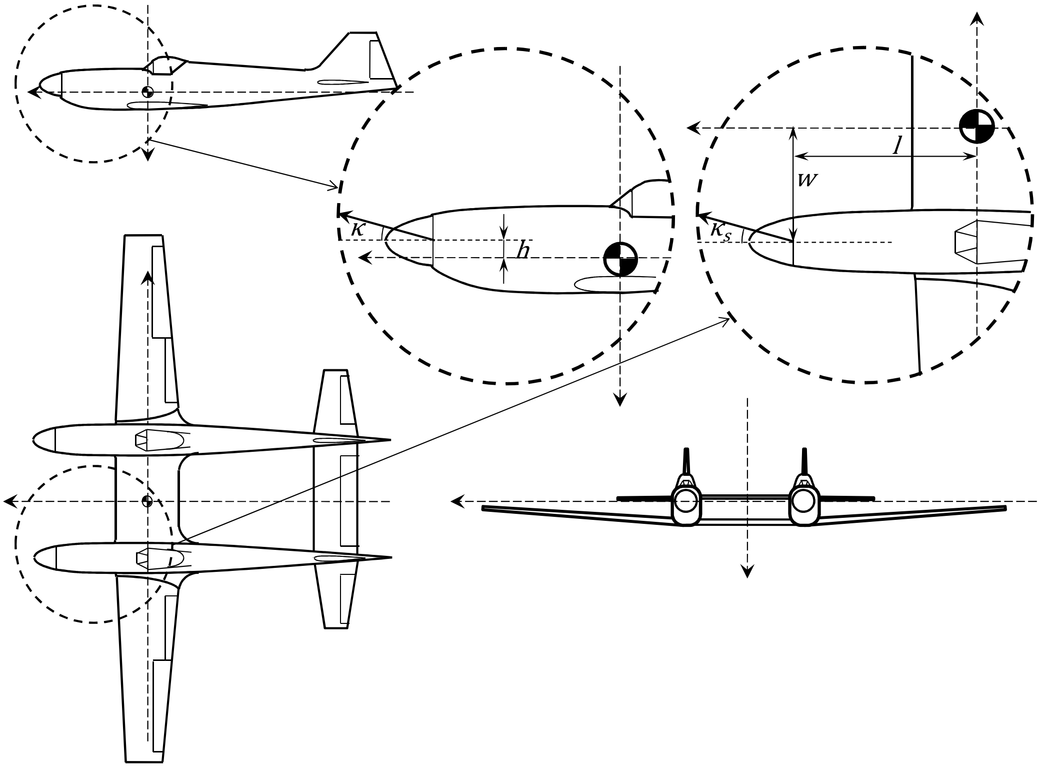


Figure 2.8: Definition of thrust lines on a fictional twin fuselage aircraft design similar to that of the CSIR's Modular UAV

2.8 Wind Tunnel Data

2.8.1 Introduction

One of the Modular UAVs manufactured was used in wind tunnel testing in order to characterise the airframe. This testing was done by Mr. Peter Skinner at the CSIR 7m Wind tunnel in 2009. The wind tunnel tests measured the lift, drag, pitching moment, side force, rolling and yawing moments of the aircraft for different angles of attack, angles of sideslip and control deflections. All of the data was collected statically and hence no dynamic data has been obtained. The data was also collect with no power added. Thus the dynamic stability derivatives will be determined using fundamental principles of stability analysis and the static wind tunnel data, where possible. That process is described in Section 2.8.6. The static wind tunnel data is presented in the sections that follow.

2.8.2 Data Processing

The wind tunnel data had to be processed in order to obtain the required parameters. This involved several separate steps to condition the data appropriately.

The data for 0° control deflection form the basis upon which all of the other data is referenced. The data set consisted of multiple sets of coefficient vs. angle of attack data for different sideslip angles. Thus, for the parameters that varied with angle of attack, the procedure was relatively simple as a simple regression could be performed. However, for the lateral coefficients which vary primarily with sideslip angle, an interpolation was required to obtain constant angles of attack across the data sets. Once this was completed, a regression could be performed with sideslip angle being the independent variable.

For the control surface deflection data, a similar process was conducted. A test was run for each control surface deflection increment. As an example, for a 4° aileron deflection, a data set of all the coefficients as a function of angle of attack was generated for each increment in sideslip angle. To compare the values and obtain a change in coefficient, an interpolation between angle of attack measurements was done so as to obtain consistent angle of attack values across the entire data set. After the interpolation, the data was subtracted from the zero reference condition and a change in coefficient generated. This was then collected and transposed to generate a data set with the control deflection as the independent variable. There was thus a data set for each coefficient, at 5 different angles of attack, for 6 sideslip angles.

2.8.3 Baseline Results

Lift Curve slope (C_{L_α})

The lift coefficient variation was as expected and found to be that typically present in aircraft as shown in Figure 2.9. The lift curve slope was found to be linear up to angles of approximately 10° at which point a reduction in lift curve slope was noted. A maximum lift coefficient of 1.46 was recorded for the condition of 0° sideslip angle.

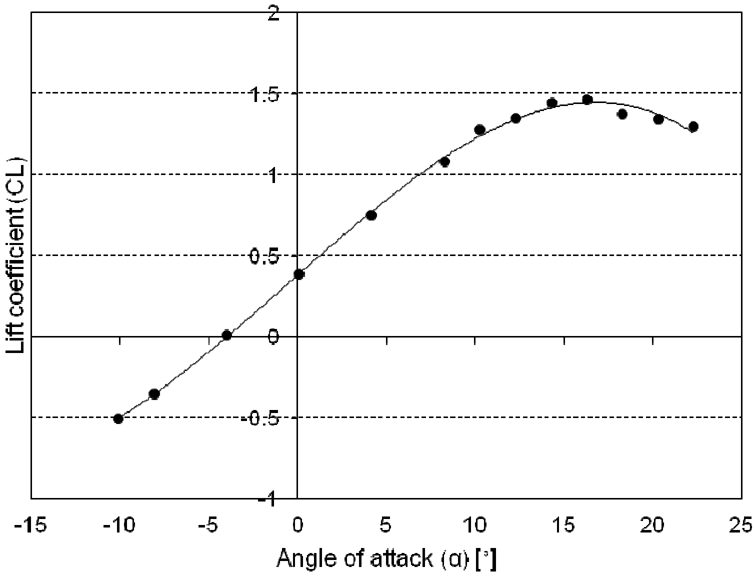
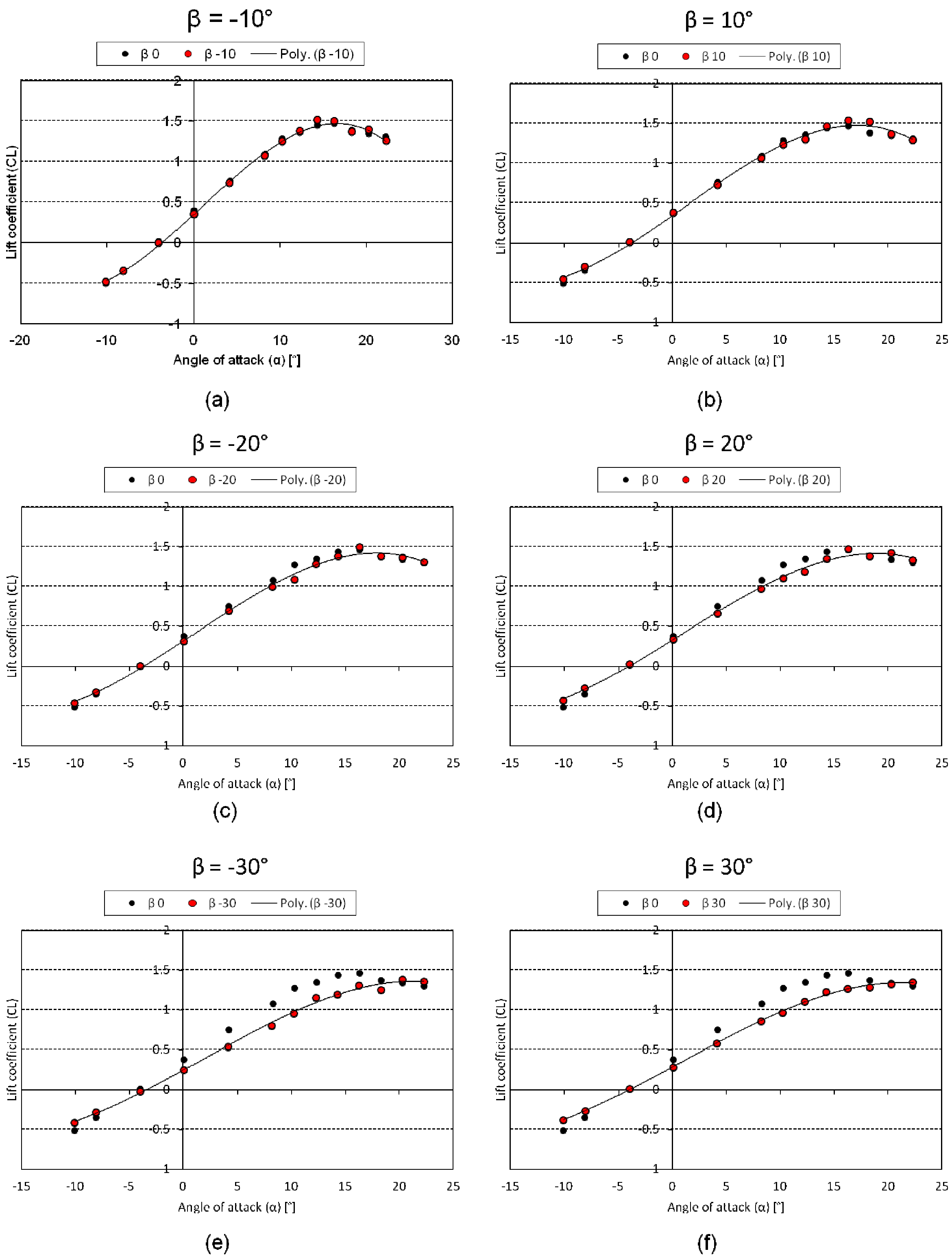


Figure 2.9: Lift coefficient variation with angle of attack for 0° sideslip angle (β)

Figure 2.10 (a) to (f) show the lift coefficient variation with angle of attack for different sideslip angles. The lift coefficient variation for 0° sideslip angle is included on these figures to highlight the variation of the lift coefficient with sideslip angle. As can be seen in the figures, the lift coefficient is typically lower for increased sideslip angle. A 3rd-order polynomial was found to fit the data satisfactorily and the results of this regression are presented in Table 2.4.

Table 2.4: Polynomial coefficients for the lift coefficient variation with angle of attack for different sideslip angles as presented in Figure 2.10 (a) to (f)

Coefficient	**β = -30°**	**β = -20°**	**β = -10°**	**β = 0°**	**β = 10°**	**β = 20°**	**β = 30°**
a0	0.24669	0.31829	0.34371	0.38014	0.34445	0.33062	0.286927
a1	0.07436	0.08944	0.09816	0.09663	0.09485	0.08405	0.073074
a2	0.00036	0.00031	0.00037	-0.00016	0.00048	0.00021	0.000127
a3	-6.518E-05	-0.00011	-0.00013	-0.00011	-0.00012	-0.00008	-5.763E-05



(a)

(b)

(c)

(d)

(e)

(f)

Figure 2.10: Lift coefficient variation with angle of attack for various sideslip angles (β)

Drag Coefficient Variation with Angle of Attack (C_{D_α})

The drag coefficient variation with angle of attack varied in a largely parabolic manner, as shown in Figure 2.11, with the presence of a slight "drag bucket" near the lower angles of attack. This was expected based on the characteristics of the aerofoil used on the Modular UAV.

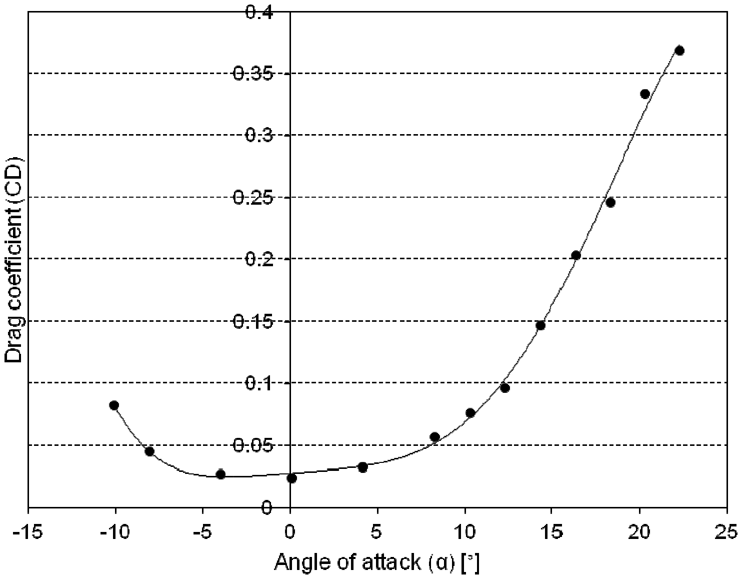
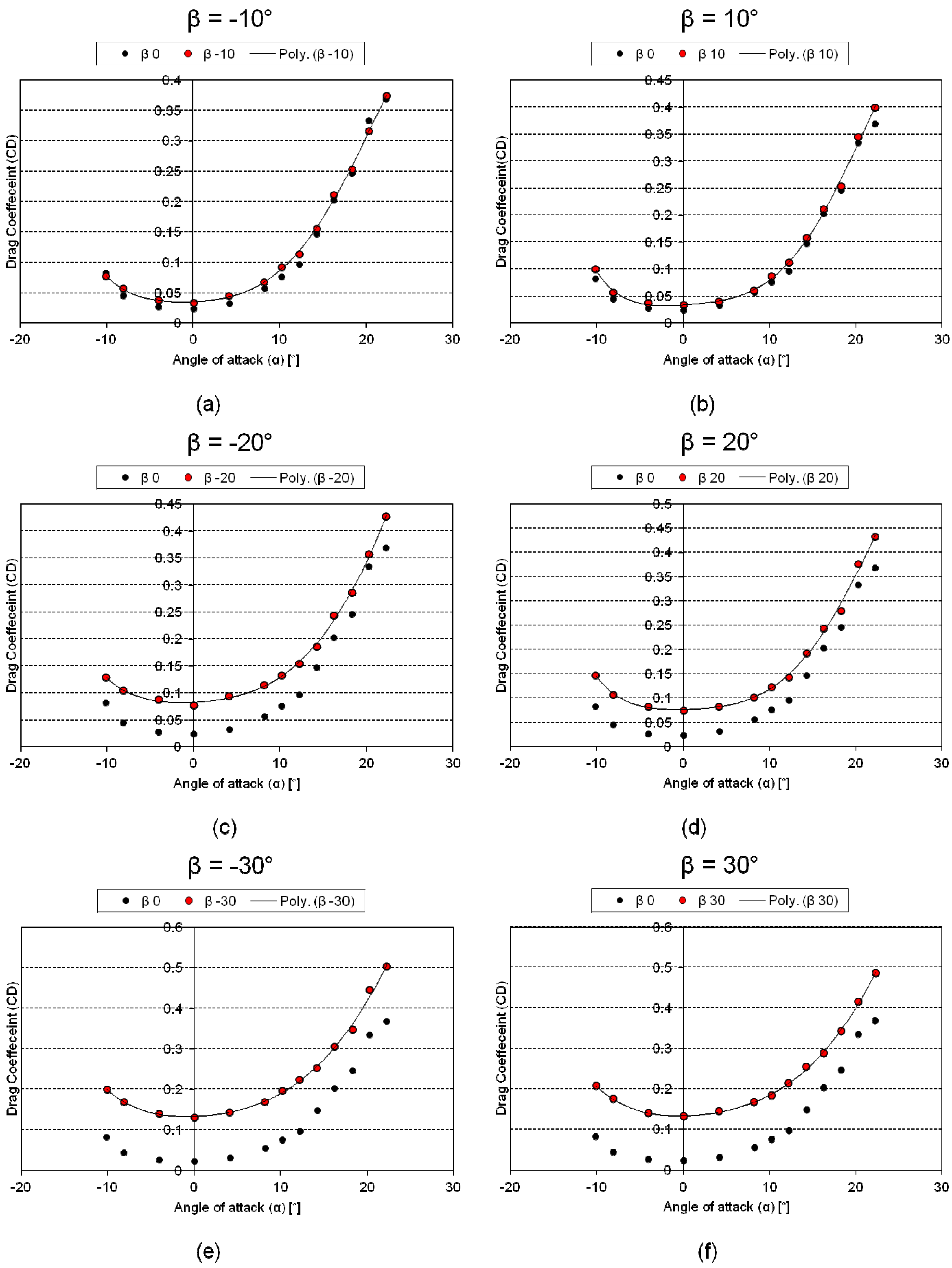


Figure 2.11: Drag coefficient variation with angle of attack for 0° sideslip angle (β)

The drag coefficient increased but the shape widened with increasing sideslip angle as shown in Figure 2.12 (a) to (f). A least squares linear regression was performed, the results of which are given in Table 2.5. As can be seen by the coefficients, the polynomial is somewhat over fitted and a 3rd order polynomial would be better in this regard; however, the 5th-order polynomial was required to capture the presence of the drag bucket.

Table 2.5: Polynomial coefficients of the drag variation with angle of attack for different sideslip angles

Coefficient	**β = -30°**	**β = -20°**	**β = -10°**	**β = 0°**	**β = 10°**	**β = 20°**	**β = 30°**
a0	0.13265	0.08259	0.03485	0.02733	0.03354	0.07664	0.13298
a1	0.000724	0.00076	0.00055	0.00109	0.00095	0.00028	0.00039
a2	0.000448	0.00027	0.00026	-3.306E-06	0.00016	0.00023	0.00045
a3	-9.024E-06	-3.42E-06	5.1819E-06	-1.788E-06	-7.175E-06	-7.669E-06	-1.234E-05
a4	1.5635E-06	1.7554E-06	1.9301E-06	4.744E-06	3.7827E-06	3.1323E-06	1.6155E-06
a5	-2.717E-08	-3.741E-08	-6.195E-08	-1.505E-07	-1.072E-07	-8.195E-08	-2.64E-08



(a)

(b)

(c)

(d)

(e)

(f)

Figure 2.12: Drag coefficient variation with angle of attack for various sideslip angles

Pitching Moment Coefficient Variation with Angle of Attack (C_{M_α})

The pitching moment variation with angle of attack was found to be largely linear at small angles of attack; however, a reduction in slope occurred at higher angles of attack as shown in Figure 2.13.

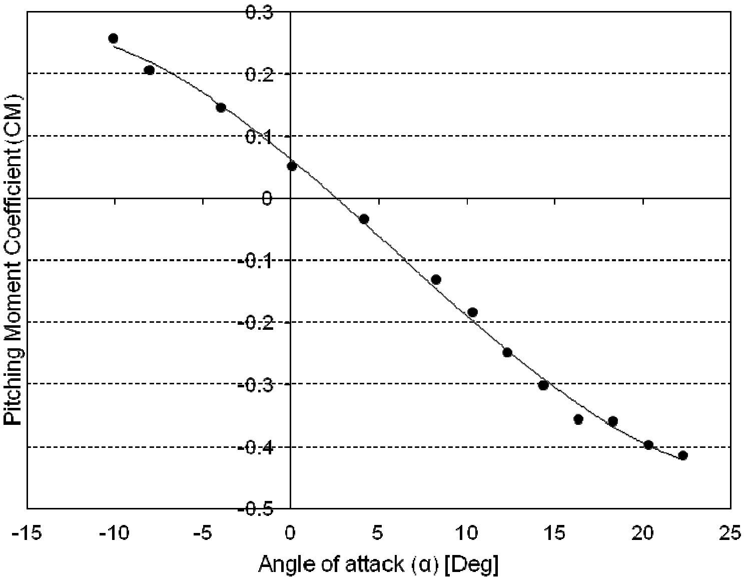
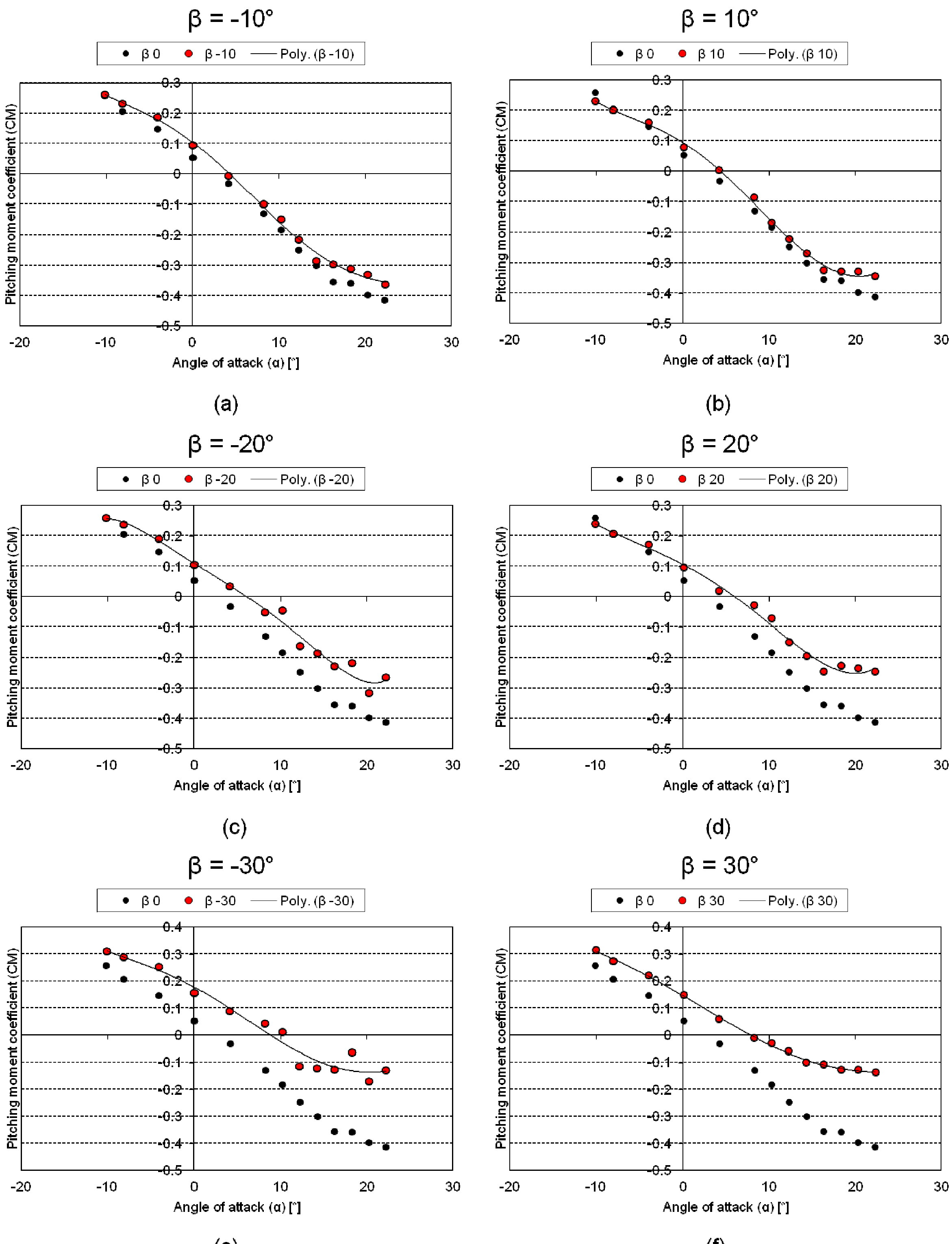


Figure 2.13: Pitching moment coefficient variation with angle of attack for 0° sideslip angle (β)

The pitching moment variation with angle of attack reduced as the sideslip angle increased and more scatter was noted at sideslip angles of 30°, as shown in Figure 2.14 (a) to (f). It was found that a 3rd-order polynomial fit adequately described the pitching moment variation and the results of the regression are shown in Table 2.6.

Table 2.6: Polynomial coefficients of the pitching moment variation with angle of attack for different sideslip angles

Coefficient	**β = -30°**	**β = -20°**	**β = -10°**	**β = 0°**	**β = 10°**	**β = 20°**	**β = 30°**
a0	0.17601	0.11974	0.10042	0.06370	0.09379	0.10991	0.14429
a1	-0.01911	-0.01910	-0.02344	-0.02363	-0.02244	-0.01876	-0.01872
a2	-0.00035	-0.00037	-0.00051	-0.00036	-0.00066	-0.00045	-0.00006
a3	0.00002	0.0002	0.00002	0.00002	0.00003	0.00002	0.00002



(a) (b)

(c) (d)

(e) (f)

Figure 2.14: Pitching moment coefficient variation with angle of attack for various sideslip angles

Side Force Coefficient Variation with Sideslip Angle (C_{Y_β})

The side force coefficient variation with sideslip angle was found to vary linearly at low angles. However, at high angles the slope of the line decreased slightly, as shown in Figure 2.15. There was very little variation for slight changes in angle of attack; however, at larger angles there was a rapid change in slope, with the side force variation with sideslip angle becoming flat near stall angle as shown in Figure 2.16 (a) to (f). It is speculated that this change in slope at high angles of attack may be due to a blanketing effect of the wings; however the direct cause is not known. This is a concern as at high angles of attack the negative slope suggests that the aircraft will be slightly unstable.

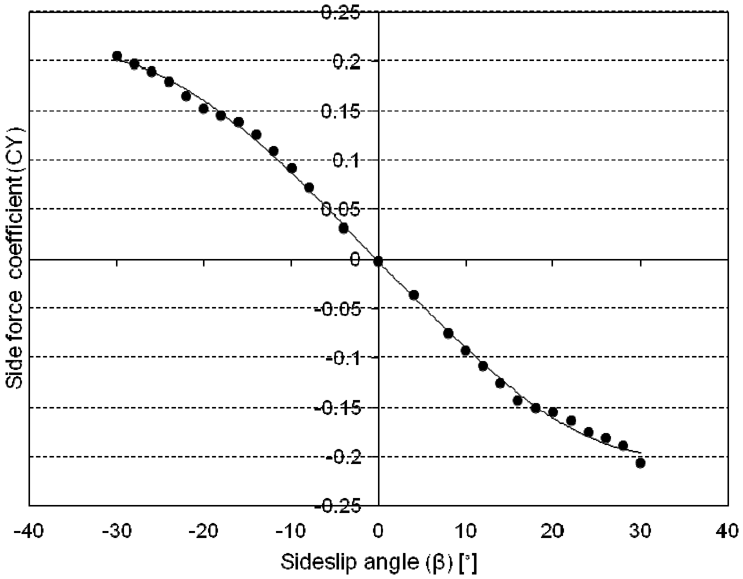
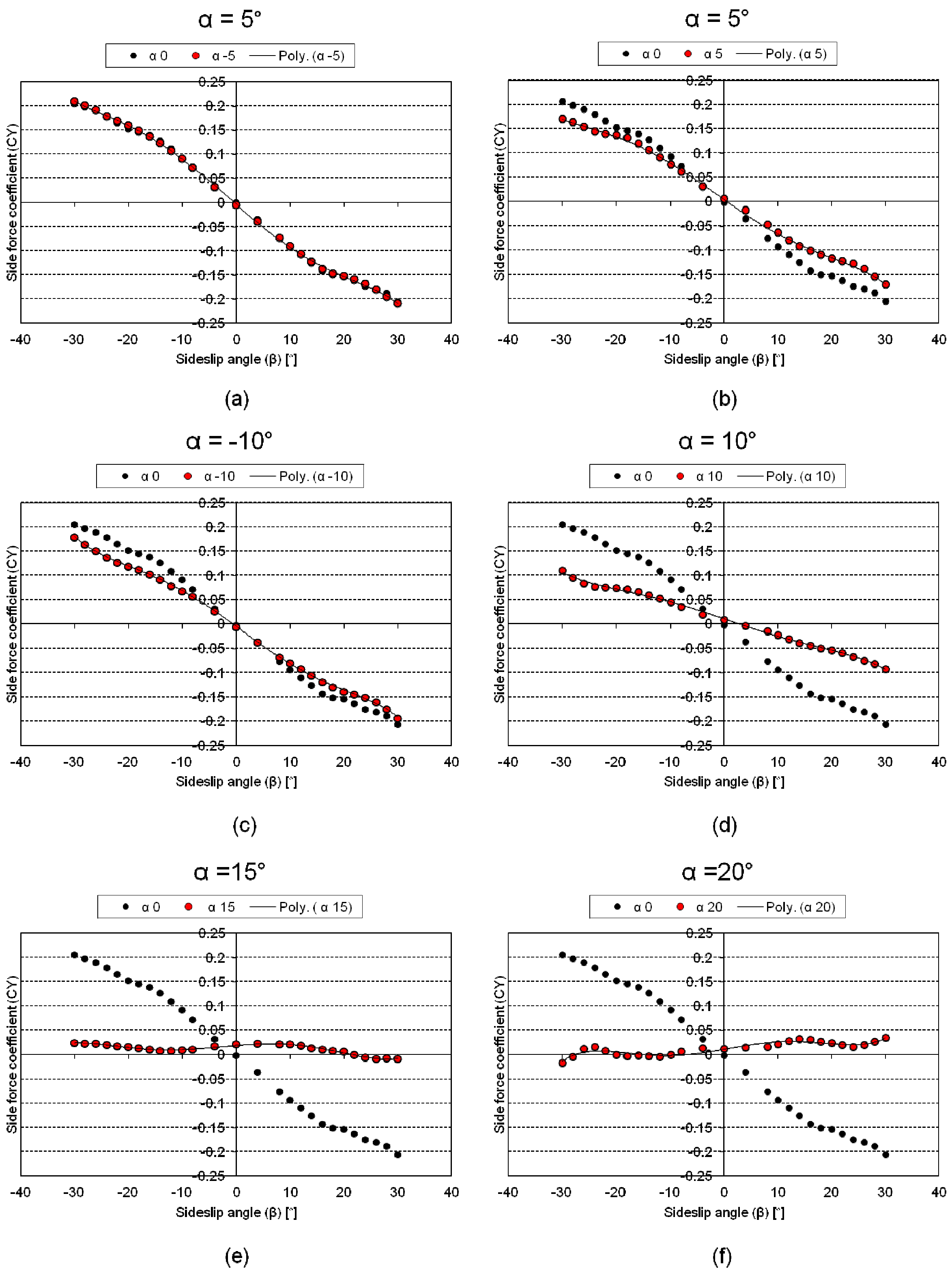


Figure 2.15: Side force coefficient variation with angle of attack for 0° sideslip angle (β)

A 3rd-order least squares regression was found to fit the data satisfactorily. The results of the regression and hence the coefficients used in the flight dynamic program are given in Table 2.7.

Table 2.7: Least squares regression results for side force coefficient variation with sideslip angle for different angles of attack

Coefficient	**α = -10°**	**α = -5°**	**α = 0°**	**α = 5°**	**α = 10°**	**α = 15°**	**α = 20°**
a0	-0.0071142	-0.0018153	-0.0019902	0.00734716	0.01042921	0.01541615	0.01200092
a1	-0.007085	-0.0089396	-0.0091446	-0.0070487	-0.0032439	0.00027276	0.00078588
a2	-6.959E-10	6.2803E-06	5.2861E-06	-3.225E-06	-3.97E-06	-1.245E-05	-9.902E-08
a3	1.3179E-06	2.4308E-06	2.817E-06	1.7791E-06	6.7076E-08	-1.084E-06	-3.796E-07



(a)

(b)

(c)

(d)

(e)

(f)

Figure 2.16: Side force coefficient variation with sideslip angle for different angles of attack

Rolling Moment Coefficient Variation with Sideslip Angle (C_{l_β})

The rolling moment coefficient followed a primarily linear trend with negative slope, indicating a stable tendency as shown in Figure 2.17. It was also noted that, at higher sideslip angles, a reduction in slope occurred. With an increase in magnitude of angle of attack, there was a steepening of the curve in the linear region as well as a more pronounced decrease in slope at high sideslip angles as seen in Figure 2.18 (a) to (f). It was noted that, at an angle of attack of 20°, there was a dip in rolling moment for large negative sideslip angles. The cause for this is unknown, but may be due to a blanketing effect of the fuselages.

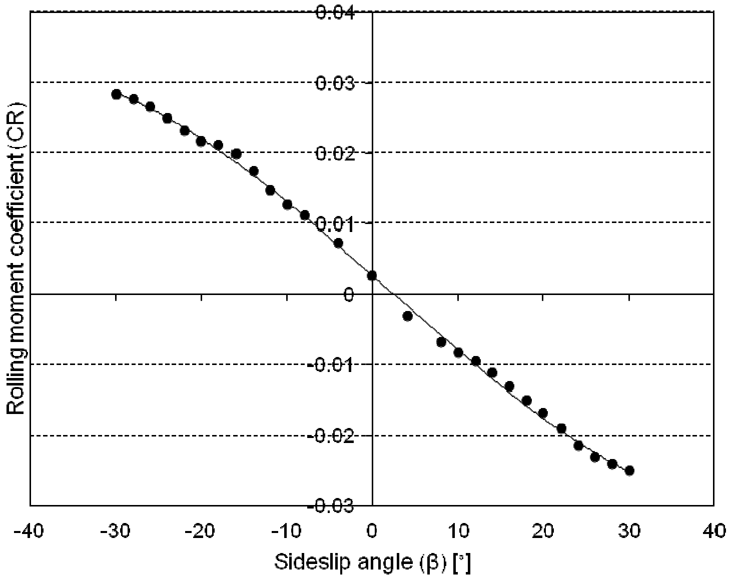


Figure 2.17: Rolling moment coefficient variation with sideslip angle at an angle of attack of 0°

A 3rd-order linear regression was found to fit the data well and was used to describe the rolling moment behaviour. The results of this regression and hence the coefficients used in the flight dynamic program are given in Table 2.8.

Table 2.8: Least squares regression results for rolling moment variation with sideslip angle for different angle of attack

Coefficient	**α = -10°**	**α = -5°**	**α = 0°**	**α = 5°**	**α = 10°**	**α = 15°**	**α = 20°**
a0	0.00528417	0.00407978	0.00240225	0.00192936	0.00185443	-0.0095291	-0.0083153
a1	-0.0006466	-0.001117	-0.0010764	-0.0014423	-0.0014066	-0.0019763	-0.002432
a2	-9.049E-06	-3.741E-06	1.616E-06	-1.999E-06	-1.742E-07	2.0831E-05	1.5522E-05
a3	1.3842E-07	4.6779E-07	2.354E-07	1.0894E-06	7.3168E-07	1.8134E-06	2.3682E-06

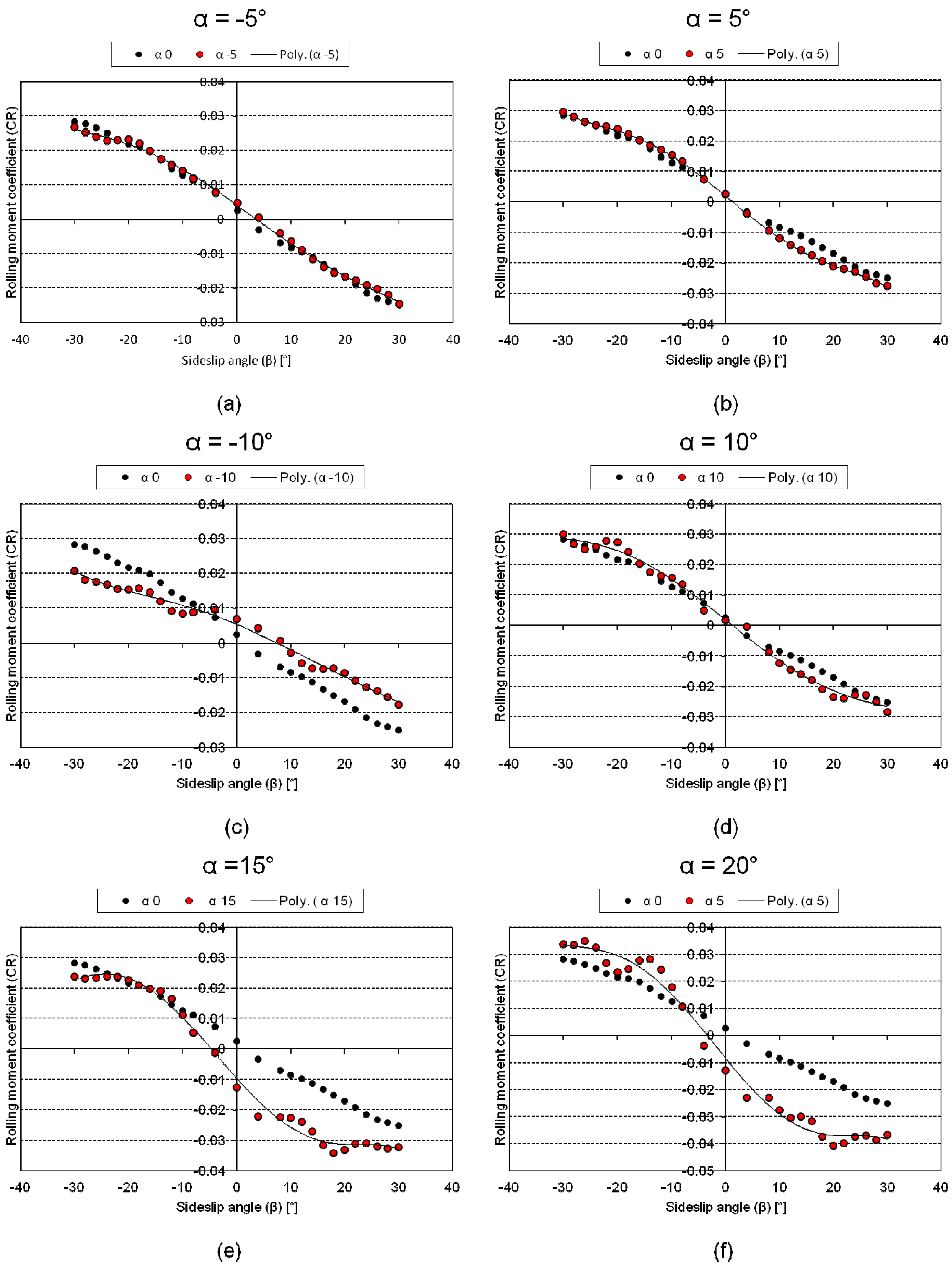


Figure 2.18: Rolling moment coefficient variation with sideslip angle for different angles of attack

Yawing Moment Coefficient Variation with Sideslip Angle (C_{N_β})

The yawing moment coefficient variation with sideslip angle, shown in Figure 2.19, was found to be linear for small angles of sideslip. As the magnitude of the sideslip angle passed approximately 15°, there was a reduction in yawing moment and the slope became negative. Thereafter, there was a recovery of yawing moment and the slope became positive. This trend was evident at low angles of attack but diminished at high angles of attach as seen in Figure 2.20 (a) to (f).

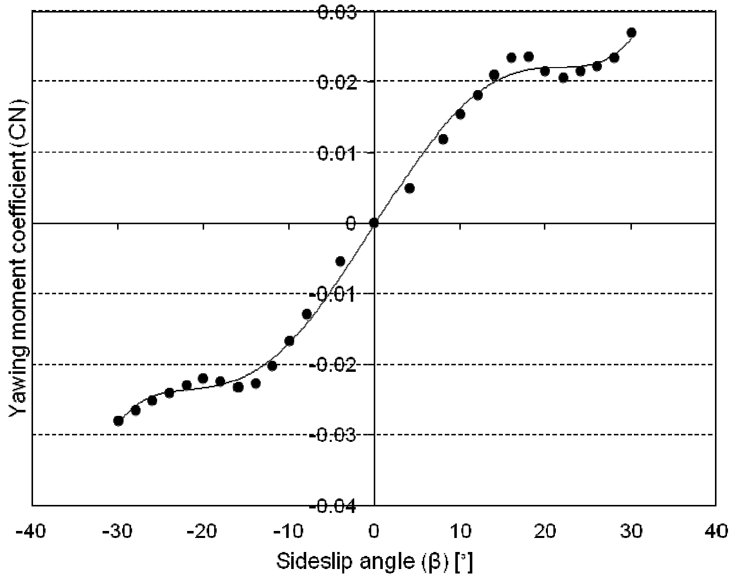


Figure 2.19: Yawing moment coefficient variation with sideslip angle at an angle of attack of 0°

To capture the dip in yawing moment at high sideslip angles, a 5th-order polynomial was chosen. The results of the regression are shown in Table 2.9.

Table 2.9: Least squares regression results for yawing moment variation with sideslip angle for different angle of attack

Coefficient	α = -10°	α = -5°	α = 0°	α = 5°	α = 10°	α = 15°	α = 20°
a0	-0.0017320	-0.0002301	-0.0002808	-0.0008298	-0.0014260	-0.0011462	-0.0035026
a1	0.00178371	0.00202250	0.00189811	0.00187141	0.00185015	0.00148768	0.00149542
a2	-5.126E-06	-4.633E-06	-9.679E-07	-2.931E-06	4.826E-07	1.768E-06	3.949E-06
a3	-2.483E-06	-2.72E-06	-2.57E-06	-2.335E-06	-1.699E-06	7.055E-07	1.743E-06
a4	3.871E-09	4.209E-09	-1.500E-11	3.029E-09	-1.229E-10	-2.553E-09	-5.651E-09
a5	1.579E-09	1.669E-09	1.636E-09	1.517E-09	1.038E-09	-8.433E-10	-1.556E-09

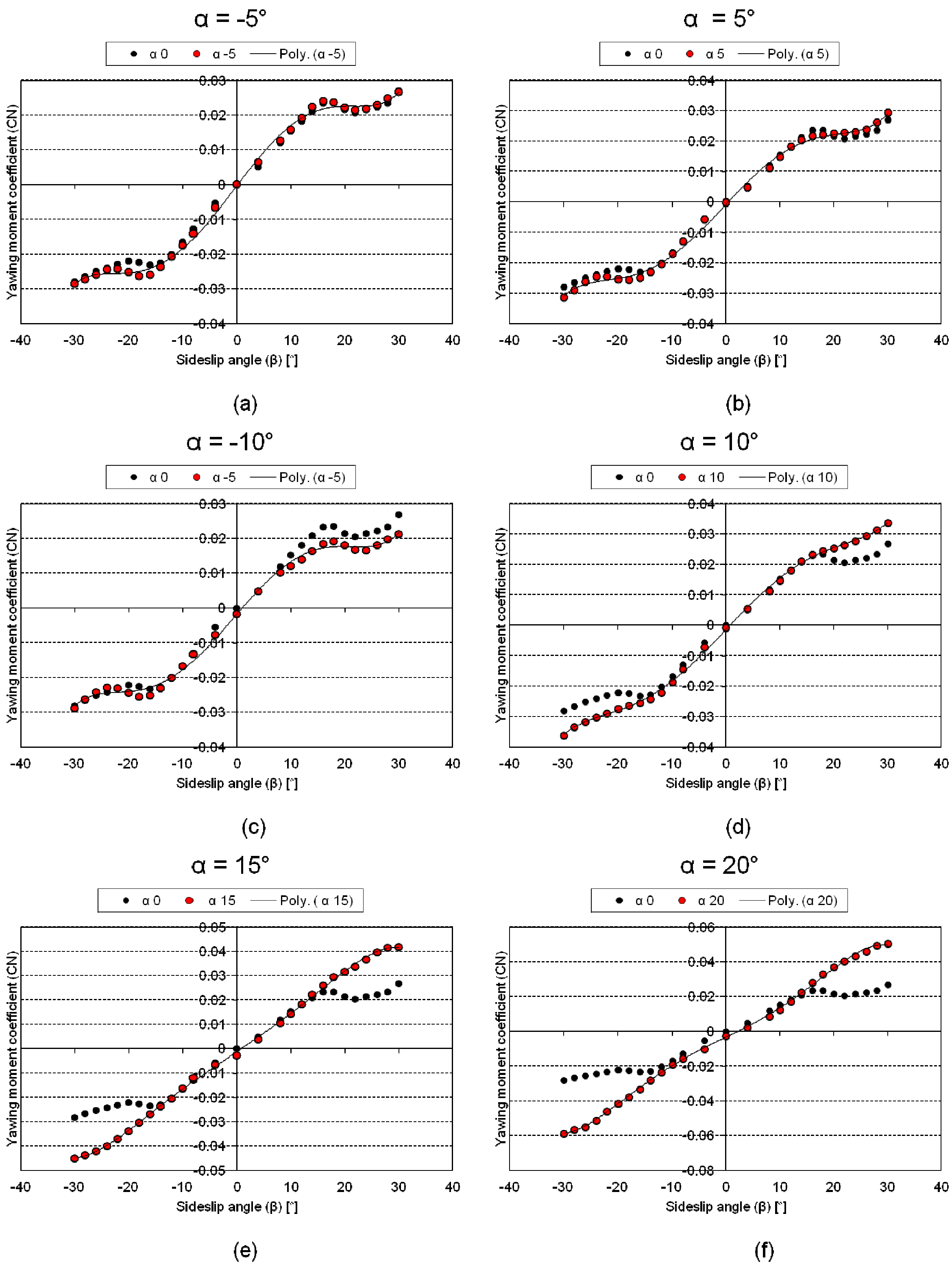


Figure 2.20: Yawing moment coefficient variation with sideslip angle for different angles of attack

2.8.4 Control Effects

The effects of the control surfaces were also tested statically at each angle of attack and sideslip angle in increments of 2°. From this data, the effect of each control surface could be determined by comparing each data set to that of the baseline wind tunnel results previously presented. A set of polynomials were created that best represented the relationship between the aerodynamic coefficient and the control deflection at each angle of attack and sideslip angle. This resulted in a number of 3D lookup tables containing the coefficient of a single term of the polynomial for a number of different sideslip angles and angles of attack. This meant that the coefficients of the polynomial would be found in a set of lookup tables and then the control surface deflection, being the independent variable, would be used to determine the extent of the change in all coefficients.

The number of relations and trends associated with all of the controls at all angles of attack and sideslip angles is considerable. For each control surface, using angle of attack increments of 5° and sideslip angle increments of 10°, 210 trend lines are generated. Thus, for three types of control surfaces, a total of 630 different curves, with their polynomial coefficients, are required to describe fully the effects of controls on the aircraft. Unfortunately, the majority of the control data only extended to a control deflection of ± 15°. This angle was well within the linear region, and as such the majority of polynomials are first-order. It is also noted that this lack of data will limit the maximum and minimum allowable deflections set in the flight dynamic model, as extrapolation of data was not conducted at any point in the flight dynamic code. All of these relations cannot be included in this dissertation for practical reasons; however, the full set of control data can be found in Appendix C. The primary effects, and in some cases the secondary effects, of controls are, however, presented in the figures that follow.

Elevator

The primary effect of elevator deflection is to cause a change in pitching moment of the aircraft. This trend was largely linear as demonstrated by Figure 2.21 (a) to (e), with large variations in pitching moment seen at large angles of attack and sideslip angles. The secondary effects, while present, were small and are thus not included here. However, they were implemented in the flight model, the details of which can be found in Appendix C.

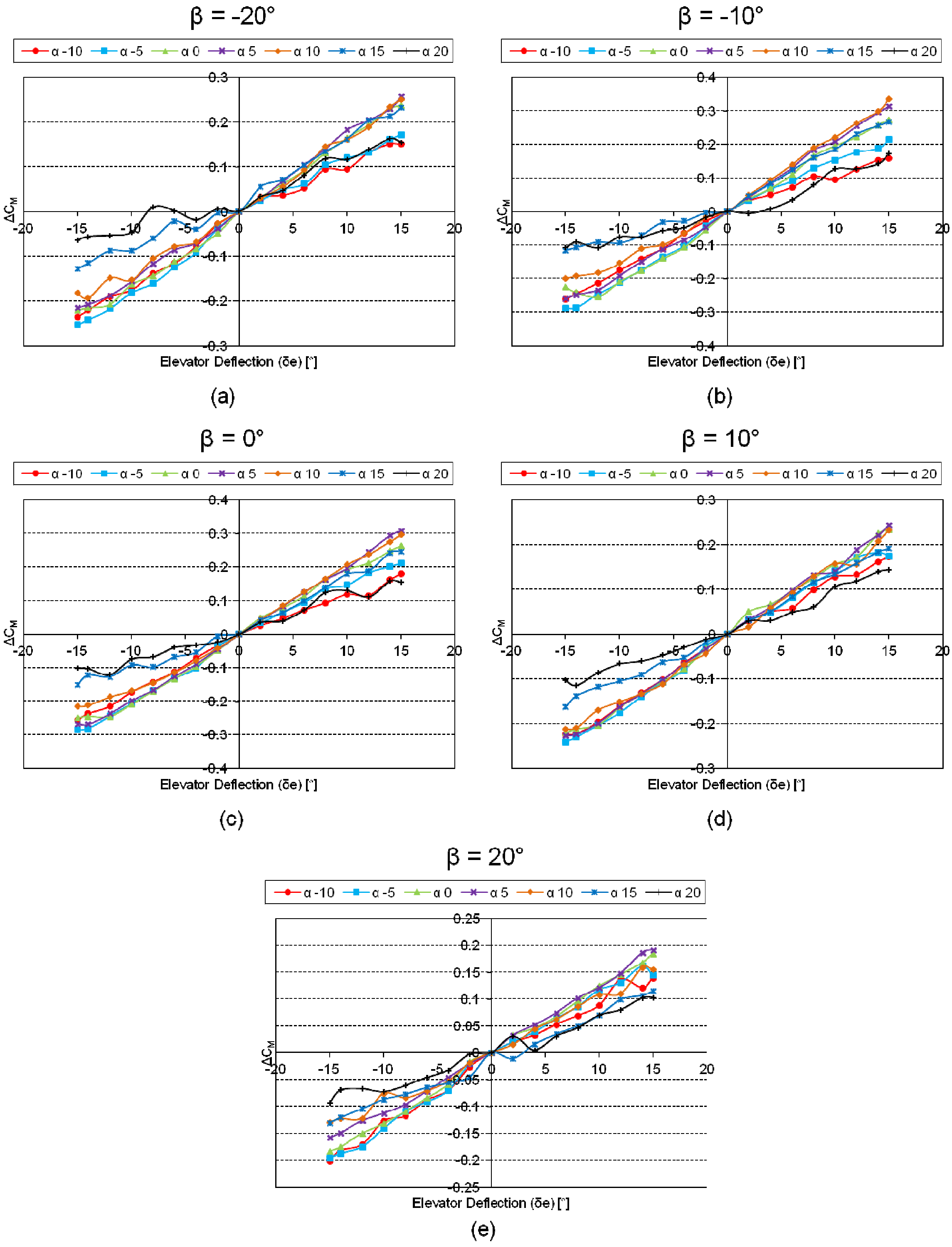
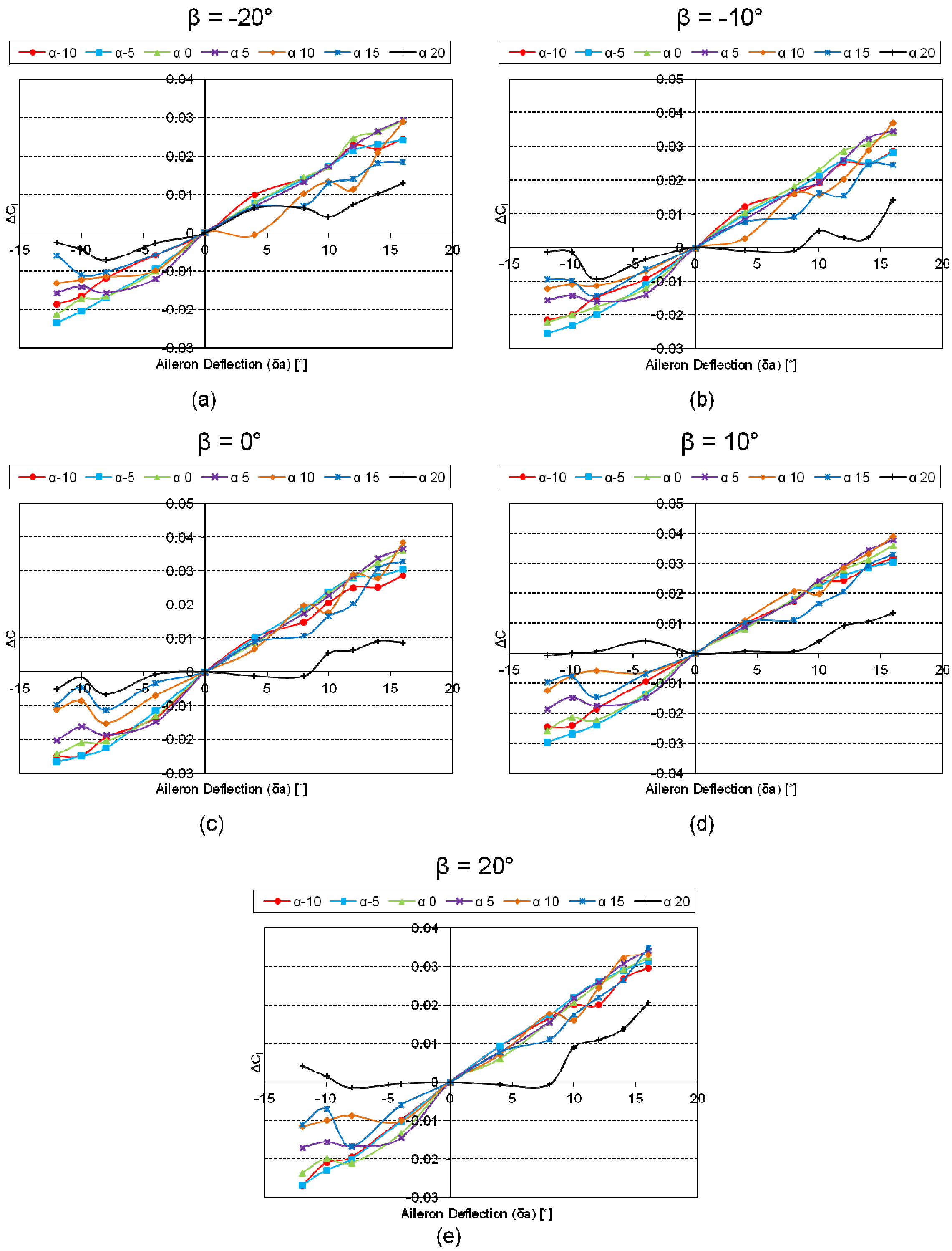


Figure 2.21: Change in pitching moment coefficient with elevator deflection, for varying angles of attack and sideslip angles

Aileron

The primary effect of the ailerons is to cause a change in rolling moment coefficient; however, a significant pitching moment and change in lift coefficient were also present. The change in rolling moment, presented in Figure 2.22 (a) to (e), is largely linear, with some deviation in the data noted at high angles of attack. Of note is the large change in slope seen at high angles of attack and sideslip angles, indicating the necessity of detailed modelling of the control effects.



(a) (b) (c) (d) (e)

Figure 2.22: Change in rolling moment coefficient with deflection of a single aileron, for varying angles of attack and sideslip angles

The change in pitching moment coefficient with aileron deflection is presented in Figure 2.23 (a) to (e). The figures indicate a trend that is largely linear with a sudden change in control characteristics being seen at an angle of attack of 20°. This is likely to be caused by a stalled wing condition. At lower angles of attack, a first order polynomial was found to describe the effect of the aileron on pitching moment adequately. However, at large angles of attack, a higher order polynomial was used, the details of which can be found in the flight dynamic code in Appendix C.

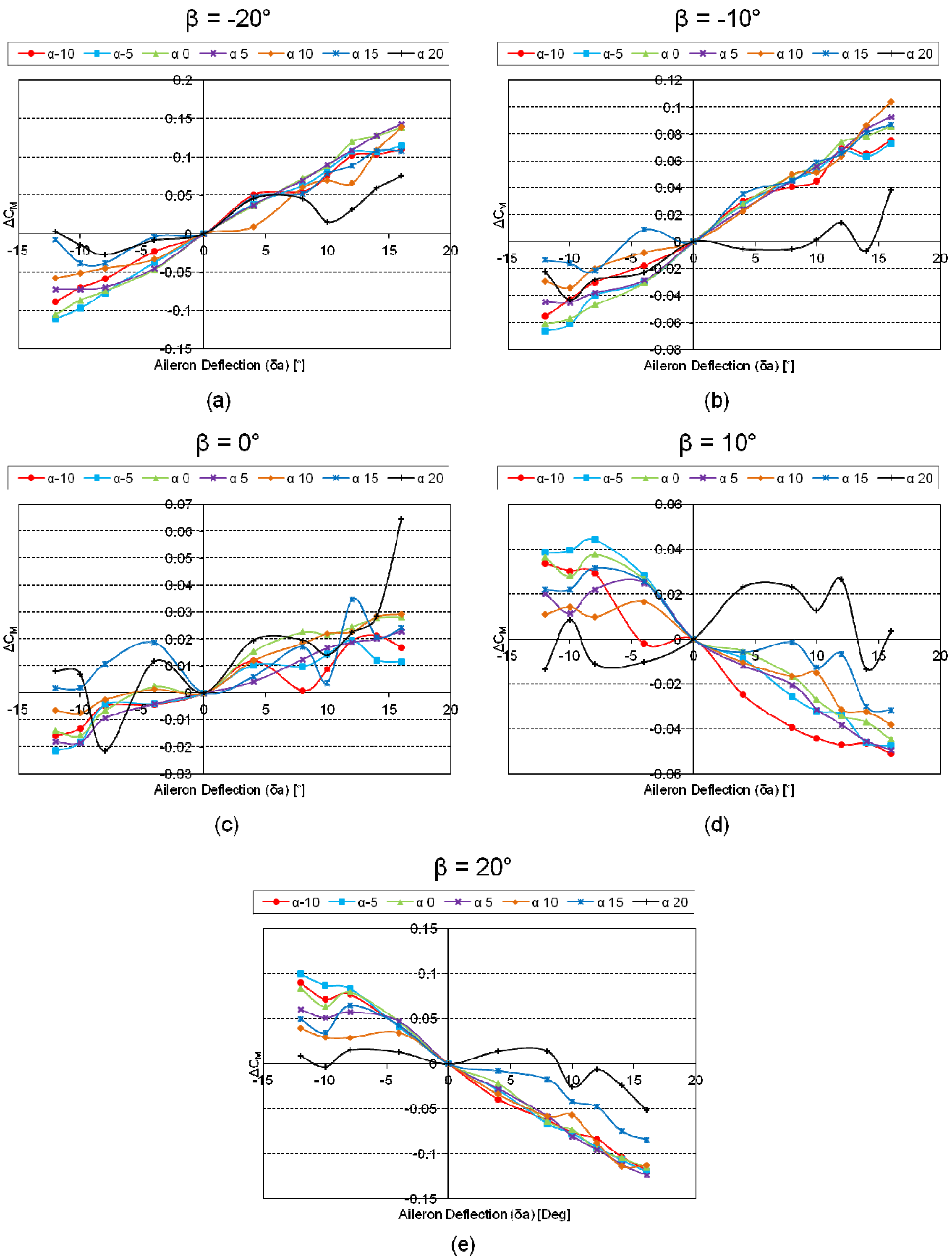
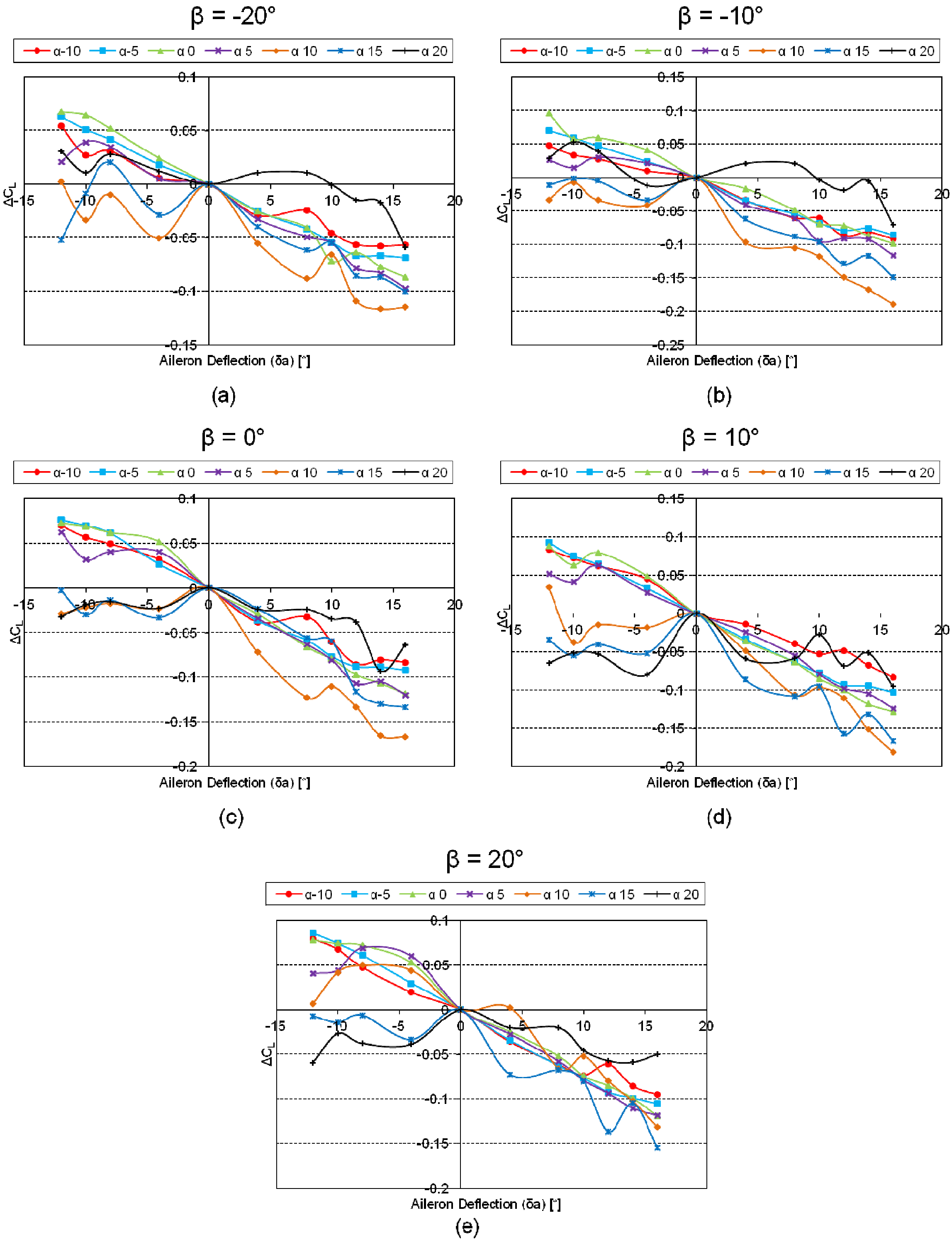


Figure 2.23: Change in pitching moment coefficient with deflection of a single aileron, for varying angles of attack and sideslip angles

The change in lift coefficient caused by the ailerons was also found to vary with a largely linear trend as shown in Figure 2.24 (a) to (e). Similarly to the pitching moment case, a first-order polynomial was used to describe the change in lift coefficient for small angles of attack; however, at larger angles of attack an increase in polynomial terms was required. The details can be found in Appendix C.



β = -20°

(a)

β = -10°

(b)

β = 0°

(c)

β = 10°

(d)

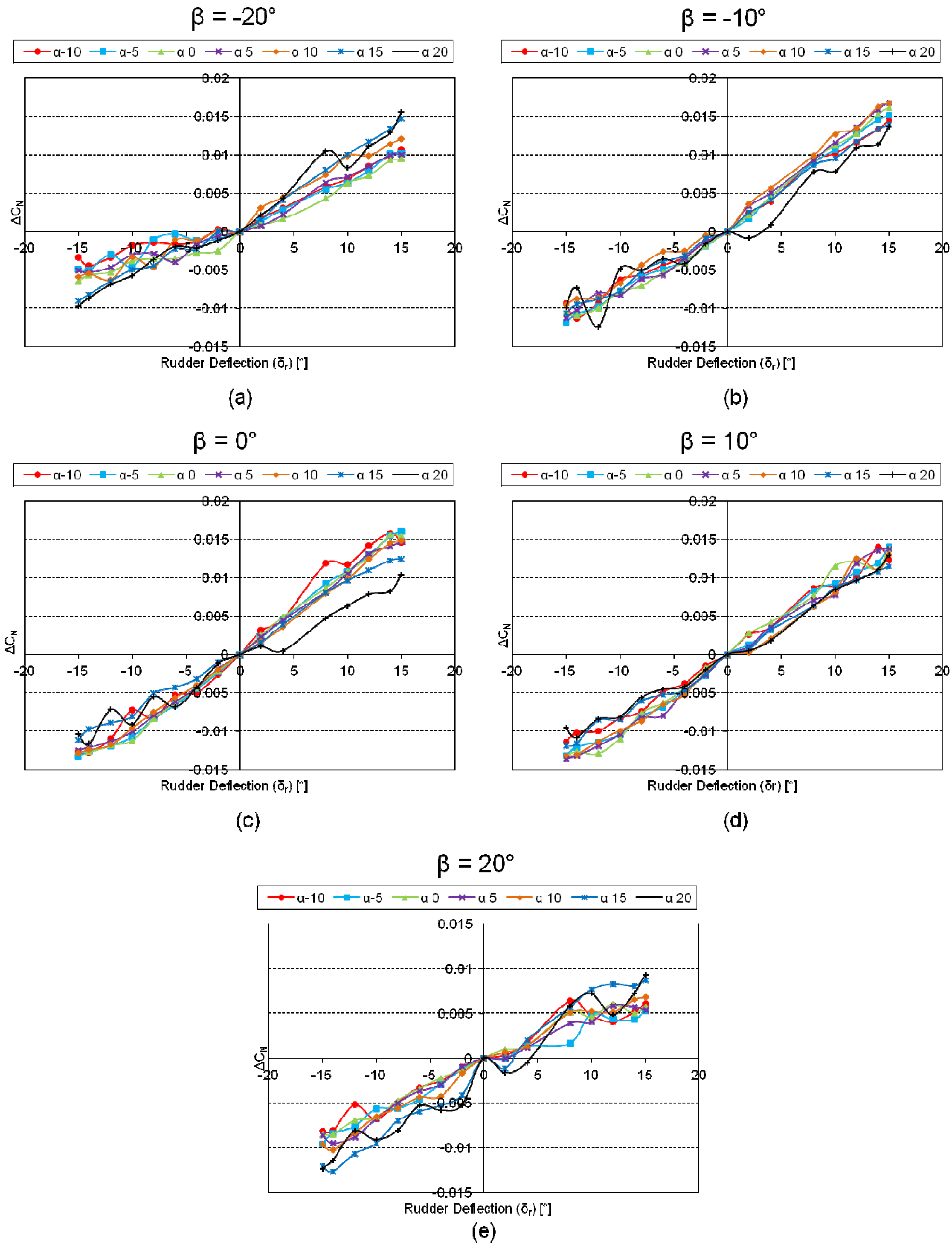
β = 20°

(e)

Figure 2.24: Change in lift coefficient with deflection of a single aileron, for varying angles of attack and sideslip angles

Rudder

The primary effect of rudder was to create a change in yawing moment coefficient which was largely linear across the entire range of sideslip angles, as shown in Figure 2.25 (a) to (e), although minor variations existed at very high sideslip angles. In the case of yawing moment coefficient, a first-order polynomial was used for all flight conditions.



(a)

(b)

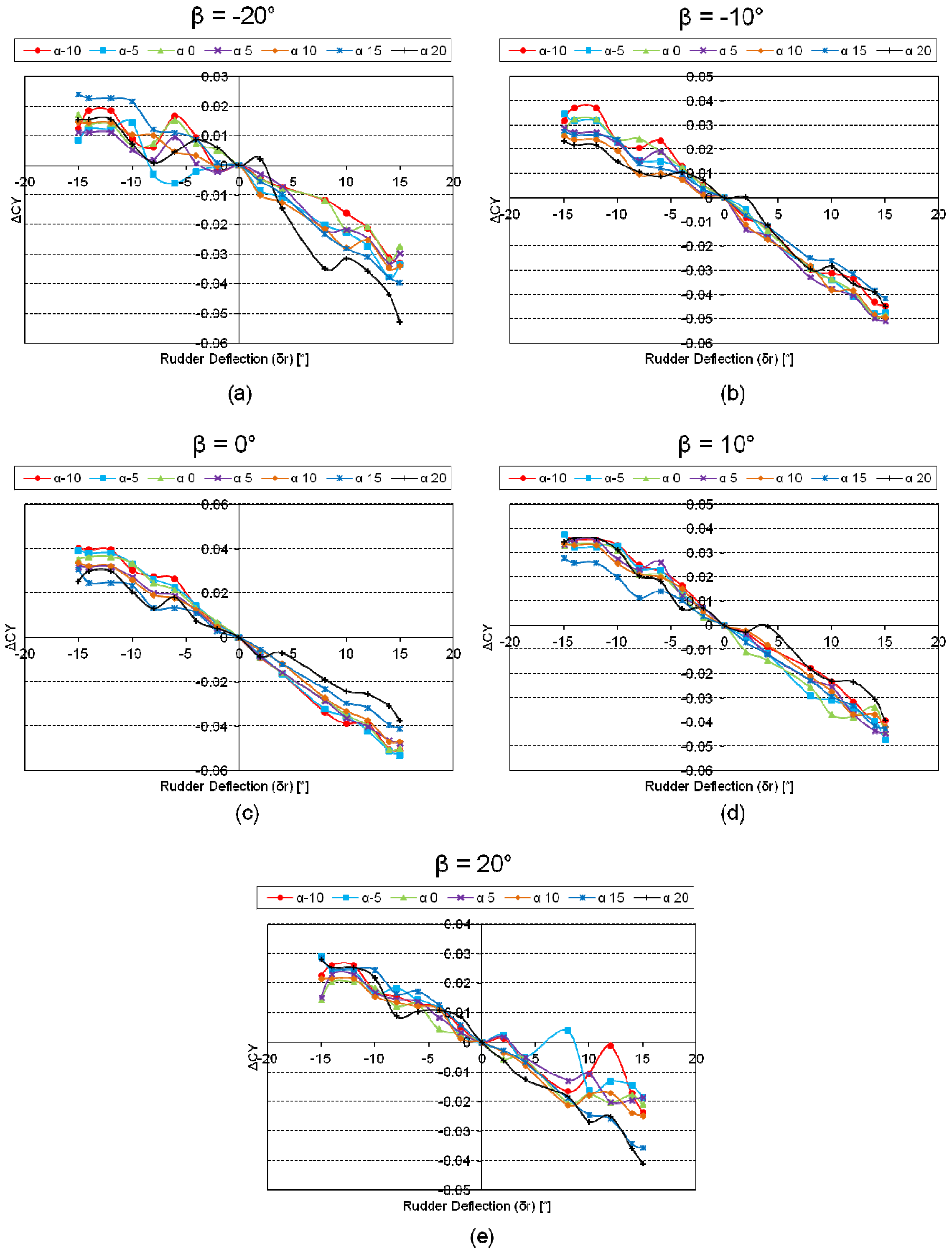
(c)

(d)

(e)

Figure 2.25: Change in yawing moment coefficient with deflection of a single rudder, for varying angles of attack and sideslip angles

The rudder also produced a pronounced change in side force coefficient. This was also found to be largely linear with a subtle change in slope noticeable, as shown in Figure 2.26 (a) through (e). This was captured in the flight dynamic code, by using a fourth-order polynomial. Again, the details of this can be found in the flight dynamic code presented in Appendix C.



(a)

(b)

(c)

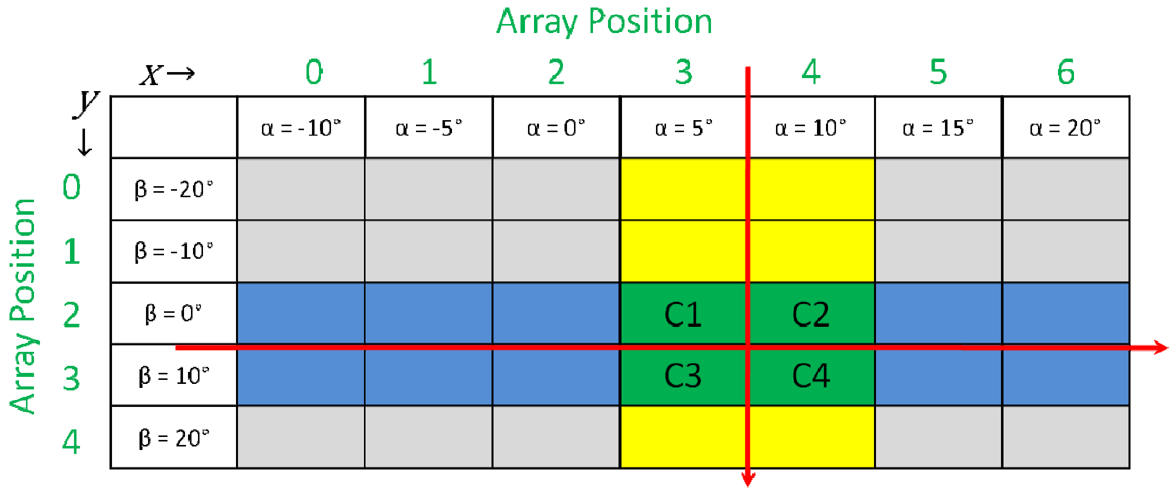
(d)

(e)

Figure 2.26: Change in side force coefficient with deflection of a single rudder, for varying angles of attack and sideslip angles

2.8.5 Data Retrieval

Each aerodynamic coefficient, whether it be for the baseline aerodynamic characteristics or the control data, varies with either angle of attack or sideslip angle or both. The data was reduced into a set of polynomials, the coefficients of which were tabulated. For the control characteristics, the table was of the form shown in Figure 2.27, while a simplified table was required of the baseline aerodynamic characteristics. Because the control tabulation and interpolation was the most complex, it will be described in more detail, while the baseline results are interpolated using a simplified process to that described below.



	α = -10°	α = -5°	α = 0°	α = 5°	α = 10°	α = 15°	α = 20°
β = -20°							
β = -10°							
β = 0°				C1	C2		
β = 10°				C3	C4		
β = 20°							

Figure 2.27: Diagram indicating the arrangement of the aerodynamic data arrays used in the simulation code

To find the correct polynomial trend, the angle of attack and sideslip angle were entered into the data retrieval algorithm. This was used to determine the positions of the correct polynomial coefficients to use in the interpolation of the data. This is done, aided by Figure 2.27, using Eqs. (2.69) to (2.72).

$$x_1 = RoundDown(0.2\alpha + 2) \tag{2.69}$$

$$x_2 = RoundUp(0.2\alpha + 2) \tag{2.70}$$

$$y_1 = RoundDown(0.1\beta + 2) \tag{2.71}$$

$$y_2 = RoundUp(0.1\beta + 2) \tag{2.72}$$

Once the polynomial coefficients have been found for all terms of the polynomial, four different values of aerodynamic coefficient are determined using:

$$C1_{aero} = a1_1\delta_{Control} + a1_2\delta_{Control}^2 + \cdots + a1_6\delta_{Control}^6 \tag{2.73}$$

$$C2_{aero} = a2_1\delta_{Control} + a2_2\delta_{Control}^2 + \cdots + a2_6\delta_{Control}^6 \tag{2.74}$$

$$C3_{aero} = a3_1\delta_{Control} + a3_2\delta_{Control}^2 + \cdots + a3_6\delta_{Control}^6 \tag{2.75}$$

$$C4_{aero} = a4_1\delta_{Control} + a4_2\delta_{Control}^2 + \cdots + a4_6\delta_{Control}^6 \qquad (2.76)$$

where $a1_1$ refers to the coefficient of the first term of the polynomial, at C1.

A simple linear interpolation is then used, firstly between $C1_{aero}$ and $C2_{aero}$ and then between $C3_{aero}$ and $C4_{aero}$ using:

$$CA_{aero} = \frac{(C2_{aero} - C1_{aero})}{5}(\alpha - \alpha_0) + C1_{aero} \qquad (2.77)$$

$$CB_{aero} = \frac{(C4_{aero} - C3_{aero})}{5}(\alpha - \alpha_0) + C3_{aero} \qquad (2.78)$$

Finally, interpolation is conducted between $CA_{aero}\ and\ CB_{aero}$, resulting in the final coefficient to be used in the coefficient build up equations described in Section 2.7.2.

$$C_{aero} = \frac{CB_{aero} - CA_{aero}}{10}(\beta - \beta_0) + CA_{aero} \qquad (2.79)$$

This process was repeated for all baseline characteristics and all control characteristics, the details of which can be found in Appendix C.

2.8.6 Dynamic Effects

Unfortunately, the dynamic effects were not directly measured in the wind tunnel. However, they are extremely important in describing the dynamic behaviour of an aircraft. In order to determine these characteristics, a number of different techniques were used as outlined in the preceding sections.

Ultimately, the variation of all the aerodynamic coefficients, with pitch rate, roll rate, yaw rate, and rate of change of angle of attack are required. Some of these can be neglected as they can be considered to be negligible when compared to the other effects.

Aerodynamic Variations with Pitch Rate (q)

The aerodynamic variations with pitch rate stem largely from the effects of the horizontal tail, with some contribution from the wings and fuselage. The change in lift coefficient can be determined by examining the change in flow pattern at the tail, shown in Figure 2.28.

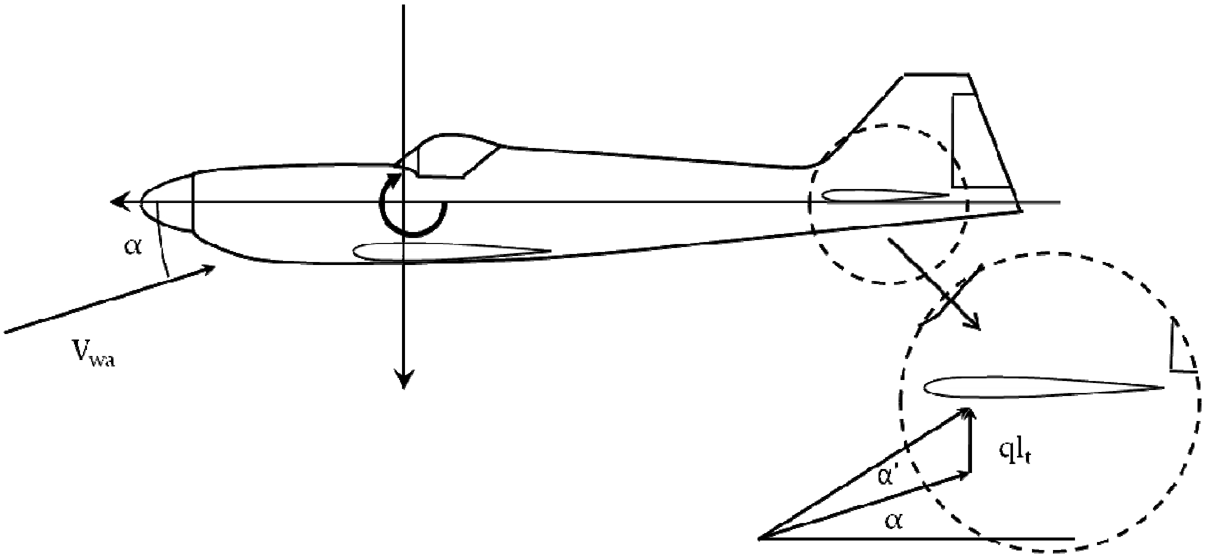


Figure 2.28: Diagram showing the change in local flow at the tail based on a pitch rate

The change in flow around the tail results in a change in total lift and change in pitching moment, which can be non-dimensionalised to a change in lift coefficient and change in pitching moment coefficient. The change in lift coefficient is given, approximately, and assuming that the majority of the change in lift is generated due to the tail, by[41]:

$$\Delta C_L = \Delta C_{L_T} \frac{S_T}{S_{Ref}} \eta_T \tag{2.80}$$

The change in tail coefficient is determined based on the change in angle of attack at the tail as shown in Figure 2.28. The change in angle of attack of the tail is given approximately by:

$$\alpha_{Tail} = \alpha + \alpha' \tag{2.81}$$

where,

$$\alpha' = \arctan\left(\frac{l_t q}{V_{wa}}\right) \tag{2.82}$$

and l_t is the tail moment arm.

To determine the change in lift coefficient correctly, a lookup function to determine the lift curve slope at the modified tail angle of attack is required:

$$\Delta C_{L_T} = a_{Tail_{@(\alpha+\alpha')}} \arctan\left(\frac{l_T q}{V_{wa}}\right) \tag{2.83}$$

Finally, the change in lift coefficient due to pitch rate can be determined as follows:

$$\Delta C_{L_q} = a_{Tail_{@(\alpha+\alpha')}} \arctan\left(\frac{l_T q}{V_{wa}}\right) \frac{S_T}{S_{Ref}} \eta_T \tag{2.84}$$

where S_T is the tail area and η_T is a tail efficiency factor. From this and assuming that the change in pitching moment with pitch rate is also primarily due to the horizontal tail, we can deduce that the change in pitching moment can be given as [41]:

$$\Delta C_{M_q} = \Delta C_{L_q}\frac{l_T}{\bar{c}} = a_{T_{@(\alpha+\alpha')}} \arctan\left(\frac{l_T q}{V_{wa}}\right)\frac{S_T}{S_{Ref}}\eta_T\frac{l_T}{\bar{c}} \tag{2.85}$$

where $\bar{c}$ is the mean aerodynamic chord of the wing. It is assumed that the change in drag with pitch rate is negligible and can be ignored. Similarly, it is assumed that there will be no real change in side force, rolling moment and yawing moment with pitch rate. Thus these have been set to zero.

In summary, Eqs. (2.86) to (2.91) are used to determine the change in the aerodynamic coefficients due to pitch rate.

$$\Delta C_{L_q} = a_{T_{@(\alpha+\alpha')}} \arctan\left(\frac{l_T q}{V_{wa}}\right)\frac{S_T}{S_{Ref}}\eta_T \tag{2.86}$$

$$\Delta C_{D_q} = 0 \tag{2.87}$$

$$\Delta C_{\mathrm{M_q}} = a_{T_{@(\alpha+\alpha')}} \arctan\left(\frac{l_T q}{V_{wa}}\right)\frac{S_T}{S_{Ref}}\eta_T\frac{l_T}{\bar{c}} \tag{2.88}$$

$$\Delta C_{\mathrm{Y_q}} = 0 \tag{2.89}$$

$$\Delta \mathrm{C}_{\mathrm{l_q}} = 0 \tag{2.90}$$

$$\Delta C_{\mathrm{N_q}} = 0 \tag{2.91}$$

In order to use this, the properties of the horizontal tail are required. The properties of the horizontal tail were extracted from the wind tunnel data by taking the sum of moments about the centre of gravity, Eq. (2.92) [37], as shown in Figure 2.29.

$$\circlearrowleft_{+ve} \quad M_{cg} = M_{0_{WB}} + L_{WB}(h - h_0)\bar{c} - l_T L_T + M_{0_T} \tag{2.92}$$

where h refers to a percentage chord position of the centre of gravity and h_0 refers to the percentage chord position of the aerodynamic centre. M and L refer to the moments and lift generated by the various surfaces and are defined in Figure 2.29.

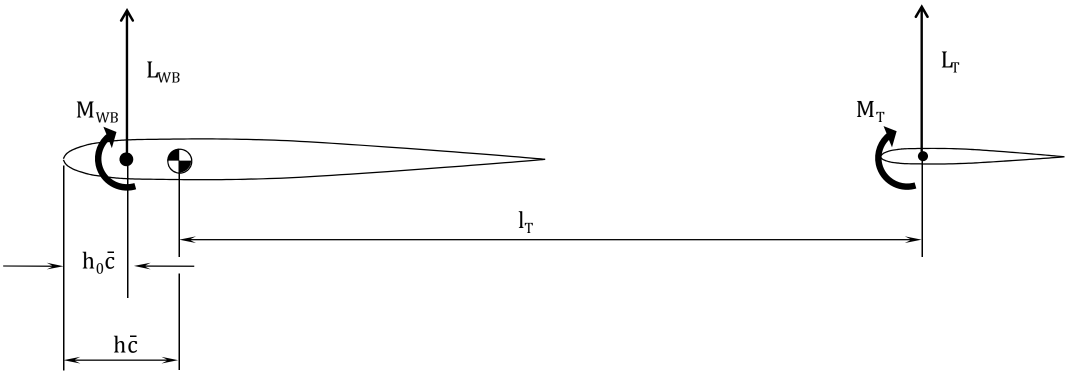


Figure 2.29: Force diagram of the forces in the X-Z plane of a typical aircraft configuration

The wind tunnel data records total lift. Hence, we can break up the lift into components due to the wing and to the tail, using [37]:

$$\uparrow_{+ve} \;\; L_{Total} = L_{WB} + L_T \tag{2.93}$$

$$\therefore L_{WB} = L_{Total} - L_T \tag{2.94}$$

Substituting this into the moment equation above, results in:

$$M_{cg} = M_{0_{WB}} + M_{0_T} + \; L_{Total}(h - h_0)\overline{c} - \; \big((h - h_0)\overline{c} + l_T\big)L_T \tag{2.95}$$

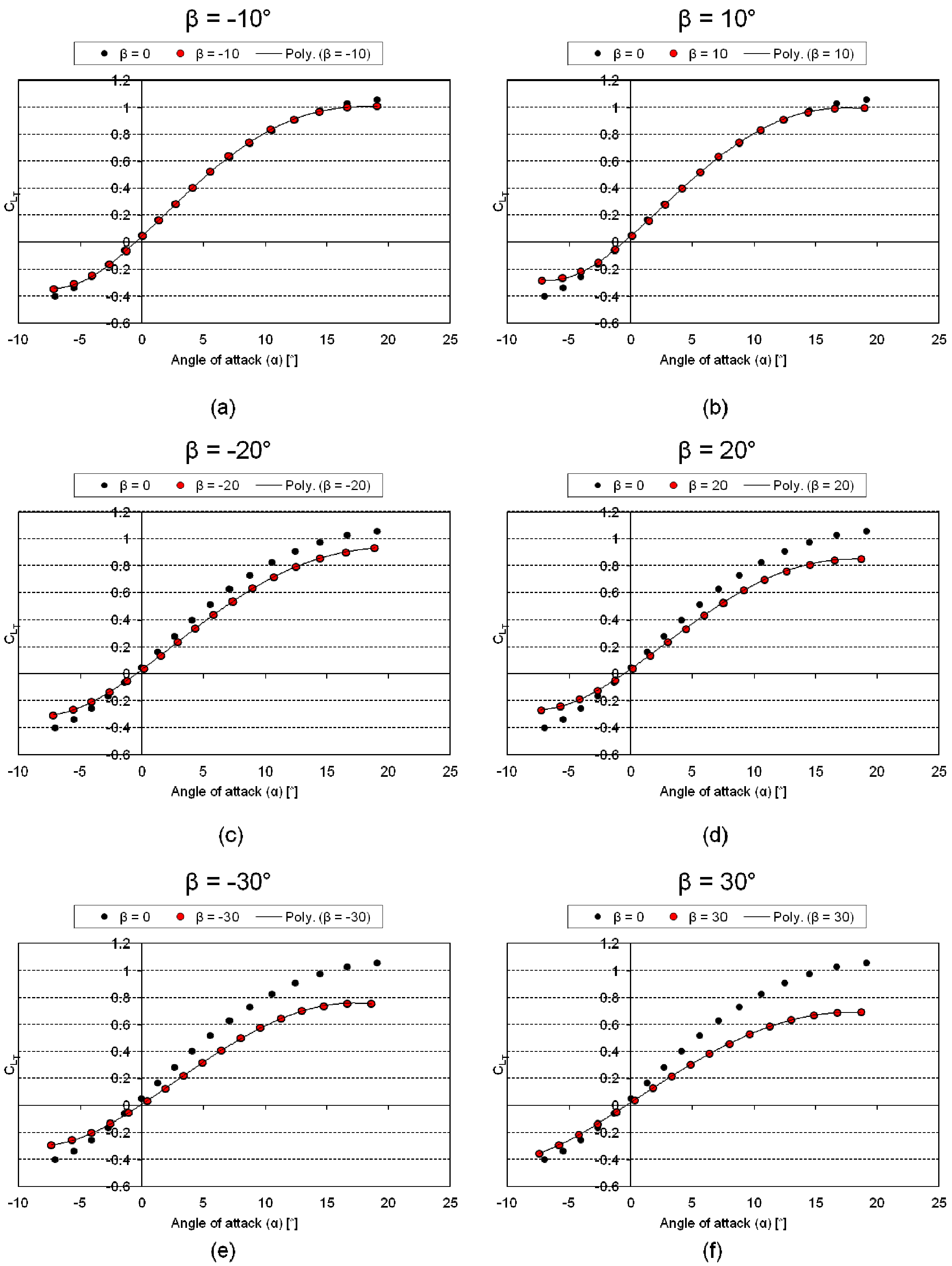
Converting to coefficient form and rearranging results in an expression to convert total lift and moment coefficients to an estimate for tail lift coefficient, we obtain

$$C_{\mathrm{L_T}} = \frac{C_{m0} + \; C_{L_{Total}}(h - h_0) - C_{m_{cg}}}{(h - h_0)\dfrac{S_T}{S_{Ref}} + V_T} \tag{2.96}$$

Using the results from this equation, one can extract the wing-body lift coefficient variation, using

$$C_{\mathrm{L_{wb}}} = C_{L_{Total}} - C_{\mathrm{L_T}}\frac{S_T}{S_{Ref}} \tag{2.97}$$

If this equation is used for each wind tunnel data point, the tail lift characteristics, shown in Figure 2.30, can be determined.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 2.30: Horizontal tail lift coefficient variation with angle of attack at various sideslip angles as extracted from the wind tunnel data

Aerodynamic Variations with Roll Rate (p)

The aerodynamic variations with roll rate are very important to describe the roll damping of the aircraft. It is also more difficult to determine when compared to the effects of pitch rate and yaw rate as the roll rate causes a change in velocity that varies across the wing span. This in turn causes a change in angle of attack which varies along the wing span. Thus, in order to determine the effect of roll rate, a strip theory approach was used as shown in Figure 2.31. The wing was divided into chordwise strips of width dy, and the change in angle calculated at each station. The change in lift was then determined for each station and a numerical integration performed to estimate the changes in lift and rolling moment.

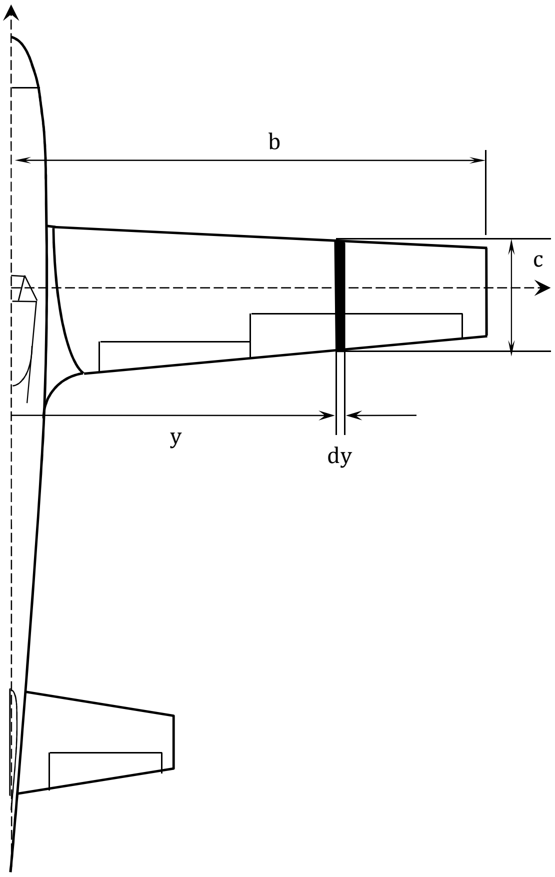


Figure 2.31: Diagram showing the nomenclature and conventions used in strip theory to determine the variation on aerodynamic coefficients with roll rate

Examining the aerofoil section at a point y, as shown in Figure 2.31, it can be shown that the aerofoil, already at an angle of attack α, experiences a change in angle of attack. Assuming the roll is to the right and looking at the right wing, the change in angle of attack, from Figure 2.32, is given by:

$$\Delta\alpha = \alpha + \alpha' \tag{2.98}$$

where,

$$\alpha' = \arctan\left(\frac{py}{V_{wa}}\right) \tag{2.99}$$

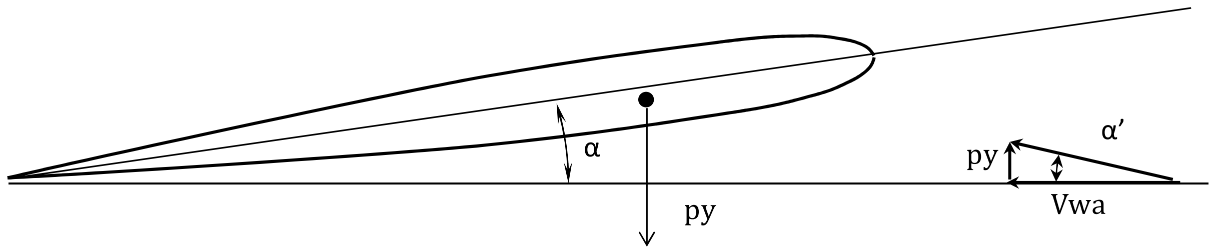


Figure 2.32: Side view of an aerofoil section undergoing a roll, or downward velocity

Once this change in angle has been determined, the changes in lift coefficient and rolling moment can be determined using Eqs. (2.100) and (2.101) adapted from [37]. In these cases the drag term was neglected.

$$\Delta C_{L_p} = \frac{1}{S_{Ref}} \int_{-b/2}^{\frac{b}{2}} \left(\bar{c}a_{wb_{@(\alpha+\alpha')}} \arctan\left(\frac{py}{V_{wa}}\right) \cos\left(\arctan\left(\frac{py}{V_{wa}}\right)\right) + C_{D_y} \sin\left(\arctan\left(\frac{py}{V_{wa}}\right)\right)\right) dy \tag{2.100}$$

$$\Delta C_{M_p} = \frac{1}{S_{Ref}} \int_{-b/2}^{\frac{b}{2}} \left(\bar{c}a_{wb_{@(\alpha+\alpha')}} \arctan\left(\frac{py}{V_{wa}}\right) \cos\left(\arctan\left(\frac{py}{V_{wa}}\right)\right) + C_{D_y} \sin\left(\arctan\left(\frac{py}{V_{wa}}\right)\right)\right) ydy \tag{2.101}$$

This is then integrated, numerically, along the span of the wing using Simpson's integration. A similar process was followed for the change in side force coefficient, with the assumption that the majority of the change in side force is generated by the vertical tail. Thus the integration is performed from the base of the fin to the top of the fin.

The change in side force coefficient is given by (2.102) also adapted from [37].

$$\Delta C_{Y_p} = \frac{1}{S_{Ref}} \int_{0}^{h} \left(\overline{c_{Vt}} a_{Vt_{@(\beta+\beta')}} \arctan\left(\frac{ph}{V_{wa}}\right)\right) dh \tag{2.102}$$

Aerodynamic Variations with Yaw Rate (r)

The aerodynamic variations with yaw rate stem largely from the vertical fin and wings. It is assumed that the effect of the fuselage is negligible. Three main changes occur with yaw rate: first, the vertical fin causes a change in side force which, in turn, results in a yawing moment change. In a similar manner to that of pitch rate, the angle of the vertical fin changes with yaw rate and there is also a rolling moment change due to the difference in velocity along the span of the wing. It is assumed that there is no contribution to drag, lift and pitching moment. The change in side force with yaw rate is largely due to the vertical fin, with some effects from the fuselage. However, it is thought that the effect of the vertical fin far outweighs that of the fuselage. Thus, the effect of the fuselage is neglected for this case. The change in side force can be determined with the aid of Figure 2.33.

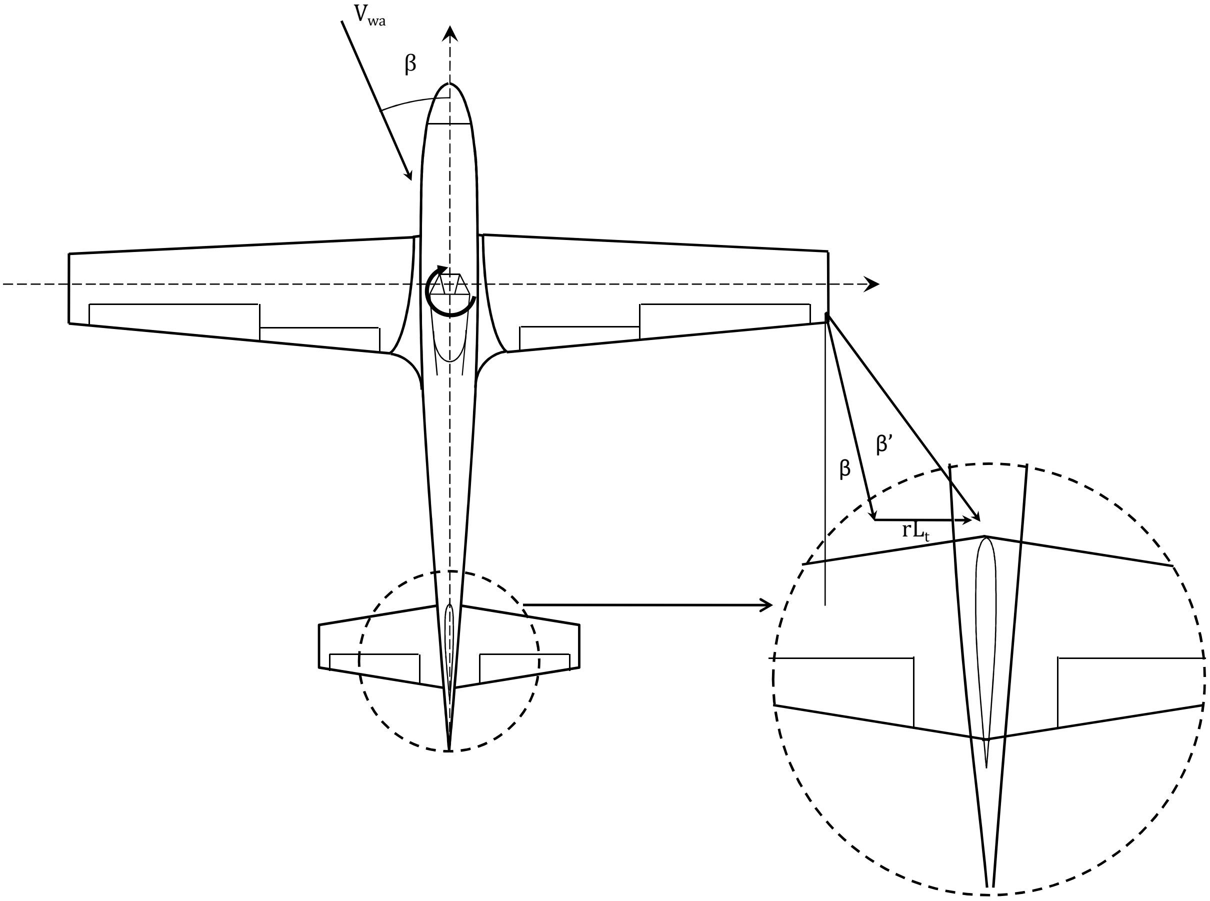


Figure 2.33: Diagram showing the effect of yaw rate on the local flow direction at the tail

A positive yaw to the right results in a movement of the vertical tail to the left when looking at Figure 2.33. This increases the local angle of attack at the tail, resulting in an increase in lift force to the right, as well as inducing a yawing moment that tries to align the aircraft to the wind vector. Thus the side force is positive, while the yawing moment is in the negative sense.

In a similar manner to the pitching moment derivatives, the lift curve slope of the vertical tail at the increased angle is examined, but the total contribution is limited by the change in angle due to the yawing moment.

Thus

$$\Delta C_Y = \Delta C_{L_{VT}} \frac{S_{VT}}{S_{Ref}} \eta_{VT} \tag{2.103}$$

and the angle at the tail is:

$$\beta_{Tail} = \beta + \beta' \tag{2.104}$$

where,

$$\beta' = \arctan\left(\frac{l_{VT} r}{V_{wa}}\right) \tag{2.105}$$

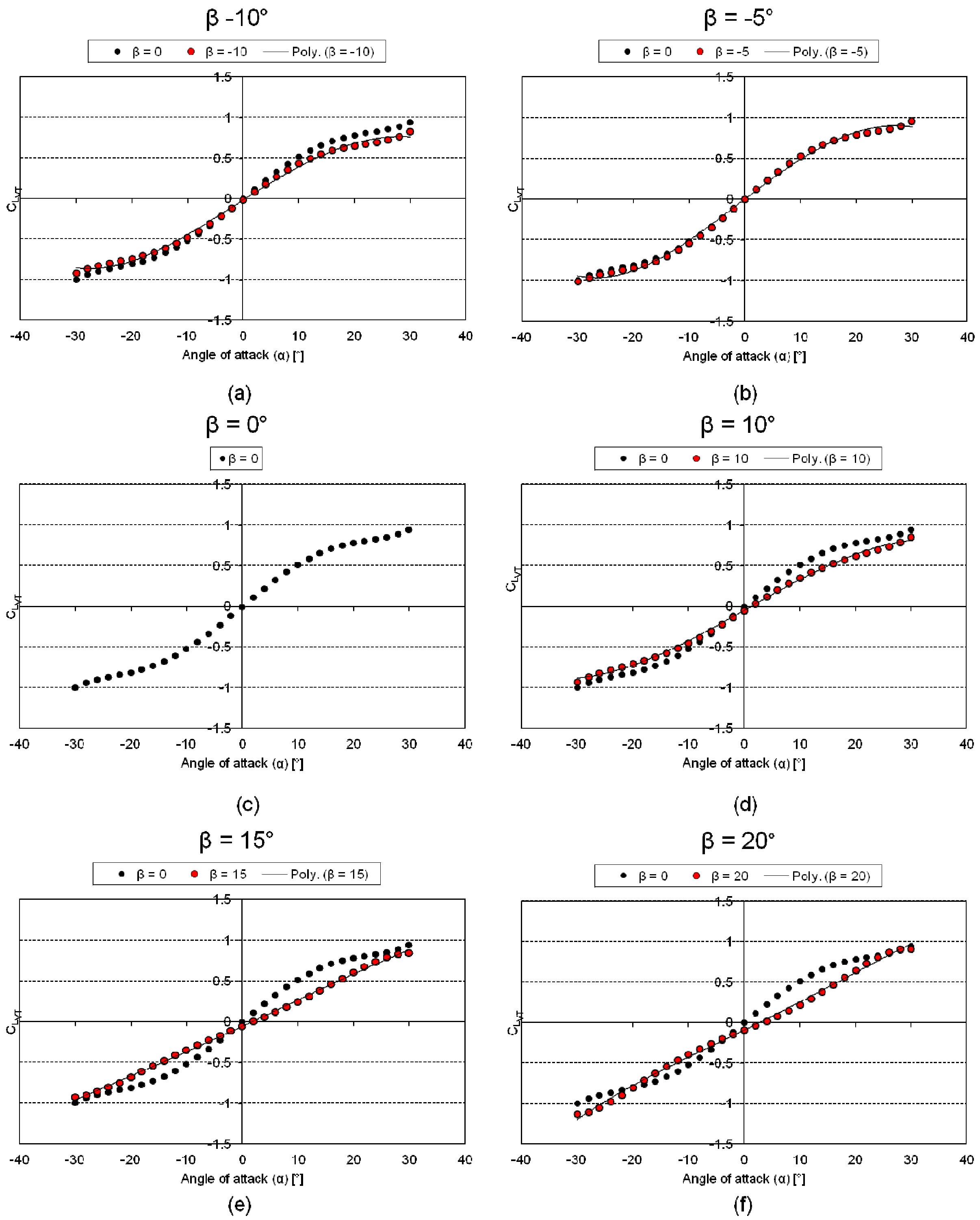
This allows the determination of the change in side force coefficient using,

$$\Delta C_{Y_r} = a_{VT_{@(\beta+\beta')}} \arctan\left(\frac{l_{VT} r}{V_{wa}}\right) \frac{S_{VT}}{S_{Ref}} \eta_{VT} \tag{2.106}$$

From this and assuming that the change in yawing moment with yaw rate is also primarily due to the vertical tail, we can deduce that the change in yawing moment coefficient is given by

$$\Delta C_{\mathrm{N_r}} = -\Delta C_{Y_r} \frac{l_{VT}}{b} = -a_{VT_{@(\beta+\beta')}} \arctan\left(\frac{l_{VT} r}{V_{wa}}\right) \frac{S_{VT}}{S_{Ref}} \eta_{VT} \frac{l_{VT}}{b} \tag{2.107}$$

Similarly to the pitching moment equation, the characteristics of the vertical tails are required. This was achieved by following a similar procedure to the horizontal tail data extraction. Instead of examining lift and pitching moment, the data pertaining to the side force and yawing moments of the aircraft were used. The detailed procedure is not detailed here; however, the characteristics of the vertical tails are presented in Figure 2.34.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 2.34: Vertical tail lift coefficient variation with angle of attack at various sideslip angles as extracted from the wind tunnel data

Aerodynamic Variations with Rate of Change of Angle of Attack ($\dot{\alpha}$)

The aerodynamic variations with rate of change in angle of attack stem primarily from a time lag effect of the downwash at the tail. This is determined using (2.108), as outlined in [41].

$$\Delta C_L = 2a_{Tail}\left(\frac{lt}{\bar{c}}\right)\left(\frac{2CL}{\pi ARe}\right)\left(\frac{S_T}{S_{Ref}}\right)\eta_T\dot{\alpha} \tag{2.108}$$

The change in pitching moment is then given by (2.109) [41].

$$\Delta C_M = 2a_{Tail}\left(\frac{lt}{\bar{c}}\right)^2\left(\frac{2CL}{\pi ARe}\right)\left(\frac{S_T}{S_{Ref}}\right)\eta_T\dot{\alpha} \tag{2.109}$$

The variation of the other aerodynamic parameters with rate of change in angle of attack is considered negligible and has been excluded from the flight dynamic analysis.

2.9 Navigation

The navigation model used in the flight dynamic model provides two outputs that are useful in following the flight path of the aircraft. The first model is centred about an arbitrary point on the Earth's surface and assumes that the Earth is flat and that the distance travelled in the time step is small. This is useful to examine the actual ground path taken by the aircraft. The second navigation model calculates the latitude and longitude as if the aircraft were actually flying. This is then sent to the autopilot for its navigational algorithms.

To determine the distance travelled from an arbitrary origin, the ground velocity in the north and easterly directions is required. This is determined through an axis transformation as discussed in Section 2.4, from which the velocity in the north direction can be derived. This is given as :

$$\begin{aligned} U_e \;=\; & cos(\psi)cos(\theta)U_b \;+\; \big(cos(\psi)sin(\theta)sin(\phi) - sin(\psi)cos(\phi)\big)V_b \\ & +\; \big(cos(\psi)sin(\theta)cos(\psi) + sin(\psi)sin(\phi)\big)W_b \;+\; V_{N\,Wind} \end{aligned} \tag{2.110}$$

The velocity in the east direction is given as:

$$\begin{aligned} V_e \;=\; & sin(\psi)cos(\theta)U_b \;+\; \big(sin(\psi)sin(\theta)sin(\phi) + cos(\psi)cos(\phi)\big)V_b \\ & +\; \big(sin(\phi)sin(\theta)cos(\phi) - sin(\psi)cos(\phi)\big)W_b \;+ V_{E\,Wind} \end{aligned} \tag{2.111}$$

Using Eqs. (2.110) and (2.111) the distance from origin is simply determined using

$$D_{North} = D_{North} + \int U_e dt \tag{2.112}$$

$$D_{East} = D_{East} \ + \int V_e dt \tag{2.113}$$

Equations (2.112) and (2.113) when plotted, provide an indication of the ground track followed by the aircraft.

For the autopilot, the latitude and longitude of the aircraft are required. This is determined using Eqs. (2.114) and (2.115), as developed in [42].

$$Latitude2 = \arcsin\ (\sin(Latitude1)\cos\left(\frac{D}{ER}\right) + \cos(Latitude1)\sin\left(\frac{D}{ER}\right)\cos\ (Track) \tag{2.114}$$

$$\begin{aligned} Longitude2 = Longitude1 + atan2(\sin(track)\sin\left(\frac{D}{ER}\right)\cos(Latitude1)\,, \cos\left(\frac{D}{ER}\right) \\ - \sin(Latitude1) * \sin\ (Latitude2)) \end{aligned} \tag{2.115}$$

where, for the equations above, D is the distance travelled (in km), ER is the radius of the earth, which has a mean value of 6371km.

To use the above equations, the distance travelled as well as the bearing are required. These are easily calculated using Eqs. (2.116) and (2.117), respectively.

$$D = \int GSdt \tag{2.116}$$

$$Track = \text{atan}\left(\frac{U_e}{V_e}\right) \tag{2.117}$$

The ground speed, GS in Eq. (2.116), is given by:

$$GS = \ \sqrt{U_e^2 + V_e^2} \tag{2.118}$$

2.10 Auxiliary Functions

2.10.1 Servo Model

The servo model attempts to simulate very basically the lag of a servo motor. This adds to the fidelity of the flight dynamic model, as the servos typically are slow in comparison to the autopilot computations. Servos, typically, have a speed specification that is dependent on voltage and specified as a single value of number of seconds to travel 60°. For the servo model, a simplistic approach was used where the position of the control surface was limited at each time step based on the speed of the servo and the maximum position that could be achieved within the time step. The logic employed by the servo model can be seen in Figure 2.35.

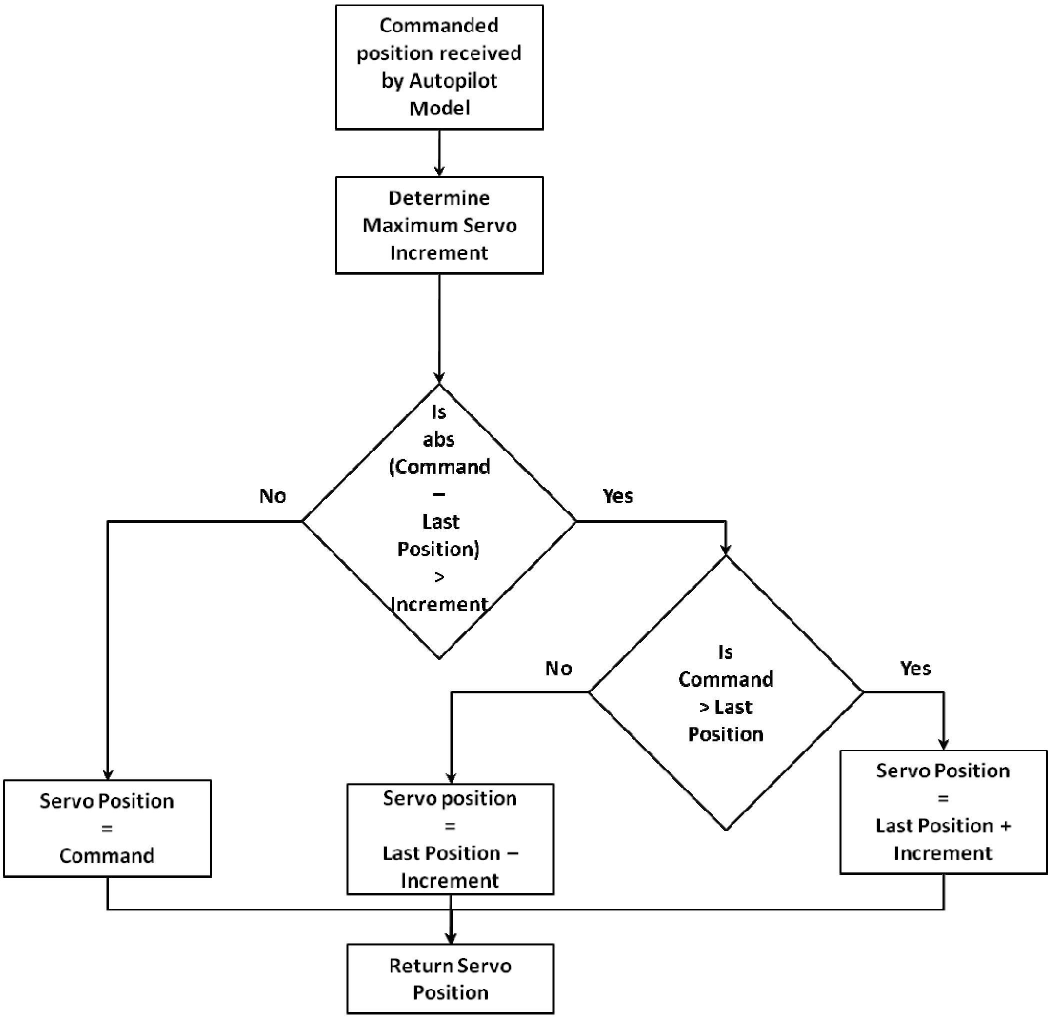

Commanded position received by Autopilot Model
Determine Maximum Servo Increment
Is abs (Command – Last Position) > Increment
No
Yes
Is Command > Last Position
No
Yes
Servo Position = Command
Servo position = Last Position – Increment
Servo Position = Last Position + Increment
Return Servo Position

Figure 2.35: Program logic of the servo motor model

2.10.2 Clip To

The "clip to" function is a simple function developed to prevent a variable from exceeding various limits. This function is used as a form of physical limit for the control surfaces where despite the command received by the autopilot, the control position will not exceed the maximum and minimum parameters. The logic used for this function is given in Figure 2.36.

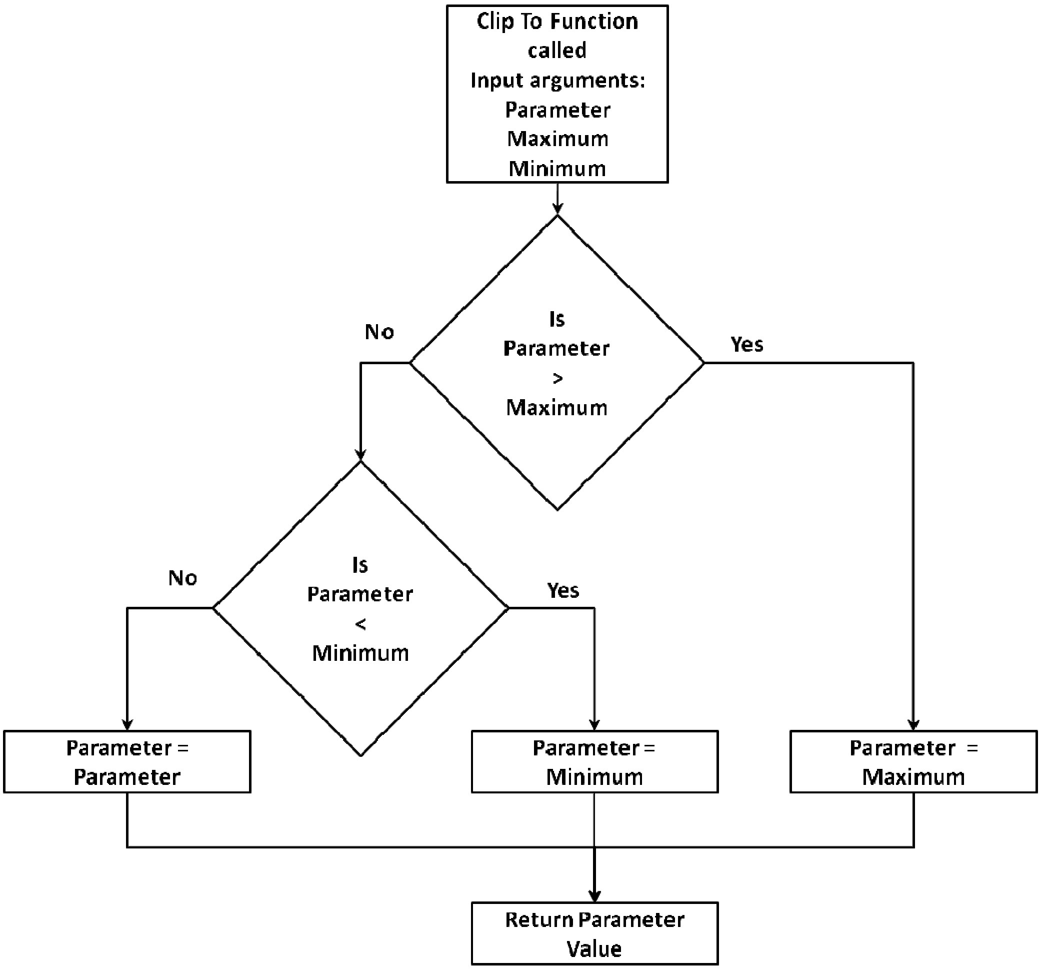


Figure 2.36: Program logic for the "Clip to" command

2.10.3 Wrap

The "wrap" function was developed primarily for the navigation functions. It is desired for the correct function of the autopilot to have the heading of the aircraft be constrained from 0° to 359° before it is parsed to the autopilot model. The program logic is shown in Figure 2.37. This function works well for small increments or increments where the heading change is less than the maximum and minimum values for that parameter, but is unable to tolerate large changes in heading, where a 360° rotation is made within one time step. This limitation is not considered significant, as is not envisaged that this case would be physically possible.

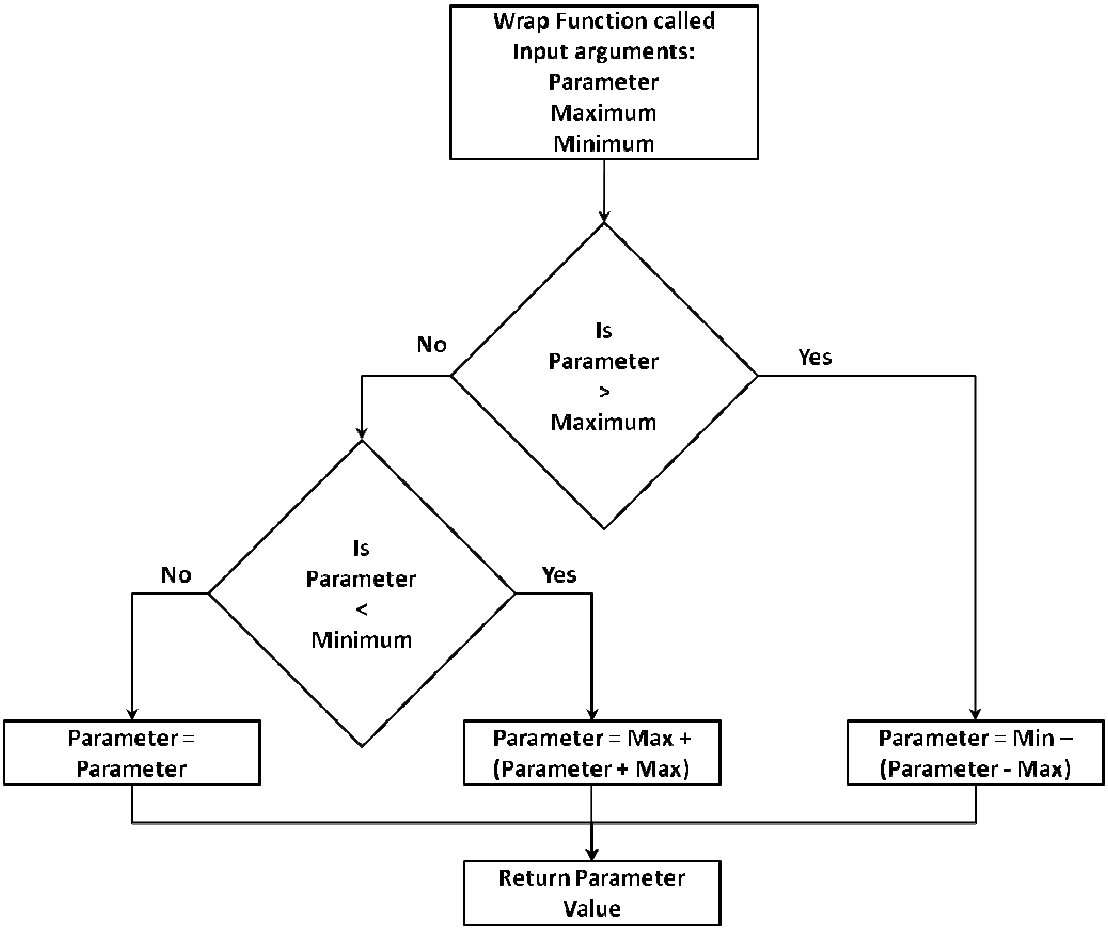


Figure 2.37: Program logic for the "Wrap" function

2.11 Concluding Remarks

The full simulation code consists of a number of functions and classes. The flight dynamic model described in the sections above indicate the basic ideas used in the code. The code is, however, far more extensive than can be described here. For details on the flight dynamic model, please refer to the "Non-Linear Aircraft" class of the simulation code, found in Appendix C.

3 Chapter 3 – Development of Autopilot Model

3.1 Chapter Outline

The Ardupilot Mega 2.0 was chosen as the platform on which to base this research. This autopilot was chosen for its low cost and open-source software code. This allows for the complete modification and examination of the current control laws. This chapter will briefly review the hardware and provide an overview of the autopilot and control structure to provide context within which to describe, in more detail, how each individual control loop works to control the aircraft. The chapter will then develop a simplified version of the autopilot code that was used to test the proposed fault-tolerant control strategy.

3.2 Ardupilot Hardware

The Ardupilot 2, shown in Figure 3.1, has at its core an ATmega 2560 Microprocessor chip [43]. This is the primary processor that stores the firmware and controls the outputs to the servos based on the inputs it receives from the various sensors. The sensors used on the autopilot include a MPU-6000 3-axis gyro and accelerometer, an MS5611 barometric pressure sensor, a Mediatek 10Hz GPS receiver, and a HMC5883L magnetometer. Other components of importance include a PPM encoder, SD card writer to store data and a 2.4 GHz Xbee wireless modem to connect to a ground station remotely. A differential pressure sensor (MPXV7002DP) is also included to measure the dynamic pressure and hence determine the airspeed of the UAS.

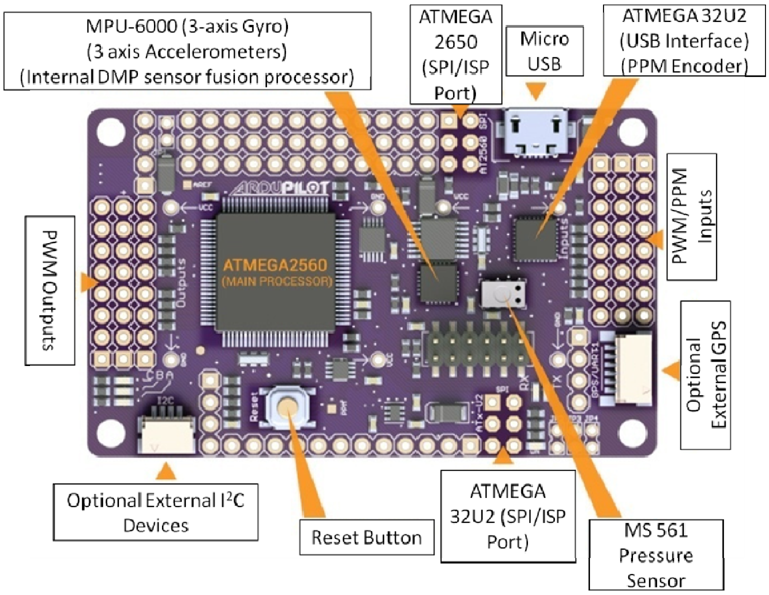

MPU-6000 (3-axis Gyro)
(3 axis Accelerometers)
(Internal DMP sensor fusion processor)
ATMEGA 2650 (SPI/ISP Port)
Micro USB
ATMEGA 32U2 (USB Interface) (PPM Encoder)
PWM/PPM Inputs
PWM Outputs
ARDUPILOT
ATMEGA2560 (MAIN PROCESSOR)
Optional External GPS
Optional External I²C Devices
Reset Button
ATMEGA 32U2 (SPI/ISP Port)
MS 561 Pressure Sensor

Figure 3.1: Ardupilot mega 2 circuit board retrieved and modified from [43]

3.3 Ardupilot Control Structure

The Ardupilot Mega makes use of cascading PID controllers to achieve the desired flight control. This starts with the inner PID loops which control the elevators and ailerons to achieve a desired pitch and roll angles. The rudder is used to control the lateral acceleration of the aircraft, ensuring a co-ordinated turn. The pitch and roll angles, used in the inner loop as set point values and termed NavPitch and NavRoll, are controlled through another PID controller that makes use of airspeed and heading as set points. Thus the elevator is used to control airspeed via a pitch angel and bank angle is controlled based on the desired heading. Finally, altitude and airspeed are used to control the position of the throttle. A block diagram showing the control structure is given in Figure 3.2.

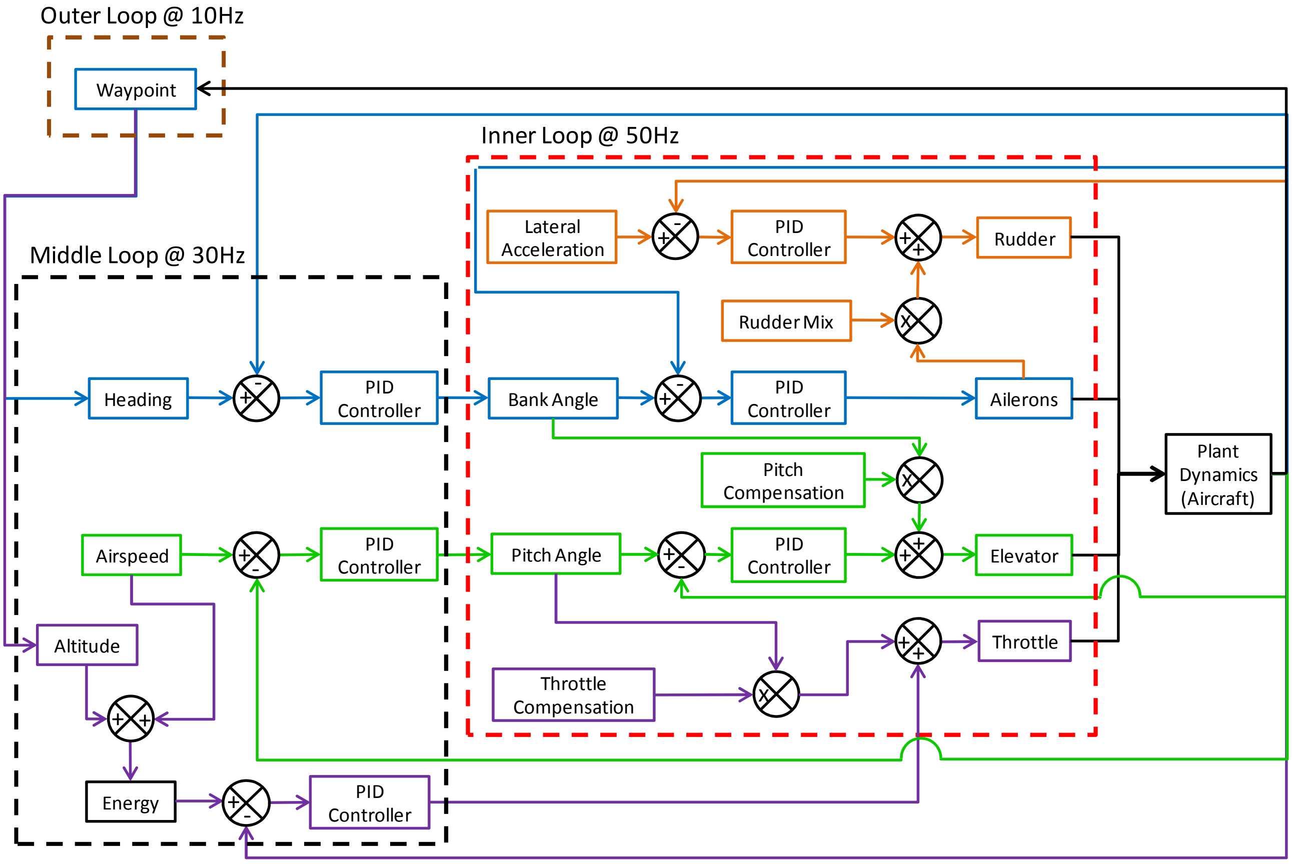


Figure 3.2: Block diagram showing the control structure of the Ardupilot mega

3.4 Autopilot Flight Modes

The Ardupilot has a number of flight modes. These flight modes are as follows.

- Stabilise
- FBW A
- FBW B
- Auto
    - Circle
    - Loiter
    - Takeoff
    - Land

Of importance to the control laws are the first four. The last four flight modes are all essentially "Auto" mode, with navigational macros being run in the background.

3.4.1 Stabilise

The "Stabilise" flight mode is the most basic flight mode available other than manual. This flight mode, when engaged, returns the aircraft to level flight. This flight mode is thus often used to train new pilots. This flight mode only makes use of the inner control loop with the NavRoll and NavPitch set points being set to 0. If a non-zero command is received from the RC transmitter, the autopilot uses that command. Hence, this mode is a form of assisted manual flight. Throttle management and navigation is performed by the pilot.

3.4.2 FBW A

The "Fly by Wire A" flight mode is similar to "Stabilise" as only the inner loop is functional. However, in this case, the set points, NavPitch and NavRoll, are set to a value other than 0 by the RC transmitter. The position of the RC transmitter indicates a proportional bank angle or pitch angle, based on the maximum allowable pitch and roll setting of the autopilot. For example, if the maximum bank angle of the autopilot is 45°, moving the aileron control stick to the far left would result in a bank angle of 45°, while moving the stick to the half way point would result in a bank angle of 22.5°. Thus "FBW A" gives a form of angle control. This makes the aircraft particularly easy to fly as the aircraft will maintain a turn with a single stick movement. However, throttle management and navigation is still performed by the pilot.

3.4.3 FBW B

The "Fly by wire B" flight mode is similar to "FBW A" in roll control but starts to include the middle loop for airspeed and altitude management. The NavPitch set point is no longer controlled by the pilot but, rather, by the airspeed while throttle is controlled by the autopilot using both kinetic energy and potential energy obtained from airspeed and altitude measurements. Thus all the pilot is required to do is navigate the aircraft.

3.4.4 AUTO

The "Auto" flight mode incorporates all control loops and only requires the operator of the system to enter waypoints and commands such as take off, land, loiter, or circle. In this case, the heading is controlled by examining the current position of the aircraft and calculating the heading to fly to get to the next waypoint.

3.5 PID Controllers

The Ardupilot makes use of a common PID controller class that searches for the correct proportional, integral and derivative gains depending on the PID controller called. There are also built-in integration wind up protectors in this class. The modified implementation of this class found in the Ardupilot code [44] is presented below.

PID control can be separated into its three components and then simply summed together; thus each section of the PID controller will be presented individually.

The proportional control is based on the error from the desired state and is generally given by:

$$Error = Setpoint - Measured \tag{3.1}$$

The error is then used to determine the output of the proportional controller. This output is given as:

$$Output = Kp \times Error \tag{3.2}$$

The integral controller examines the time history of the error and attempts to reduced the past history to a minimum. This is simply done by performing a numerical integration of the error at each time step. This is given by:

$$Integral = Integral + \frac{1}{2}(Error + PreviousError) \times dt \tag{3.3}$$

The output of the integral controller is then given by:

$$Output = Ki \times Integral = Ki \times \left[Integral + \frac{1}{2}(Error + PreviousError)dt\right] \tag{3.4}$$

A form of integral wind up protection is included in the integral controller. This is simply implemented by setting a maximum output value.

The derivative controller aims to reduce overshoot by examining the rate of change of error. This is given by:

$$Derivative = \frac{Error - Previous\ Error}{dt} \tag{3.5}$$

The output of the derivative controller is then given by:

$$Output = Kd \times Derivative = Kd\left(\frac{Error - Previous\ Error}{dt}\right) \tag{3.6}$$

The output of the completed controller is finally given by:

$$\begin{aligned} Output = {} & Kp \times (Setpoint - Measured) \\ & + Ki\left[Integral + \frac{1}{2}(Error + Previous\ Error) \times dt\right] \\ & + Kd\left(\frac{Error - Previous\ Error}{dt}\right) \end{aligned} \tag{3.7}$$

The full implementation of the PID controllers can be found in the "PID" class of the simulation program, found in Appendix C.

3.6 Inner Loop

The inner control loop, shown in Figure 3.3 , operates the control surfaces and throttle directly. The loop runs at approximately 50Hz depending on the required processing time of the ATmega Chip. However, for the simulations it has been assumed that this 50Hz is consistent and accurate. The ailerons and elevators are controlled by the NavRoll and NavPitch variables which are set points for the aircraft to fly to. A standard PID feedback loop is used to achieve the desired response characteristics. In addition to the PID loop, the elevator control is complimented with a feedforward command based on bank angle. The rudder is controlled using a PID loop with a set point of lateral acceleration, the lateral acceleration being set to 0, to achieve a coordinated turn. The rudder is also complimented by a feedforward command based on aileron deflection. The throttle control is fed from the medium speed loop and is hence more of a medium loop control, but it does receive a feed forward command based on pitch angle and hence has been included here.

In the implementation of the inner loop, the pitch error used to control the elevator is, from [44], given by:

$$\epsilon_\theta = \theta - \left(NavPitch + \phi Pitch_{Roll_{Comp}} + \delta_{Throttle} \times \eta_{Throttl\ 2\ Pitch\ Compensation}\right) \tag{3.8}$$

While the roll and yaw errors are respectively given by:

$$\epsilon_\phi = \phi - NavRoll \tag{3.9}$$

$$\epsilon_\psi = 0 - a_y \tag{3.10}$$

These errors are fed into the inner loop PID controllers and the outputs sent to the control surfaces.

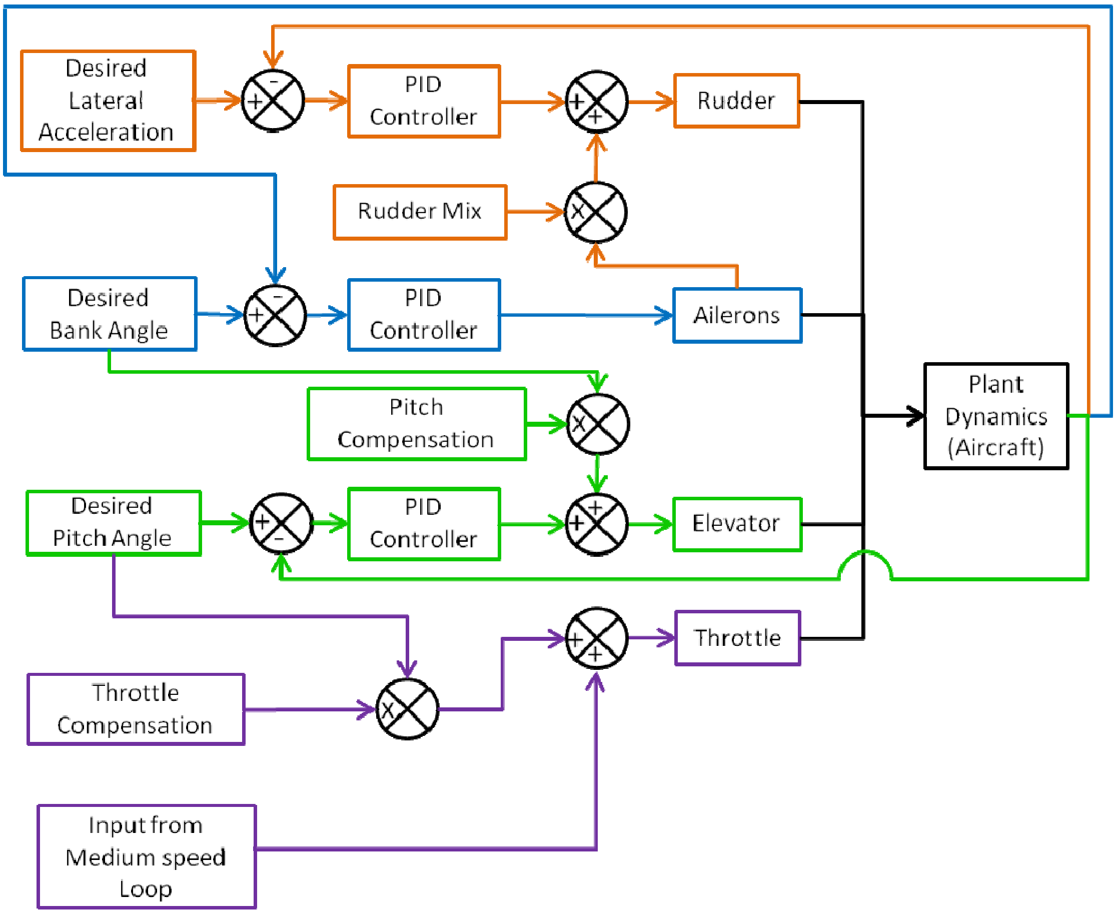


Figure 3.3: Detail of Figure 3.2, showing the control logic of the inner control loop of the Ardupilot.

3.7 Middle Loop

The middle control loop is run at approximately 30Hz. This loop is used to control the NavPitch and NavRoll angles in the inner loop and provide an input for throttle control. In this loop, airspeed is used through a PID controller to control the NavPitch, which in turn controls the elevator angle within the inner loop. The heading controlled by the outer loop is used to determine a bank angle which in turn controls the ailerons and rudder within the inner loop. Finally, airspeed and altitude (in the form of energy) are used to determine the throttle position.

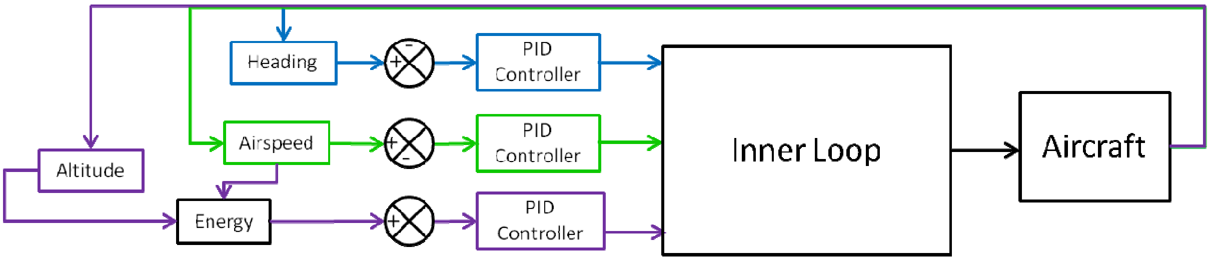


Figure 3.4: Detail of Figure 3.2, showing the control logic of the middle control loop of the Ardupilot.

The airspeed error is given by Eq. (3.11), which is then used in a PID controller to determine the NavPitch angle.

$$\epsilon_{Airspeed} = Airspeed - Set\ Airspeed \tag{3.11}$$

The throttle position is determined using specific energy error, given by Eq. (3.12), with the altitude error used in Eq. (3.12), given by:

$$\epsilon_{SpecificEnergy} = \epsilon_{Alt} g - \epsilon_{Airspeed}^2 \tag{3.12}$$

$$\epsilon_{Altitude} = Set\ Altitude - Altitude \tag{3.13}$$

The heading error is simply given as:

$$\epsilon_{Heading} = Set\ Heading - Heading \tag{3.14}$$

3.8 Outer Loop

The outer loop simply consists of the calculation of the heading to fly waypoints. The equation used to find the bearing between two waypoints is, from [42], given as:

$$\begin{aligned} Set\ Heading = \ & atan2(sin(\Theta_{WP} - \Theta_{\mathrm{Current}})cos(\Psi_{\mathrm{WP}}), cos(\Psi_{\mathrm{Current}})sin(\Psi_{\mathrm{WP}}) \\ & - sin(\Psi_{\mathrm{Current}})cos(\Psi_{\mathrm{WP}})cos(\Phi_{\mathrm{WP}} - \Phi_{\mathrm{Current}})) \end{aligned} \tag{3.15}$$

This forms the last part of the simplified autopilot code. There are, however, many nuances and subtle logic implementations. These have not been included here, but the full autopilot code can be examined in the "Autopilot Class" of the simulation program, presented in Appendix C.

After a brief trial and error tuning process, a satisfactory set of PID gains were determined for the autopilot This process was conducted due to the difficulties associated with tuning cascading PID controllers. The process followed was as suggested by the developers of the Autopilot code, and involved tuning the Inner loop PID controllers, then tuning the middle loop controllers and finally the outer loop controllers in turn. A set point was chosen for each PID controller and the response to a disturbance observed. The gains were then adjusted until the response was deemed satisfactory. The gains as determined through this process and used throughout the remainder of the simulations are presented in Table 3.1.

Table 3.1: PID gain settings used in all control simulations

PID controller	Proportional Gain	Integral gain	Derivative Gain
Pitch PID	0.8	0.3	0.1
Roll PID	0.6	0.2	0.1
Yaw PID	0.0	0.0	0.0
Airspeed PID	0.1	0.1	0.15
Energy PID	0.03	0.005	0.01
Heading PID	1.0	0.05	0.0

4 Chapter 4 – Development of Adaption Algorithm

4.1 Chapter Outline

This chapter forms the backbone of the design of an intelligent, fault-tolerant control system for an UAS. As such, the chapter will outline the proposed control strategy and the reasoning behind the decisions made in the design of the fault-tolerant control system. This will be followed by the development of the fuzzy logic adaption algorithm with the membership functions, rule base, inference engine and defuzzification process reviewed and the implementation thereof detailed in full. This is followed by the development of the allocation algorithm with attention paid to the practicalities of this system. Verification of the allocation algorithm's function will be performed.

4.2 Proposed Control Logic

The strategy to be developed is a two pronged approach that includes the use of a model reference fuzzy logic adaption algorithm as well as a control allocation algorithm. Figure 4.1 shows the proposed strategy and how the strategy relates to the existing autopilot structure. The model reference fuzzy logic adaption algorithm is the primary focus of this research and allows the autopilot system to become “Intelligent”. As outlined in Chapter 3, the autopilot makes use of a number of PID controllers to control the aircraft. Each of these controllers (pitch, roll, airspeed, altitude and heading) will be assigned to a fuzzy logic adaption algorithm, as indicated in Figure 4.1, with each module tracking a specified ideal model. The adaption algorithm will then adjust the gains of each PID controller based on the feedback from the aircraft, to ensure that the response of the aircraft matches the defined ideal model. It was hypothesised that this will create a system that is tolerant to changes in aircraft behaviour due to numerous failures or disturbances. It is intended that the adaption algorithm will be able to maintain the control performance of the aircraft for the tested failures.

The development of a control allocation algorithm stemmed from some of the preliminary simulations conducted which showed that, under certain circumstances, the aircraft was incapable of meeting the desired flight path as the controls had reached their physical limits. In these cases, there was insufficient control authority remaining to control the aircraft adequately and, had an adaption algorithm been implemented in isolation, no positive difference would have been made. It is possible that, in some cases, an adaption algorithm may induce instabilities into the system rather than provide fault-tolerance. Thus, it is the opinion of the author that some form of control allocation is vital in any fault-tolerant system. The chosen control allocation algorithm is the Daisy Chaining method which distributes the required control authority to multiple control effectors as required, as indicated in Figure 4.1. It is theorised that it will aid the fuzzy logic adaption algorithm in correct tracking and prevent instabilities arising due to insufficient control authority.

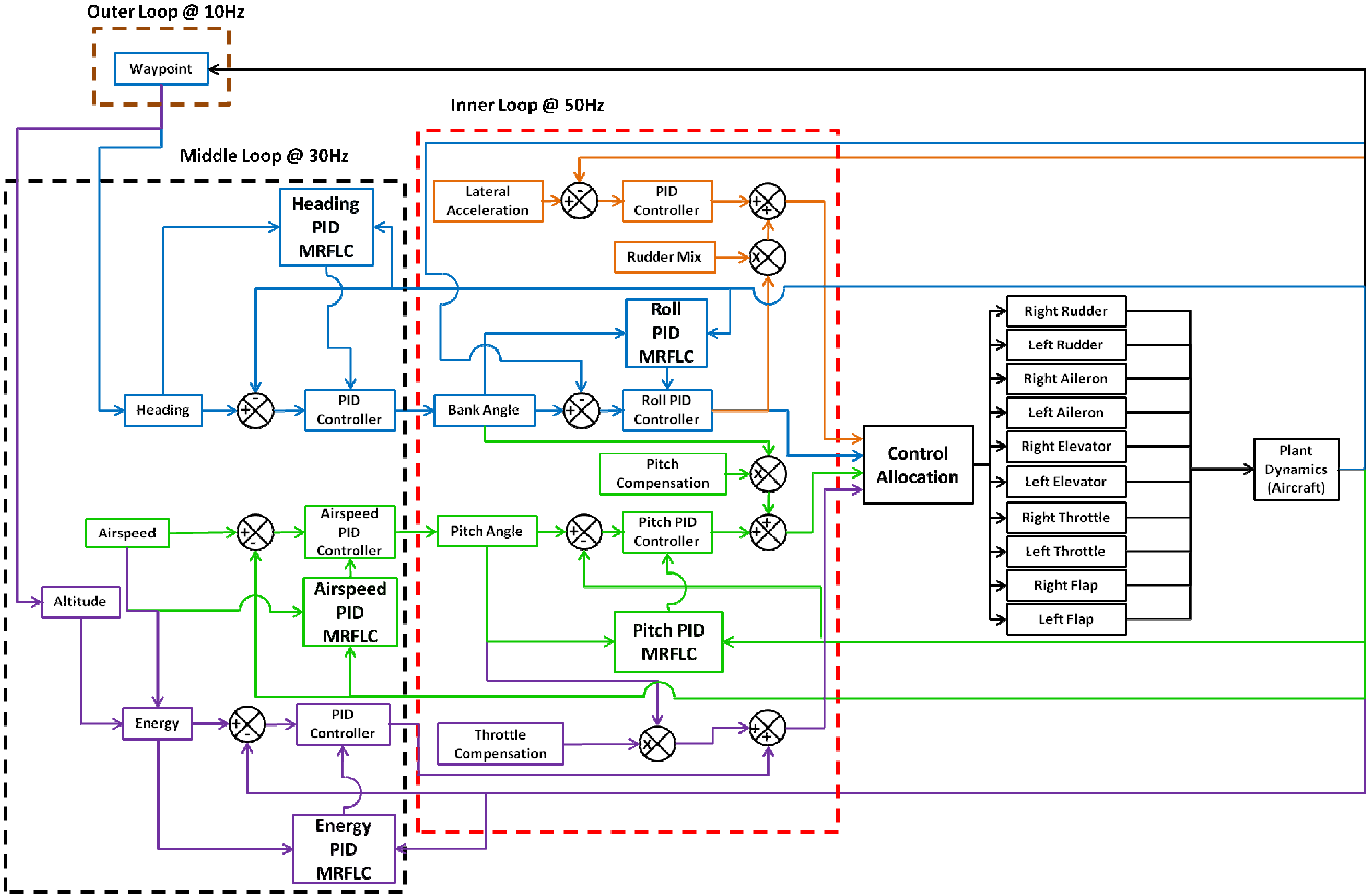


Figure 4.1: Block diagram showing the proposed control logic to be developed in the design of an intelligent fault-tolerant UAS

4.3 Fuzzy Logic Adaption Algorithm

Fuzzy logic, developed by Lotfi Zadeh [8], essentially creates a form of artificial intelligence by allowing a computer to determine the extent to which a condition is true. Based on this condition, an output can be given. Fuzzy logic makes use of five different processes that were discussed in Section 0, but are summarised for convenience below. These processes are as follows.

1. Pre-processing
2. Fuzzification
3. Rule Base
4. Inference Engine
5. Defuzzification

Typically, fuzzy logic is used to describe a data set or to control a system directly. In this case, fuzzy logic will be used to adjust the values of the gains of each PID control loop of the autopilot.

4.3.1 Ideal Model Development

It was decided to make use of a form of model reference control to determine the desired behaviour of the system. This required a simple but effective method to determine the change in a controlled parameter that is a realistic response to a command. An ideal response was generated that is loosely based on a first-order system. To achieve this, a simple model that is proportional to the error between the command and current measured value of the controlled variable was developed. This ideal model is given by (4.1). This determines the next value of the commanded variable based on the previous state of the variable, the size of the error between the ideal and command and the step size.

$$Ideal_{i+1} = Ideal_i + \varepsilon\tau dt \tag{4.1}$$

where, τ is a constant that increases or decreases the response time of the ideal behaviour and ε is the difference between the ideal variable and the command, given by:

$$\varepsilon = Command - \ Ideal_i \tag{4.2}$$

The first-order behaviour described above was chosen as, typically, it is desirable to have little to no overshoot in flying operations while still maintaining a reasonably quick response. The selection of τ is critical to the systems performance and its selection results in a method to tune the adaption algorithm. Typical responses to a step function for various values of τ are shown in Figure 4.2. Intuitively, higher values of τ provide a faster response and are more appropriate for aircraft control when compared to lower values, as the ideal response becomes too slow to be effective.

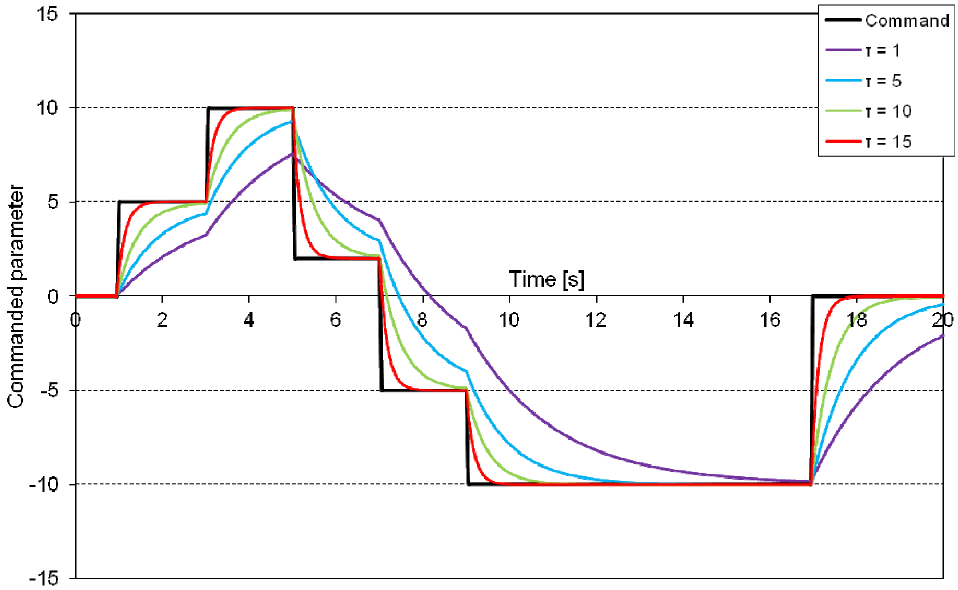


Figure 4.2: Typical ideal model response for various values of τ

With the establishment of the ideal behaviour, the determination of error and rate of change in error can be determined. It was required, however, to be able to distinguish an overshoot or undershoot from the ideal. This was determined using simple subtraction that varied depending on whether the command was greater than the ideal. Thus the model error was found using:

$$Model\ Error = \begin{Bmatrix} Command > Ideal & Ideal - Actual \\ Command < Ideal & Actual - Ideal \end{Bmatrix} \tag{4.3}$$

A similar technique was used to determine the rate of change in model error but it was desired to have the ability to recognise if the error was decreasing or increasing. The rate of change in model error is thus determined using:

$$Model\ \dot{Error} = \begin{Bmatrix} Model\ Error > 0 & \dfrac{Previous\ Error - Current\ Error}{dt} \\ Model\ Error < 0 & \dfrac{Current\ Error - Previous\ Error}{dt} \end{Bmatrix} \tag{4.4}$$

The use of conditional equations for error and rate of change in error allow for the use of a consistent set of rules that is not influenced by the direction of the command.

4.3.2 Fuzzy Membership Functions

Fuzzy logic is based on the ability to receive data, which is a concrete number, and transform this data into fuzzy data. In order to do this, membership functions are required. The development of these membership functions allow the control algorithm to examine the input and make a decision as to in which category the input falls. The size and shape of the membership functions, along with the rule base, determine the behaviour of the fuzzy system.

For the PID adaption fuzzy controller, two inputs are required, namely, the error and the rate of change in error. This in essence creates a Proportional – Differential fuzzy controller, with the ideal model being the set point to which the fuzzy control tracks. Two sets of membership functions are thus required in order to describe the error and the rate of change in error. The membership function can be any shape desired by the designer; however, commonly in control applications, the membership functions are trapezoidal, with triangular membership functions being a special case. As a starting point, it was decided to have the membership functions as shown in Figure 4.3 and Figure 4.4 . The triangular nature was initially chosen for simplicity. In fact, the values were found to produce satisfactory results after a brief trial and error period.

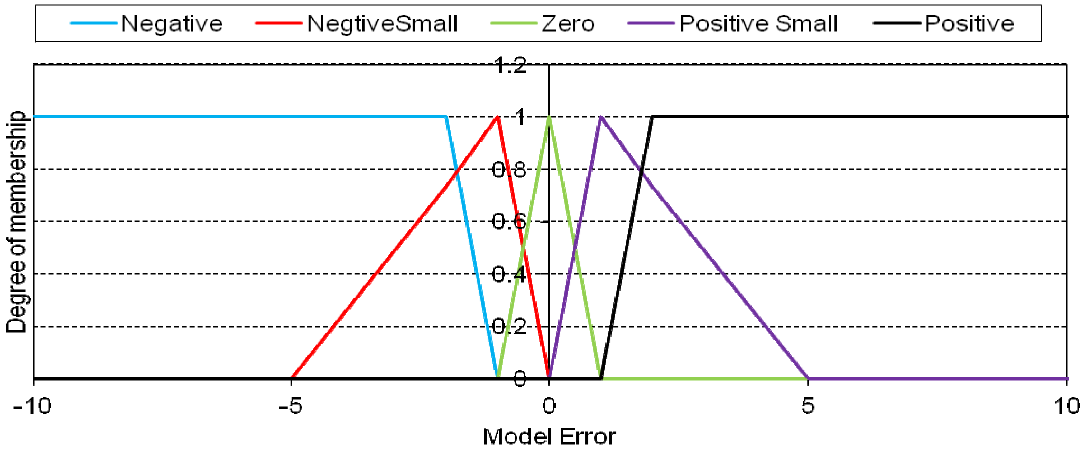


Figure 4.3: Membership functions for model error for the model reference adaptive controller

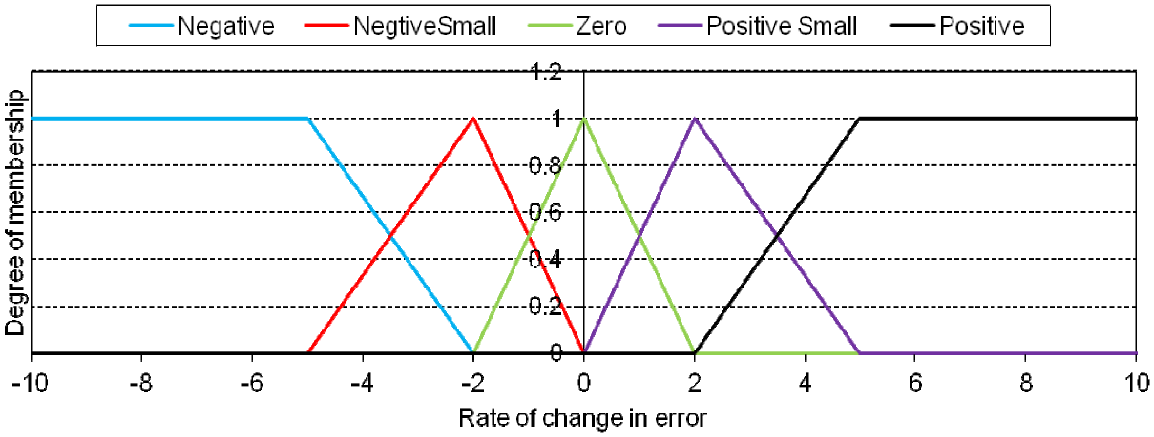


Figure 4.4: Membership functions for rate of change in model error for the model reference adaptive controller

4.3.3 Fuzzification

Fuzzification is the process of converting a fixed value to a degree of fulfilment of the membership functions. Essentially this process is a simple linear interpolation between the peaks of each membership function; however, careful consideration, to ensure an efficient and speedy computation, is required. This is particularly true for autopilots which may be limited by computational power.

In the practical implementation of the fuzzification process, an array was used to define the three or four points of each membership function. A set of simple 'If-Then' conditional statements were then used to determine the value of the degree of fulfilment of that membership function. To provide an example of the implementation of the fuzzification process, an arbitrary trapezoidal membership function is shown in Figure 4.5.

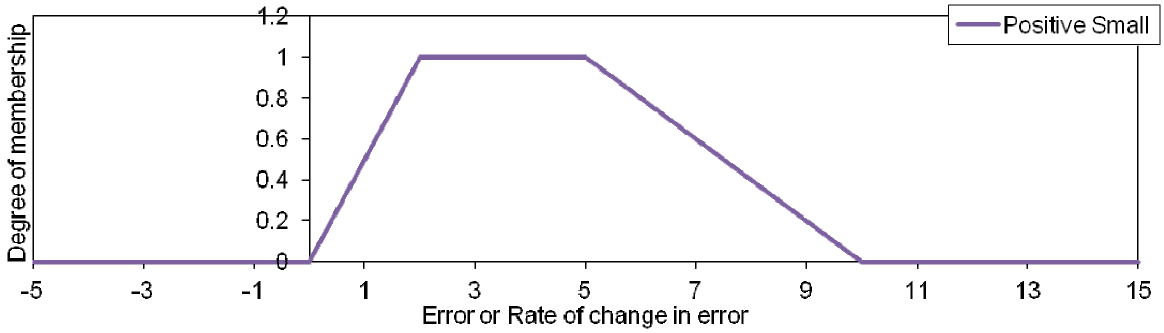


Figure 4.5: Positive small membership function used in example

From Figure 4.5, the array, given as Eq.(4.5), can be generated. Using this array and the logic outlined in Figure 4.6, a decision is made as to whether the input to the fuzzification process falls within the definition of a membership function. If the value is found to fall outside of the membership function, then a decision is made as to whether the membership function is a shoulder function or not, in which case the degree of membership receives a value of 1 or 0. If the value falls inside the membership function, a linear interpolation between the appropriate array entries is performed using Eq. (4.6). For further details, please refer to the "Fuzzify" function located in the fuzzy adaption class of the simulation code. This can be found in Appendix C.

$$Postive\ Small = [0.0 \quad 2.0 \quad 2.0 \quad 5.0] \tag{4.5}$$

$$DOM = \frac{Input - Membership\ Array[n]}{Membership\ Array[n-1] - Membership\ Array[n]} \tag{4.6}$$

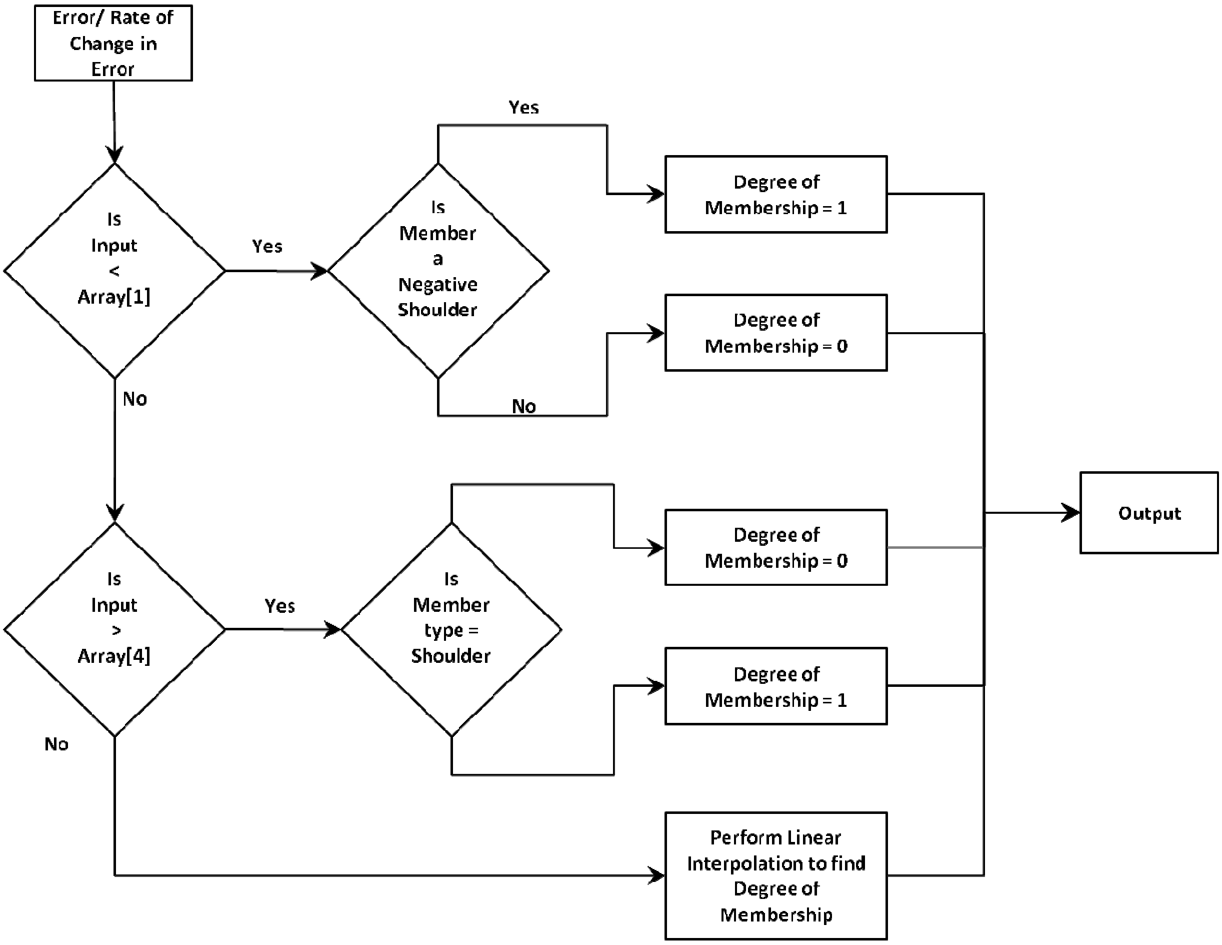


Figure 4.6: Logic used in the fuzzification of the error and rate of change in error

4.3.4 Fuzzy Rules

The fuzzy rule set is the defining feature of fuzzy logic. The rule set is where the designers' knowledge is conveyed to the system in the form of 'If-Then' rules. Because two parameters are used to determine the state of the system, the most convenient method of conveying the rule set is through the use of a *nxn* matrix, where *n* represents the number of membership functions. In the case of this research project, there are five membership functions relating to an error and rate of change of error making 25 combinations of membership function that needs to be accounted for. However, because the fuzzy logic controller must control the proportional, integral and derivative gains of a PID controller, 75 rules are required in total. To develop the rule base, a review of the behaviour of PID controllers was examined and the characteristics of each element reviewed. Based on this, a logical approach was taken.

Proportional Gain

The proportional controller is arguably the most important in a PID controller as it initiates primary corrective behaviour of the system after a disturbance or set point change. In order to establish the correct rules to follow, a series of step inputs was considered. When the proportional gain is too low, the response is typically slow with little to no overshoot of the controlled parameter. When the proportional gain is high, the rise time is reduced; however, there is generally an overshoot of the set point, often resulting in a damped harmonic type motion. If the proportional gain is increased further, the damping of this harmonic motion reduces and eventually becomes undamped, eventually resulting in a dynamically unstable condition. The fuzzy logic adaption rule base needs to consider the following features of a typical response to a step input:

- Undershooting behaviour
- Overshooting behaviour
- Steady-state error

These features will be identified by examining the direction of the input to the system and the resulting error from the ideal model. If the response is slower than the ideal model, this represents an undershoot condition. If the response is greater than the ideal model then this represents an overshoot condition. If the error is non-zero and the rate of change of error is zero then there is a steady-state error.

The rule base was formulated with the aid of Figure 4.7. In the figure, a series of step functions (red line) have been used to generate a typical response. The black line indicates the response of a typical system, while the blue line indicates the ideal model to be followed. At the first step function, the ideal model responds faster to the step change than the real model, indicating an undershoot condition.

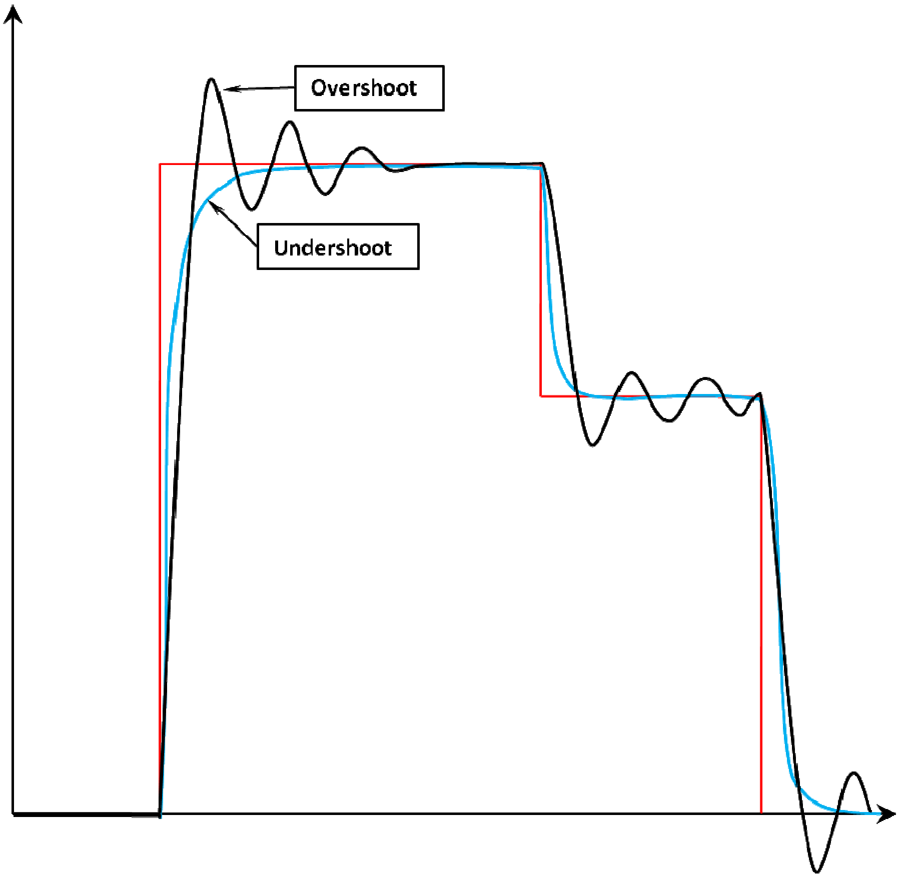


Figure 4.7: Diagram indicating the typical response of a system to a step input

For this portion of the response, it would be desirable to increase the proportional gain to track the ideal model more closely. However, at a certain point, the actual response matches the ideal model and, in the simplest sense, the proportional gain is at its ideal value. Following this point, the actual response exceeds the ideal and goes into an overshoot condition suggesting that the proportional gain should be reduced in an attempt to track the ideal response once more. This provides the basis from which to start generating the rule base; however, this would be likely to yield an oscillatory effect around the ideal model response. Initially, the proportional gain would increase to meet the ideal response but this would be likely to lead to an overshoot condition and a corresponding reduction in proportional gain. This would continue as no consideration is given to how quickly the actual response is reaching the ideal response. To improve the rule base, the rate of change of error is required. If the rate of change in error is decreasing, then the proportional gain could be kept constant, increase only slightly, or even decrease depending on the magnitude of the rate of change in error. If the rate of change in error is increasing, then depending on the condition of the error, the proportional gain may need to be increased or decreased. Following this logic, the full set of rules for the adaption of the proportional gain was developed and is presented in Table 4.1.

Table 4.1: Table indicating the rule base for the proportional gain constant

		Error				
		N	**NS**	**Z**	**PS**	**P**
Rate of Change of Error	**N**	No Change	Increase Slightly	Decrease Fractionally	Decrease slightly	Decrease Fractionally
	NS	No Change	Increase Fractionally	No Change	Decrease Fractionally	No Change
	Z	Decrease Fractionally	No Change	No Change	No Change	Increase Fractionally
	PS	Decrease Slightly	Decrease Fractionally	No Change	Increase Fractionally	Increase Slightly
	P	Decrease	Decrease slightly	Increase Fractionally	No Change	Increase

Integral Gain

The integral gain behaves differently to the proportional gain. When the proportional gain is used in isolation, it will be observed that a steady-state error is present. This is inherent in any proportional controller as an error is required to initiate a command. The integral part of the PID controller examines the past history of the error through the use of the integral and thus provides a correcting output that reduces the steady-state error to zero over time. The higher the integral gain, the faster the steady-state error is reduced. However, if the integral gain is too high, the contribution to the output by the integral portion of the controller will be excessive and it is likely that there will be excessive overshoots. The rule base of the integral gain was developed with the aid of Figure 4.8 which shows the typical behaviour of a system without any integral gain: again, the red line indicates the commanded step input while the blue line indicates the ideal behaviour of the system. In contrast to the proportional gain adjustment, which tracks the ideal model, the integral rule base essentially tracks the original set point of the PID controller. The concept of ideal model tracking or set point tracking, for this ideal model, are equivalent as the steady state of the ideal model should equate to the final set point of the system.

It is clear in Figure 4.8 that a steady-state error exists which is confirmed by examining the rate of change in error. Thus the integral gain needs to be increased for the system. To determine when this steady-state point has occurred, the rate of change of error is examined. When the rate of change of error approaches zero the integral gain can be increased to force the system to reach the ideal model. The integral gain, however, should not remain too large as this may cause instability in the system and thus a method of gain reduction should be considered. However, this should be considered carefully as, once the ideal model has been tracked, the integral gain cannot be lowered as this will impact on the ability of the system to maintain the desired set point. The potential exists to decrease the integral gain at a very slow rate after the system has achieved the ideal model tracking, thus ensuring that the integral has time to build up, compensating for the decreasing gain. The integral gain should also be reduced during any overshoot period. Following this logic the full set of rules for the adaption of the integral gain was developed and is presented in Table 4.2.

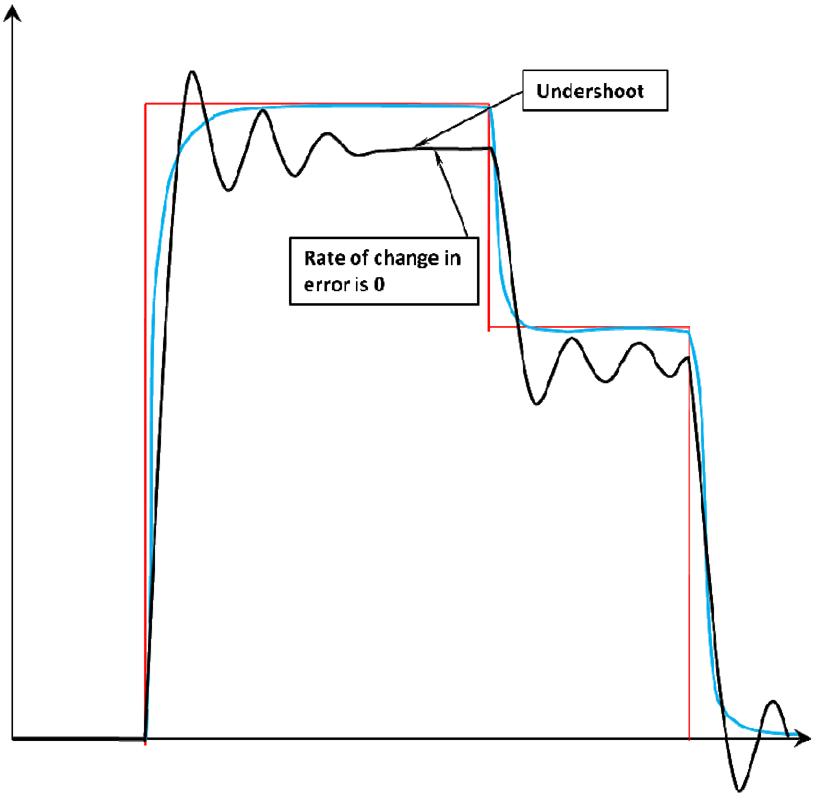


Figure 4.8: Diagram indicating the typical response of a system to a step input highlighting the need for integral gain

Table 4.2: Table indicating the rule base for the integral gain constant

		Error				
		N	**NS**	**Z**	**PS**	**P**
Rate of Change of Error	**N**	No Change	No Change	No Change	No Change	Increase Fractionally
	NS	No Change	No Change	No Change	Increase Fractionally	Increase Slightly
	Z	Decrease Fractionally	No Change	No Change	Increase Slightly	Increase
	PS	Decrease	Decrease Fractionally	No Change	Increase Fractionally	Increase Slightly
	P	Decrease	Decrease Slightly	No Change	No Change	Increase Fractionally

Derivative Gain

The derivative gain is used to prevent overshoot of the controlled parameter. Thus, unlike the integral gain which was modified when the rate of change in error was low, the derivative gains should only be adjusted when the rate of change in error is large. Thus when an overshoot condition is present and the rate of change in error is to increase this, the derivative gain should be increased to reduce this overshoot; conversely, when there is little rate of change in error, then no change should be made to the derivative gain. Should the derivative gains cause a lag in the ability of the system to reach steady state then the gains should be reduced. The rule base, as used in the simulation code, is presented in Table 4.3.

Table 4.3 Table indicating the rule base for the derivative gain constant

		Error				
		N	**NS**	**Z**	**PS**	**P**
Rate of Change of Error	**N**	Decrease Fractionally	Decrease Slightly	Increase	Increase Slightly	Increase Fractionally
	NS	No Change	Decrease Fractionally	Increase Slightly	Increase Fractionally	No Change
	Z	Increase Fractionally	No Change	No Change	No Change	Decrease Fractionally
	PS	Increase Slightly	Increase Fractionally	Increase Slightly	Decrease Fractionally	Decrease Slightly
	P	Increase	Increase Slightly	Increase	Decrease Slightly	Decrease

The details of the implementation of the rules, if they are desired, can be found in the "PID tune" function in the simulation code. This can be found in Appendix C.

4.3.5 Inference engine

The inference engine makes use of the AND operator. This operator was defined making use of a conditional function that examined the following term.

$$AND(a,b) = (a \geq b)$$

If the statement was true, then the output of the result would be b and, if the statement was false, the output would be a. This essentially created a function that determines the minimum of two numbers. In the proposed adaption algorithm, the results of the fuzzification process were used as inputs into the AND operator and combined with the rule base to determine the strength with which each rule fired. The details of this can be seen in the "PID tune" function of the simulation program, included in Appendix.

4.3.6 Defuzzification

The defuzzification process is concerned with converting the results from the inference engine and converting these results into a crisp number that can be applied to the PID gain adjustment. The first process was to define a set of weightings that could be used to describe the terms used in the rule sets numerically. These terms as well as the weightings used are presented in Table 4.4.

Table 4.4: Weighting used to defuzzify the results of the rule base

Change in PID gain	**Weighting**
Decrease (D)	-1
Decrease Slightly (DS)	-0.25
Decrease Fractionally (DF)	-0.15
No Change (NC)	0
Increase Fractionally (IF)	0.15
Increase Slightly (IS)	0.25
Increase (I)	1

This allowed the results of the rule base to be developed into a single crisp number for each gain using:

$$\Delta_{Gain} = (-1D + -0.25DS + -0.15DF + 0NC + 0.15IF + 0.25IS + 1I) \tag{4.7}$$

The adjustment of each gain was then performed using:

$$Gain = Gain + \Delta_{Gain}\epsilon m \tag{4.8}$$

where ϵ is an adaption rate used to tune the speed at which the adaption algorithm adjusts the gains and m is a modification factor that was used to scale the amount of change between the proportional, integral and derivative gains.

The factor m was required as it was desired to have a different change in gain change for the proportional, integral and derivative controllers. The modification factor used was found to yield satisfactory results after some trial and error and the values used are presented in Table 4.5

Table 4.5: Value of modification factor for the proportional, integral and derivative gains

Controller	**Modification factor**
Proportional gain	0.1
Integral Gain	0.01
Derivative Gain	0.001

The final process in the defuzzification was the constraint of the gains to preset maximum and minimum values.

These maxima and minima were found to yield satisfactory results without causing large instabilities within the system and are presented in Table 4.6.

Table 4.6: Maximum and minimum allowed gains for the proportional, integral and derivative controller gains

Controller	**Maximum gain**	**Minimum gain**
Proportional	10.0	0
Integral	2.0	0
Derivative	1.0	0

The main aspects of the fuzzy logic adaption system have been presented; however, the complete adaption algorithm can be found in the simulation code of Appendix C.

4.4 Control Adaption Algorithm Testing

The adaption algorithm was tested by providing a series of pitch angle step inputs. The control gains for the autopilot were set to zero as a starting point. Two different ideal model time constants were used to determine if the algorithm was indeed attempting to track the ideal model behaviour. Figure 4.9 shows the results of the low time constant case while the high adaption rate case can be seen in Figure 4.10. It can be seen by comparing the figures that the adaption algorithm is attempting to track the ideal model, albeit with some difficulties. The figures clearly show a difference in response between the two cases, with a slow response time observed in Figure 4.9 and a faster response time observed in Figure 4.10. The figures show that there are discrepancies between the ideal model and the actual pitch angles attained. This is likely caused by the aggressive nature of the manoeuvres and the fact that the PID gains were set to zero. It is unlikely that perfect tracking could be achieved immediately and it may take a number of manoeuvres to achieve better tracking. The results may also be improved somewhat by the tuning of the adaption algorithm adaption rates and modification factors, but this will be conducted during the simulation of more realistic flight conditions.

The change in the PID gains can be seen in Figure 4.11 and Figure 4.12. The figure shows that the proportional gain changes frequently and fairly aggressively, particularly for the high time constant case at the moment of a set point change. It was observed that the integral gain steadily increases in both cases which would be expected as the pitch angle typically showed a small steady-state error.

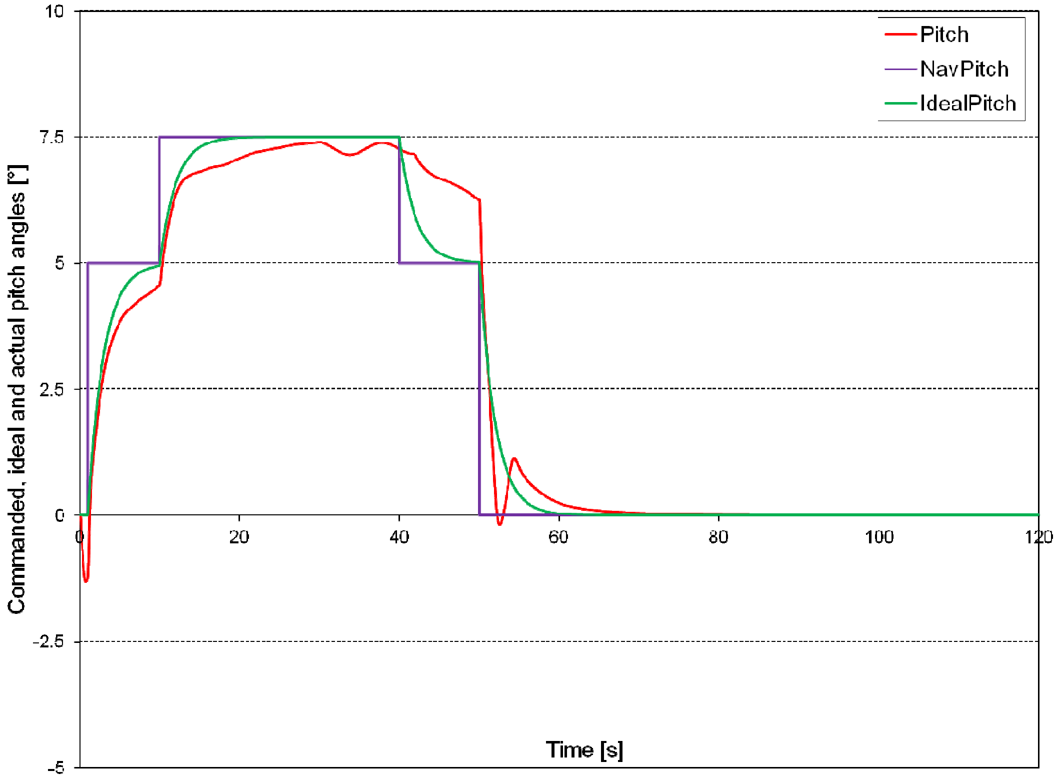


Figure 4.9: Simulated time history of the commanded step inputs, ideal pitch response and actual pitch response for a low time constant ideal model

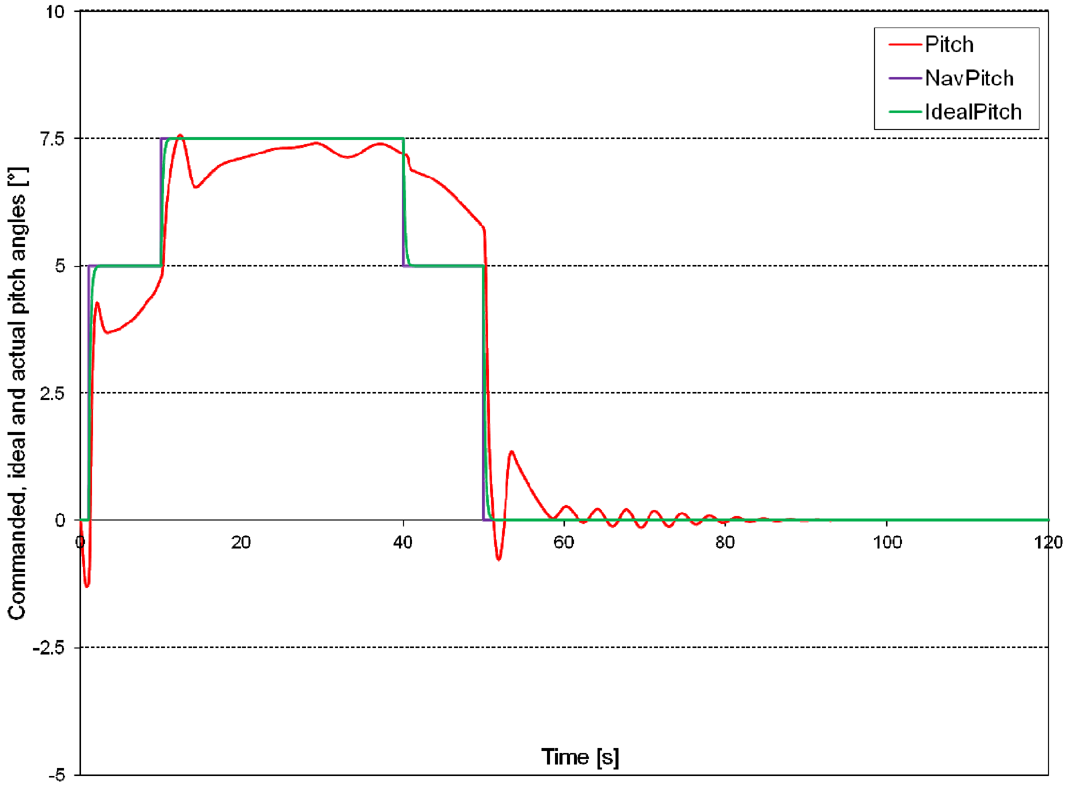


Figure 4.10: Simulated time history of the commanded step inputs, ideal pitch response and actual pitch response for a high time constant ideal model

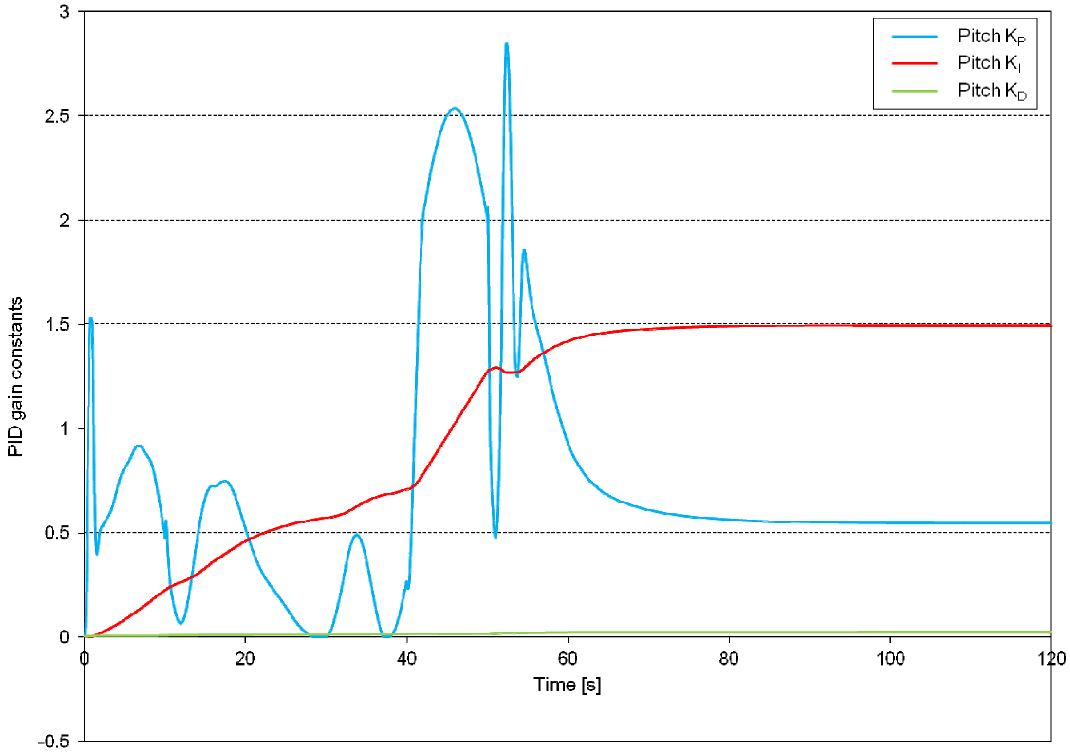


Figure 4.11: Simulated time history of the pitch PID gains for the low time constant ideal model

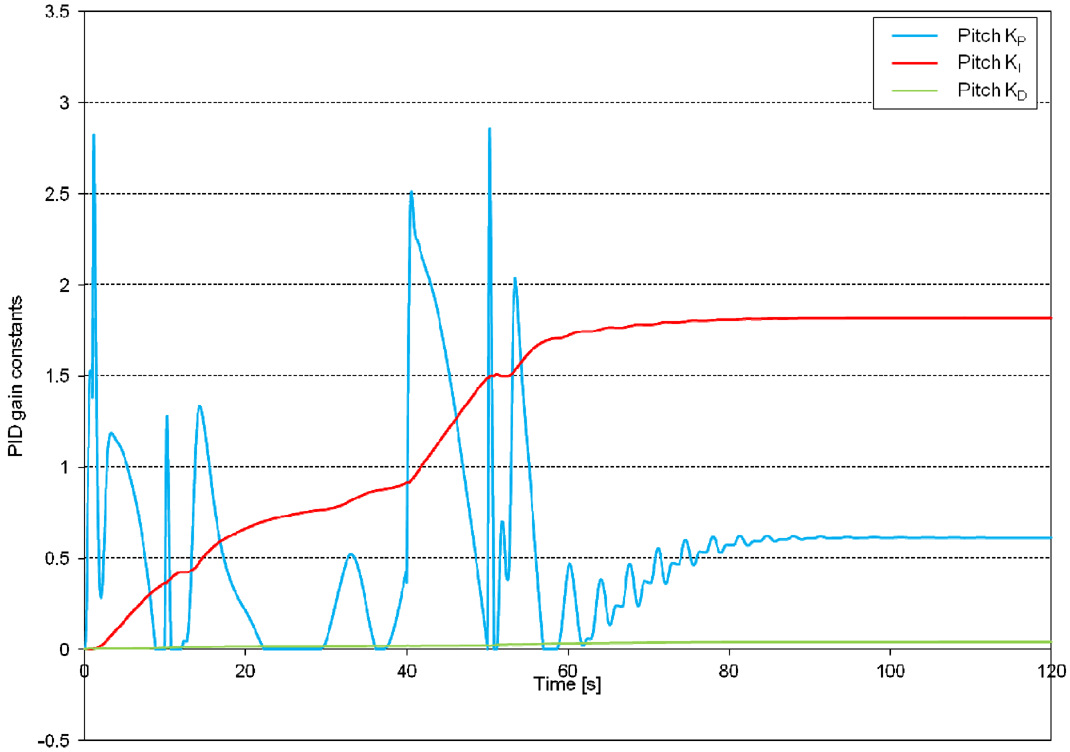


Figure 4.12: Simulated time history of the pitch PID gains for the high time constant ideal model

4.5 Control Allocation

One of the problems that is envisaged when a failure occurs is the saturation of a control surface in an attempt to stabilise the aircraft or even maintain a trimmed position for extreme control failures. If this saturation occurs, the ideal behaviour will not be able to be maintained, creating a number of potential problems some of which have been considered previously. A limit on the PID gain values was set, as mentioned in Chapter 4.3.6, to prevent the PID gains from exceeding a certain value and causing the instability of the control system. However, this potential for the adaption algorithm to cause unstable behaviour highlights the need to couple two control methodologies together. A method of control allocation is proposed that allows for the increase of PID gains to become more effective by transferring the control authority to other surfaces in the event of control saturation. This is an auxiliary feature and, as such, a simple non-optimal control allocation method will be selected as robustness and fast computation time rather than optimisation is required. The chosen control allocation technique to be implemented is the daisy chain control allocation method.

4.5.1 Daisy Chain Control Allocation Algorithm

The daisy chain control method is a simple method of distributing the remaining control effort required amongst different control surfaces. Each aircraft axis is assigned a primary, secondary and tertiary control surface. The primary control surface is deflected in the normal direction until the command exceeds its limit of travel. Once this point has been reached, the remaining control effort required is sent to the secondary control surface assigned for that control direction until it too saturates. Finally, any remaining control effort required is assigned to the tertiary control surface.

Two difficulties arise during the control allocation. The first is that the effectiveness of the secondary and tertiary control surfaces differs from that of the primary surface: typically, the effectiveness is lower. Thus to induce the same control command, a greater deflection will be required. This has the consequence that, if not properly tuned, the performance of the PID controller will degrade unless an adjustment is made. This is easily accomplished with the use of a control effectiveness matrix which essentially describes the effectiveness of a control surface in all axes. However, this can only be used effectively if all of the characteristics of the control surfaces can be determined, either through wind tunnel testing or empirical methods. If the aircraft was fully functional, this may prove to be a reasonable approach. However, in this research, it must be assumed that the aircraft is not fully functional. An assumption has to be made regarding the effectiveness of the control surface for any given direction. Ideally, this assumption should be conservative in nature and allow the PID controller to correct any discrepancy or, in the case with adaption, allow the adaption algorithm to adjust the PID gains to achieve the ideal behaviour.

The second difficulty results from use of the other control surfaces. It is conceivable that the use of the non-primary control surface will result in the degradation of control authority in the other control axes. For example, if differential elevator control is used to supplement roll to the extent that the pitch control authority is compromised, then the aircraft may become unable to maintain stable and controlled flight altogether. To solve this problem, two techniques were applied. The first was in the selection of secondary and tertiary control effectors. It was decided to limit the secondary control effectors to those not responsible for any primary control action, as far as was practical. Secondly, the status of each control surface was examined before applying the secondary and tertiary control inputs. The residual control authority was determined for each control surface after the primary deflection was applied and then used in the secondary and tertiary control allocations, thus giving the primary control direction priority over the secondary and tertiary control directions.

The control allocation assignments used in the implementation of the control allocation algorithm are presented in Table 4.7. The table shows that flaps are assigned to be secondary control effectors in the pitch and roll directions thus ensuring that the ailerons and elevators are not the secondary control effectors for any control direction. Throttle is used to supplement yaw control as the aircraft is a twin-engined aircraft with the motors offset from the centre line. The effectiveness weightings provided in Table 4.7 were determined partly through the use of simple estimates and then fine tuned by running multiple simulations and adjusting the values slightly to achieve a control performance similar to the performance with the primary effectors.

Table 4.7: Control allocation assignments and effectiveness weighting as used in the implementation of the control allocation algorithm

Control axis	Primary Effectors	Secondary Effectors	Secondary Effectiveness	Tertiary Effectors	Tertiary Effectiveness
Pitch	Elevator	Flaps in Unison	2.0	Ailerons in Unison	2.0
Roll	Ailerons	Differential Flaps	2.0	Differential Elevator	3.0
Yaw	Rudder	Differential Throttle	0.0555	N/A	N/A
Thrust	Throttle	N/A	N/A	N/A	N/A

The allocation algorithm is a multistep process, the logic of which is detailed in Figure 4.13 that begins with an input from the autopilot of the required elevator, aileron, rudder and throttle deflections. Based on the limits of control deflections set in the autopilot, the allocation algorithm first determines if any additional control effort is required for each of the primary effectors.

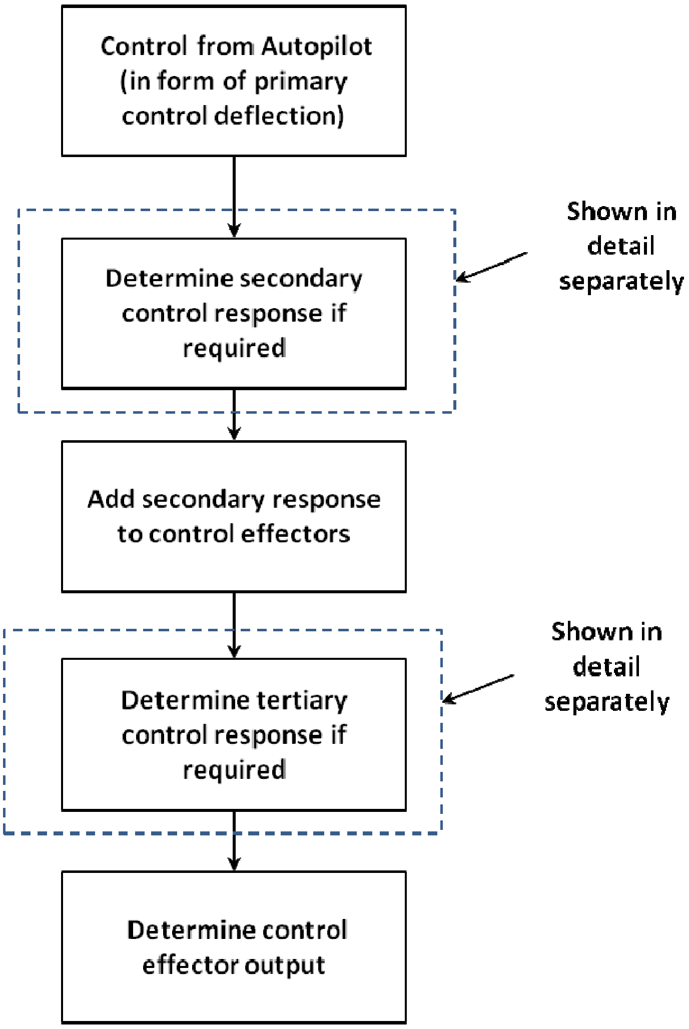

Control from Autopilot (in form of primary control deflection)
Determine secondary control response if required
Shown in detail separately
Add secondary response to control effectors
Determine tertiary control response if required
Shown in detail separately
Determine control effector output

Figure 4.13: Overall control allocation logic

This is simply achieved by comparing the maximum and minimum control deflections available with the commanded control deflections from the autopilot, a process which is shown in Figure 4.14. If additional control effort is required, the remaining control effort of the secondary control is determined. Figure 4.14 shows how this is then checked to determine whether the remaining control effort of the secondary control surface is sufficient to resolve the initial control effort required. If it is, then the control deflection is calculated using Eq. (4.9). Otherwise the remaining control effort is provided and the remaining control effort transferred to the tertiary control surface where a similar process, detailed in Figure 4.15, is performed.

$$\begin{aligned} &Secondary\ control\ defelction \\ &\quad = Additonal\ control\ effort\ required \\ &\quad \times Secondary\ control\ effectivenss \end{aligned} \tag{4.9}$$

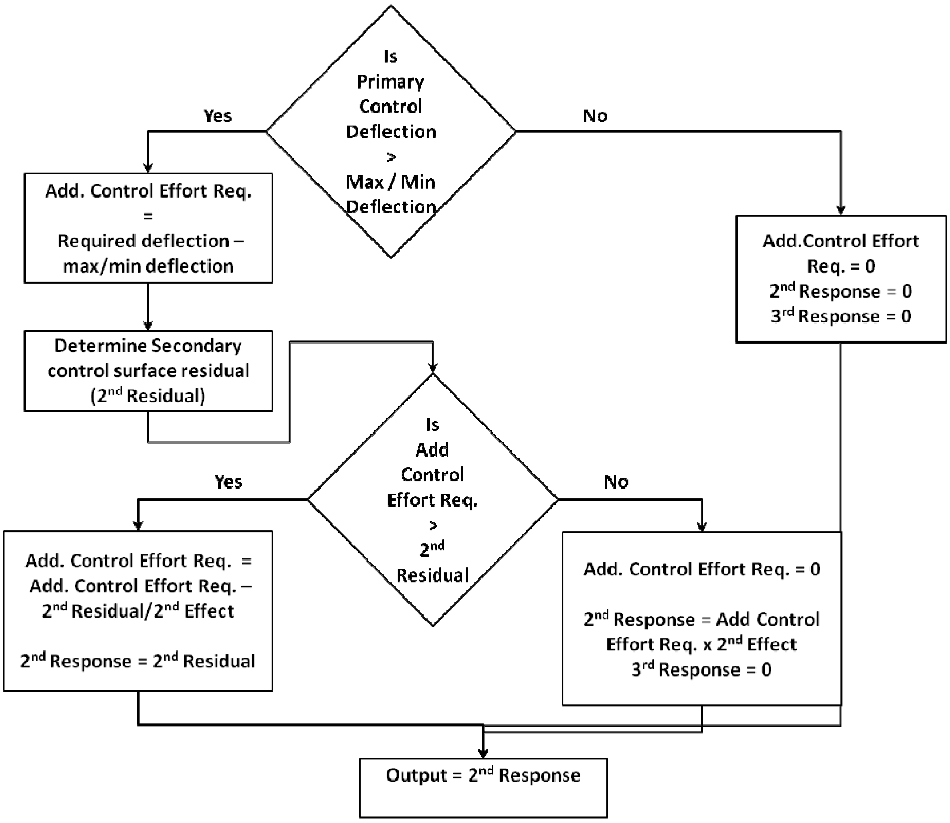

Is Primary Control Deflection > Max / Min Deflection
Yes
No
Add. Control Effort Req. = Required deflection – max/min deflection
Add.Control Effort Req. = 0
2nd Response = 0
3rd Response = 0
Determine Secondary control surface residual (2nd Residual)
Is Add Control Effort Req. > 2nd Residual
Yes
No
Add. Control Effort Req. = Add. Control Effort Req. – 2nd Residual/2nd Effect
2nd Response = 2nd Residual
Add. Control Effort Req. = 0
2nd Response = Add Control Effort Req. x 2nd Effect
3rd Response = 0
Output = 2nd Response

Figure 4.14: Block diagram showing the determination of secondary control effort required

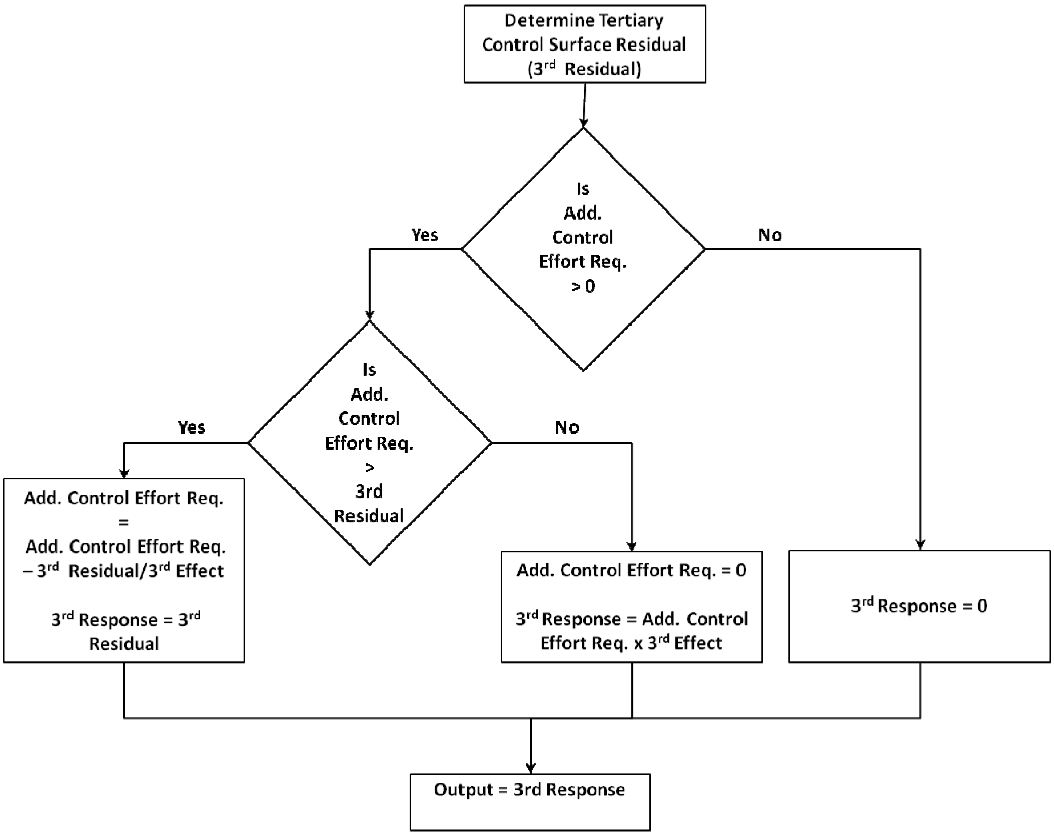

Determine Tertiary Control Surface Residual (3rd Residual)
Is Add. Control Effort Req. > 0
Yes
No
Is Add. Control Effort Req. > 3rd Residual
Yes
No
Add. Control Effort Req. = Add. Control Effort Req. – 3rd Residual/3rd Effect
3rd Response = 3rd Residual
Add. Control Effort Req. = 0
3rd Response = Add. Control Effort Req. x 3rd Effect
3rd Response = 0
Output = 3rd Response

Figure 4.15: Block diagram showing the determination of the tertiary control effort required

4.6 Control Allocation Algorithm Testing

The control allocation system was tested, by running the simulation with the autopilot in the "stabilise" flight mode. This flight mode attempts to maintain wings level with a pitch angle of 0°. This typically results in small deflections of elevator, aileron and rudder to achieve this trim position for a given throttle setting. Figure 4.16 (a) to (d) shows the required control deflections to maintain this flight condition and shows that indeed some elevator deflection, aileron deflection and rudder deflection are required. The rudder deflection stems from an aileron-rudder mix used in the autopilot. The allocation algorithms can be easily tested by reducing the primary control surface maximum and minimum deflections to 0°. The autopilot would be forced to make use of the secondary and tertiary surfaces. Should the tertiary actuators not be used, then the secondary actuator maximum and minimum limits can be reduced.

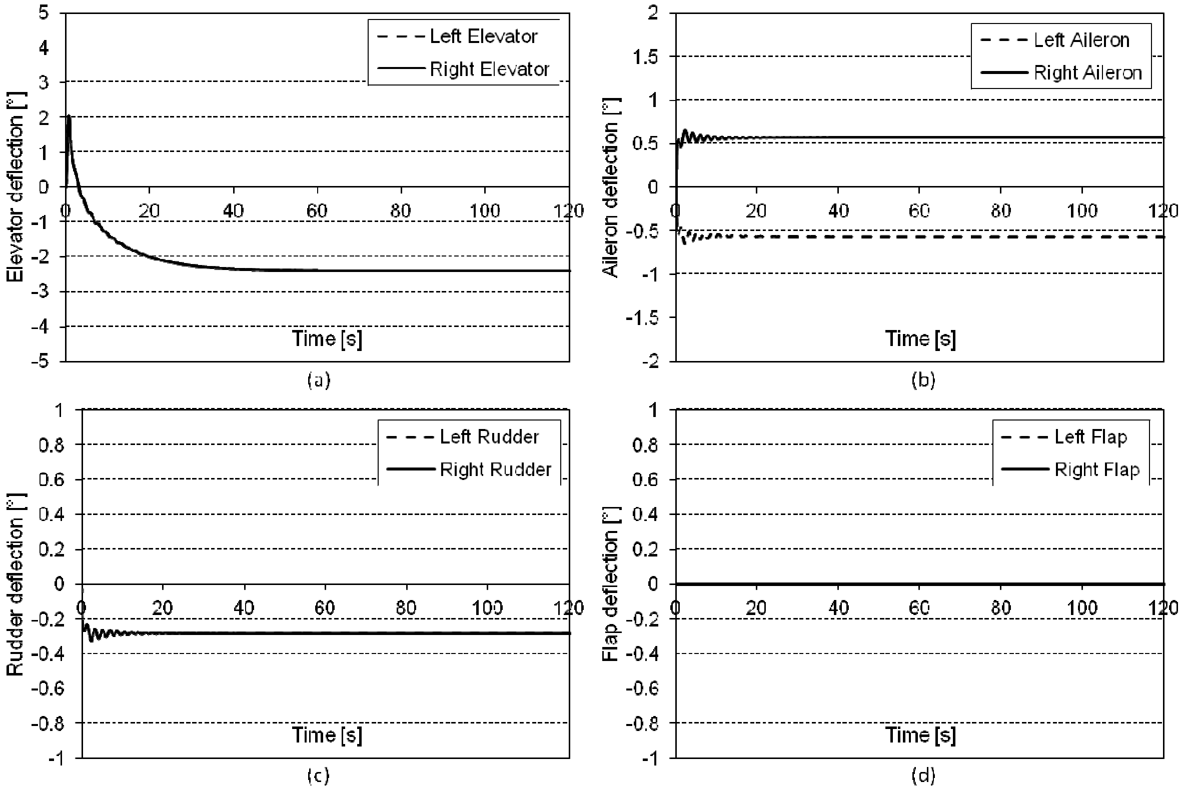


Figure 4.16: Simulated time history of elevator (a), aileron (b), rudder (c) and flap (d) deflections required to maintain level flight with default limitations on all control surfaces

The first control direction to be checked was pitch. The maximum and minimum elevator control deflections were set to 0°. Figure 4.17 (a) to (d) shows the required control defections and that there is a large flap deflection° (Figure 4.17 (d)) that occurs to generate a pitching moment to compensate for the lack of elevator. The flap authority in the pitch direction is relatively small when compared to the elevator authority and thus it can be seen that the ailerons are also required to achieve the desired pitch control effort. Of note is the offset in aileron deflection that is maintained throughout the simulation, indicating that roll control is not lost.

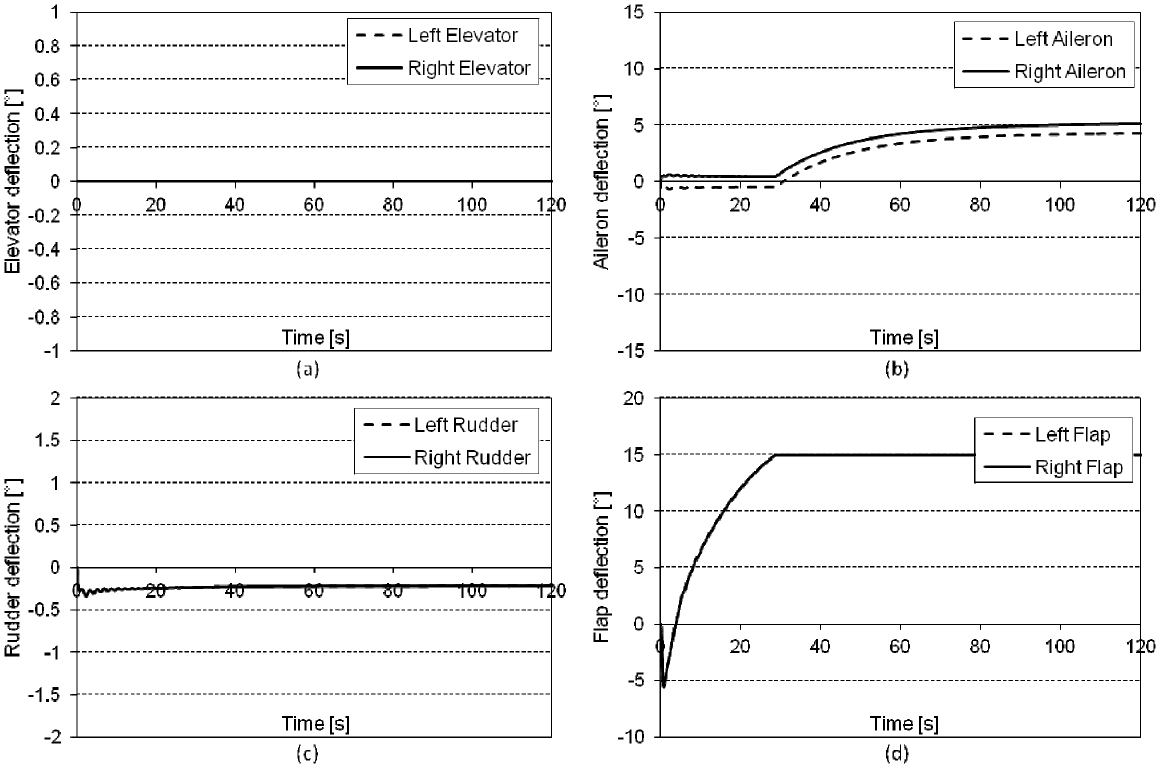


Figure 4.17: Simulated time history of the elevator (a), aileron (b), rudder (c) and flap (d) deflections required to maintain level flight with no elevator actuation

The simulation was re-run with maximum and minimum elevator deflections restored but now reducing the aileron deflections to 0°. Figure 4.18 (a) to 9d) shows the required deflection and that that without the ailerons, the flaps (Figure 4.18 (d)) are used to achieve the desired roll effort. It is noted that more flap deflection is required to achieve the same roll effort. This makes physical sense in that the flaps are inboard of the ailerons and hence have a reduced moment upon which to act. In order to check the tertiary control surface actuation in the roll direction, the simulation was run with aileron and flap maxima and minima set to 0°. This simulation, shown in Figure 4.19 (a) to (d), shows that the elevators (Figure 4.19 (a)) are now used differentially to generate the required roll effort. It is also noted that there is an offset about which the elevators are deflected: that is, they are not deflected equally from the neutral position. This is required to generate the required pitching moment to maintain 0° pitch angle.

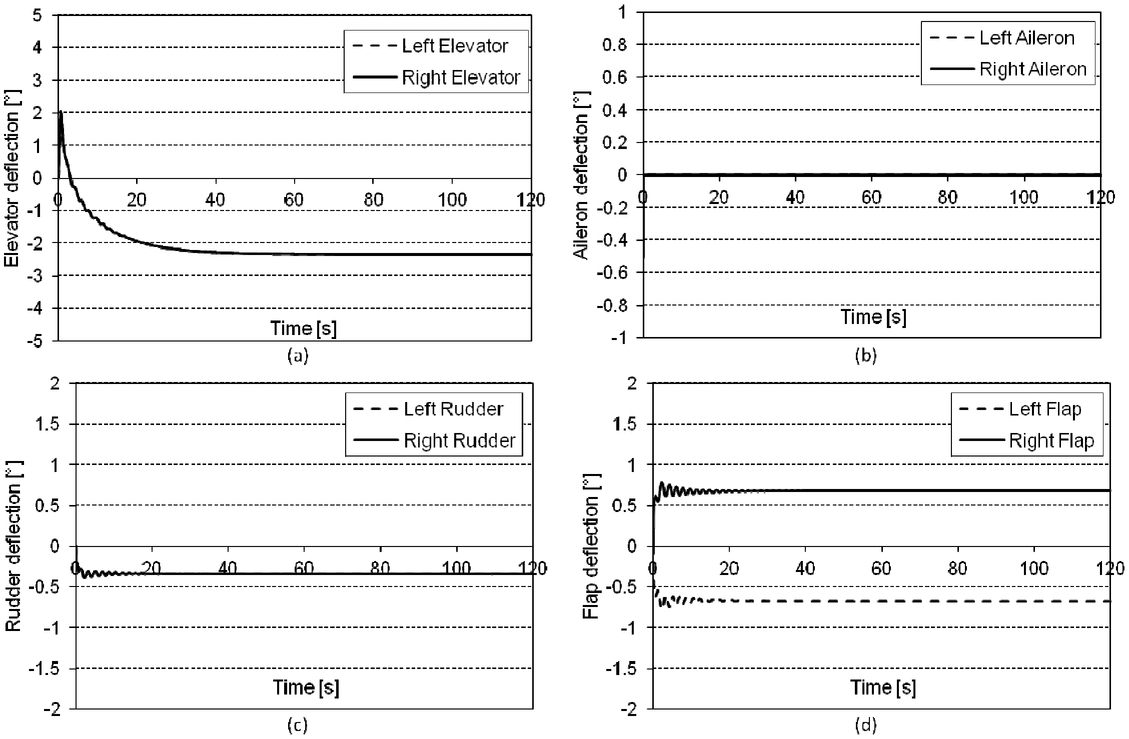


Figure 4.18: Simulated time history of the elevator (a), aileron (b), rudder (c) and flap (d) deflections required to maintain level flight with no aileron actuation

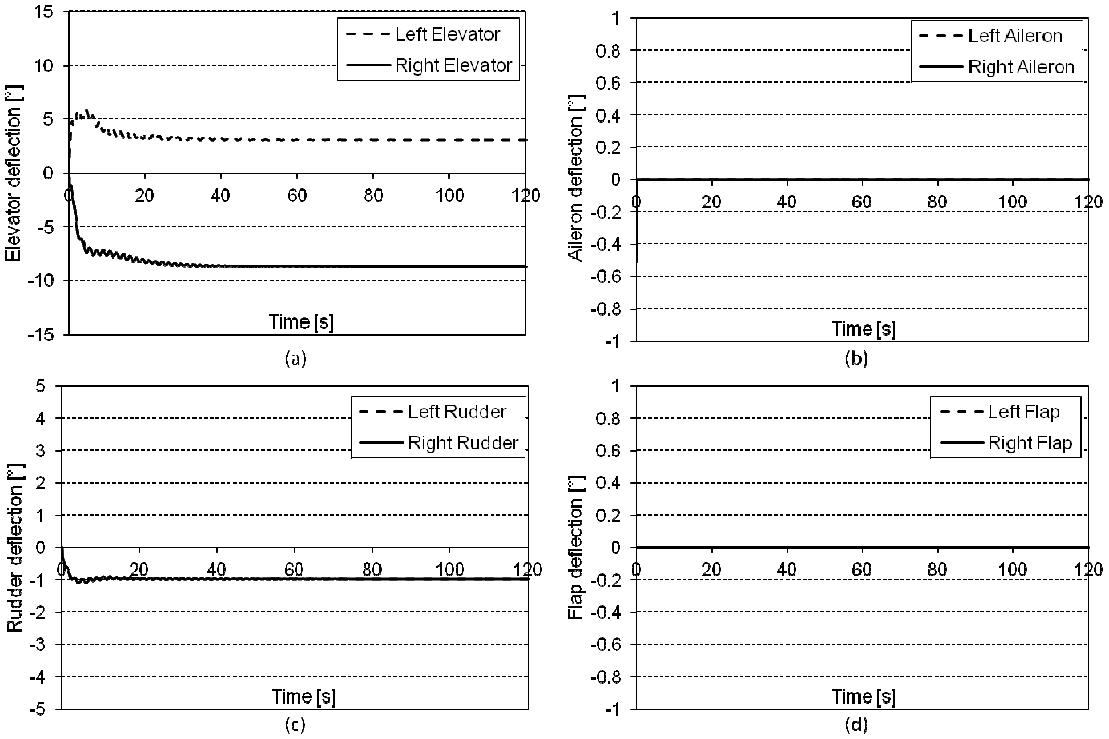


Figure 4.19: Simulated time history of the elevator (a), aileron (b), rudder (c) and flap (d) deflections required to maintain level flight with no aileron or flap actuation

Finally, the yaw control direction was tested. The maximum and minimum rudder values were set to 0° after having restored the aileron and flap maxima and minima. Figure 4.20 shows the offset in throttle positions induced in order to achieve the required yaw effort.

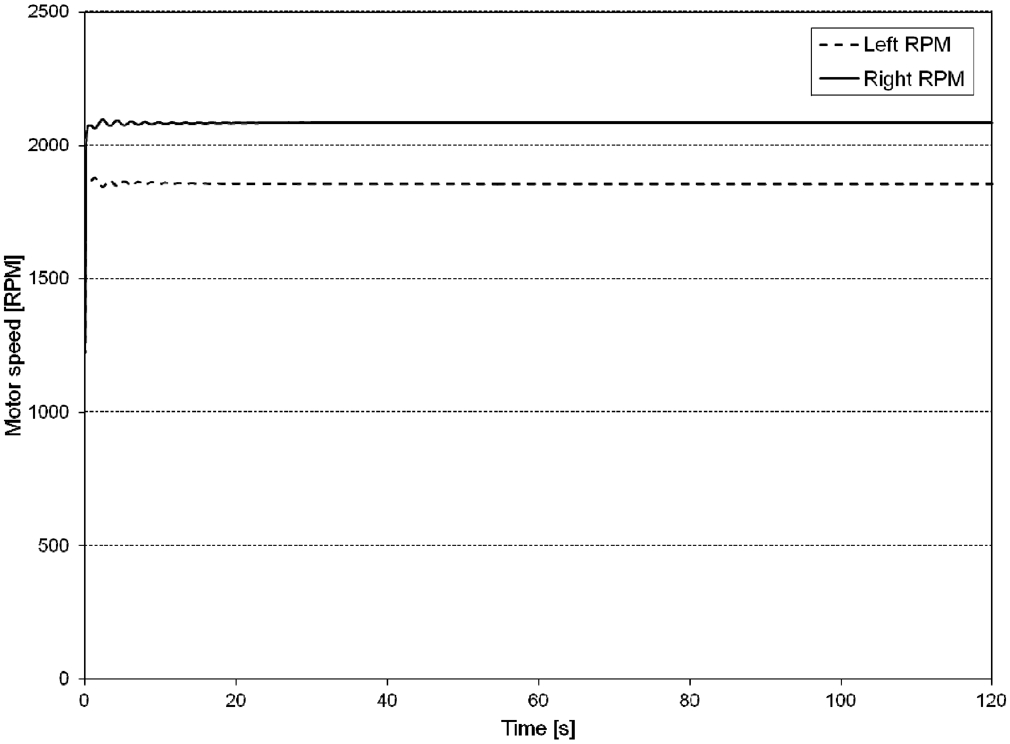


Figure 4.20: Simulated time history of the motor response required to maintain level flight with no aileron or flap actuation

In all of the above cases, the desired set point values of bank angle and pitch angle were achieved. While not included here, the plots of pitch angle and bank angle can be found in Appendix C. In all of the above results, it can be seen that the secondary and tertiary control actuators are working in the correct sense and when required and thus it can be concluded that the allocation algorithm is functioning as intended.

.

5 Chapter 5 – Simulation

5.1 Chapter Outline

This chapter begins with the verification of the flight dynamic model through the use of a linearised static flight model as well as through the use of well defined and understood performance indicators. This is only a verification that the simulation code is running correctly and only serves to confirm that results obtained are a realistic reflection of the real aircraft. After this verification, a report of the failures modelled and the testing procedure followed will be outlined. This chapter's primary concern is to test the adaption and allocation algorithms under a variety of flight conditions and with a variety of aircraft failures to determine if these algorithms are tolerant to certain control actuator failures. The results for the inner loop simulations, middle loop simulations and, finally, the outer loop simulations of the autopilot are presented.

5.2 Flight Dynamic Model Verification

The flight dynamic model makes use of the fundamental equations of motion of a rigid body about all axes. The forces are resolved using a coefficient build up method which when used in the equations of motion allow the accelerations of the aircraft about all three axes to be determined. The flight model can be verified, in the sense that key performance indicators can be checked, to determine that the behaviour of the aircraft is reasonable and that all integrators and estimators are functioning correctly.

Firstly, the longitudinal trim position was verified by running the nonlinear flight dynamic code with the autopilot in the "Stabilise" flight mode, equivalent to the methods outlined in [45]. This flight mode would maintain level flight and allow the aircraft to be trimmed. Once this trim position was established this could be checked against a well established static trim calculation. The performance of the aircraft with respect to lift and drag ratios, elevator angles, and angles of attack to maintain level flight can then be compared.

A form of lateral verification can also be conducted. This was achieved by running the simulation with the autopilot in the "FBW A" flight mode and inducing a constant banked turn. Using simple turn radius indicators and by examining the ground track, the performance of the nonlinear flight dynamic code can be verified.

It must be stressed that this does not validate that the model will behave exactly as the real aircraft as this can only be done after flight tests. But it does provide a means to determine whether the simulation results are reasonable.

5.2.1 Longitudinal Verification

A three degree of freedom nonlinear trim equation was developed with the aid of [37]. This code essentially determines the longitudinal trim position for an aircraft given linearised approximations of various aerodynamic properties. The code, described in more detail in [37] and found in Appendix C, examines the lift, drag and thrust forces acting on the centre of gravity and simultaneously solves the three nonlinear equations of motion for the aircraft to find the trim condition. The linear model as well as the nonlinear model was run at the same altitude and speed. The elevator angle to trim and resulting angle of attack were used as the primary focus of comparison and are presented in Table 5.1.

Table 5.1: Table comparing the results for longitudinal trim of the simple linear model and the nonlinear flight dynamic model

Performance indicator	Trim calculation @ 25m/s		Simulation		Percentage Difference (%)
Velocity	25.000	m/s	24.996	m/s	-0.016%
Elevator angle	-0.896	°	-1.214	°	1.39% [1]
Angle of Attack	1.166	°	1.509	°	2.28%[2]
Pitch angle	1.166	°	1.489	°	2.28%[3]
Required Thrust	17.65	N	14.16	N	19%

It was observed that there were some differences, which is understandable based on some of the assumptions made in the simplified model. In particular the large variation in the required thrust. This is largely attributed to the use of the parabolic drag curve in the simplified trim model. Upon examination of the drag coefficient at this angle of attack it was found that an error of 26.2% was present between the parabolic drag curve and the wind tunnel data. The small differences in elevator angel to trim and angle of attack errors can be attributed to similar assumptions made in the trim analysis. The results show that the nonlinear simulations are similar to the trim analysis and that the simulation code is functioning correctly. It is noted however, that this is not a validation of the simulation code, simply verification of correct functioning. A true validation would require the comparison of actual flight data, which is not yet available.

5.2.2 Lateral Verification

The lateral verification of the flight dynamic model is more difficult than the simple trim equation. The verification of the dynamic response in particular is very difficult. However, the general behaviour can be checked by commanding the aircraft into a set bank angle and observing the track

[1] Based on full elevator deflection of 30°

[2] Based on stall angle of 15°

[3] Based on stall angle of 15°

followed by the aircraft. First the track should be circular and, making use of the following equation, the radius of the circle should be checked.

$$R = \frac{V_{\infty}^{2}}{g\sqrt{n^{2}-1}} \tag{5.1}$$

where, $$n = \cos(\phi) \tag{5.2}$$

The simulation was run in FBW A flight mode, which adjusts the elevator and ailerons to maintain a constant pitch and bank angle. With some experimentation, it was found that, to maintain a bank angle of 30° at an altitude of 1524m at 25 m/s, a pitch angle of 2.3° and a throttle setting of approximately 38% was required to maintain a constant turn at constant altitude. Using Eq. (5.1) and (5.2), a turn radius of approximately 110m is expected as shown in Table 5.2. The nonlinear flight dynamic simulation resulted in a slight difference in turn radius of 107m, an error of approximately 2.9%. The resulting flight path was found to be circular as expected with a variation during the initial phases of the turn, as would be expected as the aircraft started from level flight seen in Figure 5.1.

Table 5.2: Table comparing the results for the lateral verification of the nonlinear flight dynamic model

Turn Conditions				
Flight speed	V_{WA}	25.000	m/s	
Bank angle	φ	30.000	°	
Load factor	n	1.155	g	
Results				
	Prediction		**Simulation**	
R	110.350	m	107.140	m
D	220.700	m	214.280	m

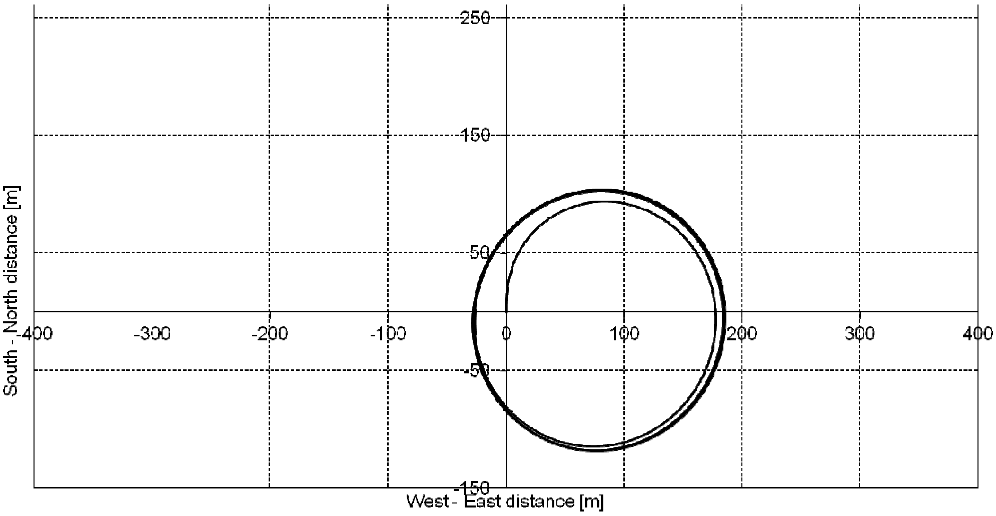


Figure 5.1: Ground track of aircraft with a constant bank angle of 30° set

5.3 Faults Scenarios Considered

The failures modelled stem from the objectives of this research as described in Chapter 1.7. These failures are considered representative of typical UAS failures that one may expect in the operation of a research UAS. Severe structural damage was not considered but, with some modifications to the flight dynamic code, this could be achieved and, as such, will be included in the recommendations for future work. For this research, the failures modelled pertain primarily to actuator or control surface failures. The failures tested were as follows:

- 0° deflection of one or more control surfaces
- A 50% of full deflection failure
- A hard over failure, where (servo moves to either extreme position)

Specifically the controls to be failed are the right elevator and right aileron. The left elevator and left aileron were not failed in these tests as they show similar results to the failure of the right control surfaces and provide no additional insight. Different combinations of left and right control surface failures could be modelled, but have not been simulated. This will be recommended for future work.

5.4 Procedures Followed

A cautious approach was adopted in testing the algorithm and, as such, the recommended method of autopilot tuning by the Ardupilot developers was adopted. This involved first testing the autopilot's inner loop controls, in "Stability" or "FBW A" flight mode. Once satisfied with the results, the middle loops are then tested by switching the autopilot into "FBW B" flight mode. Finally, the outer loops are tested in "Auto" flight mode. Initially, the allocation algorithm was disabled to gain an understanding of the behaviour of the adaption algorithm; this was later introduced for more sever failure cases.

5.4.1 Inner Loop Testing

The flight mode chosen for this simulation was the "FBW A" flight mode as it provided a means to determine the suitability of the adaption and allocation algorithms during two common flight manoeuvres. While many combinations of manoeuvres could be performed, the manoeuvres chosen were to maintain level flight and to roll to and maintain a constant bank angle. This could be viewed at a system's level as the bare minimum for the aircraft to return to a "Home" location and loiter over the home location should a failure occur. These manoeuvres also provide insight into the suitability of the proposed algorithms to achieve the desired goal and to determine any shortcomings or enhancements required. It is also important to note that, for the inner loop tests, the set point is unchanging and thus the tests will show the ability of the adaption algorithm to track a fixed set point value.

The high level procedure to test the inner loops was to establish the throttle and pitch settings to maintain level and constant banked flight. Using these settings, two simulations per failure were run. One simulation was run with the unmodified autopilot code in operation and one simulation with the adaption algorithm enabled. The allocation algorithm was only used if it is found that it is required and for the more severe control failures. A detailed procedure to test the inner loop is outlined below.

1. Find the correct pitch angle and throttle setting to maintain altitude for the manoeuvre being tested.
2. Set the throttle position found in 1.
3. Set NavPitch to the pitch angle found in 1.
4. Set NavRoll to 0° for the level flight case or 20° for the banked flight case.
5. Determine the failure to be modelled from the test matrix, Table 5.3, and set up the aircraft failure algorithm to apply the desired failure at 20 seconds after the start of the simulation.
6. Run the simulation for a period of 120 seconds with the adaption and allocation algorithms disabled.
7. Rerun the simulation with the adaption algorithm enabled, observe the results to determine if the allocation algorithm is required. If it is required, rerun the simulation with both adaption and allocation algorithms enabled; if not required, proceed to the next test point in the test matrix, Table 5.3.

Table 5.3: Inner loop simulation test matrix

Test Number	Control Surface Failed	Direction	Failure	Adaption	Control Allocation
Control	None	N/A	None	On & off	Off (unless required)
1	Right elevator	N/A	0 °	On & off	Off (unless required)
2	Right elevator	Pitch up	50% hard over	On & off	Off (unless required)
3	Right elevator	Pitch up	100% hard over	On & off	Off (unless required)
4	Right elevator	Pitch up	100% hard over	On & off	ON
5	Right aileron	N/A	0 °	On & off	Off (unless required)
6	Right aileron	Roll right	50% hard over	On & off	Off (unless required)
7	Right aileron	Roll right	100% hard over	On & off	Off (unless required)
8	Right aileron	Right roll	100% hard over	On & off	ON
9	Right elevator & right aileron	Pitch up and right roll	0 °	On & off	Off (unless required)
10	Right elevator & right aileron	Pitch up and right roll	50% hard over	On & off	Off (unless required)

11	Right elevator & Right aileron	Pitch up and Right roll	100% hard over	On & off	Off (unless required)
12	Right elevator & right aileron	Pitch up and right roll	100% hard over	On & off	On
13	Right elevator & right aileron	Pitch up and right roll	50% & 100% hard over	On & off	On & off

5.4.2 Middle Loop Testing

The flight mode used to test the middle loop is the "FBW B" flight mode. This flight mode adjusts the pitch angle and throttle to achieve a constant airspeed and altitude. This will determine two important characteristics of the adaption algorithm, the first being the ability of the adaption algorithm to track a constantly changing set point and, secondly, to determine the interaction between two coupled adaption controllers. A similar procedure to that outlined in Section 5.4.1 was performed, again for level flight and banked flight, but with the test matrix presented in Table 5.4.

Table 5.4: Middle loop simulation test matrix

Test Number	**Control Surface Failed**	**Direction**	**Failure**	**Adaption**	**Control Allocation**
Control	None	N/A	None	On & off	On
1	Right elevator	N/A	0 °	On & off	On
2	Right elevator	Pitch up	50% hard over	On & off	On
3	Right elevator	Pitch up	100% hard Over	On & off	On
4	Right aileron	N/A	0 °	On & off	On
5	Right aileron	Roll right	50% hard over	On & off	On
6	Right aileron	Roll right	100% hard over	On & off	On
7	Right elevator & right aileron	Pitch up and right roll	50% hard over and 100% hard over	On & off	On & off

5.4.3 Outer Loop Testing

The flight mode to test the outer loop is the "Auto" flight mode. Here, the autopilot is running in its most advanced form. This provides insight into the ability of the adaption algorithm to undergo a number of manoeuvres. Again a similar procedure as outlined in the previous sections was undertaken. However, only one flight manoeuvre was flown, that being waypoint following. A set of three waypoints were defined to form a triangular flight path as shown in Figure 5.2. The test matrix used to test the outer control loop is presented in Table 5.5.

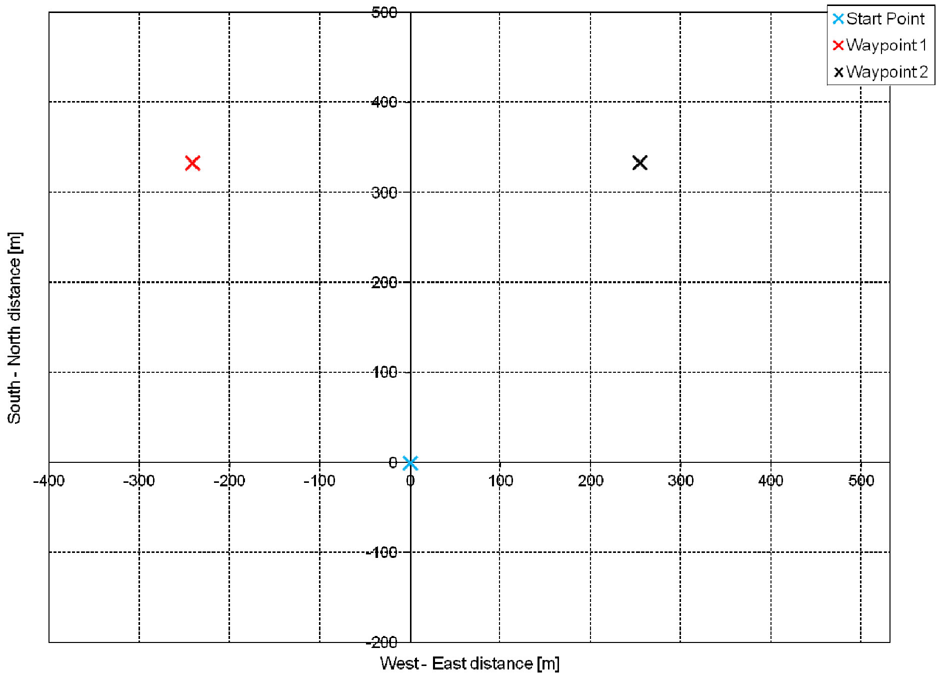


Figure 5.2: Waypoints used to test the outer control loops

Table 5.5: Outer loop simulation test matrix

Test Number	**Control Surface Failed**	**Direction**	**Failure**	**Adaption**	**Control Allocation**
Control	None	N/A	None	On & off	On
1	Right elevator	N/A	0 °	On & off	On
2	Right elevator	Pitch up	50% hard over	On & off	On
3	Right elevator	Pitch up	100% hard Over	On & off	On
4	Right aileron	N/A	0 °	On & off	On
5	Right aileron	Roll right	50% hard over	On & off	On
6	Right aileron	Roll right	100% hard over	On & off	On
7	Right elevator & right aileron	Pitch up and right roll	50% hard over and 100% hard over	On & off	On & off

5.5 Results

5.5.1 Inner control loop testing

The inner loop of the autopilot was tested first. The autopilot was “flown” in FBW A mode as described in Section 5.4.1 and the trim position determined. The pitch angle to maintain level flight was found to be 1.25° and a throttle setting of 36% was found to maintain a speed of approximately 25m/s. The tests were also run with the wings banked at 20° and the pitch angle set to 1.88° to maintain altitude with the same throttle setting used. These settings were then applied to the simulation and the test matrix followed.

To display the results of all aircraft state data would be considerable: hence for each test in this section, the primary point of interest will be the pitch and roll angles as these are the controlled parameters of the inner flight modes. The control deflection data as well as the PID gain coefficient data will also be supplied when relevant. Where any other data is of interest, it is included; however, the full set of results is found in the summary provided in Appendix C. The simulation code, also found in Appendix C can be re-run to obtain an extensive set of results.

Control - No Failures

The simulation was run without inducing any failures, both with and without the adaption algorithm for commanded level flight. Figure 5.3, which shows the time history of the pitch and bank angle, shows that a slight overshoot in the set pitch angle occurred for both the case with the adaption algorithm enabled and disabled. The overshoot is slight in both cases (less than 1°) and is unlikely to be noticed in flight. It can be seen that the simulation with the adaption mechanism enabled has a slightly smaller overshoot which indicates that the adaption mechanism is working to some degree. Figure 5.4 (a) shows the variation in PID gain constants and shows that, initially, there is a rapid increase in the pitch proportional gain, followed by a rapid decrease as the adaption algorithm detects an overshoot condition. There is also a very slight increase in integral gain and derivative gain. However, these are very minor. As previously discussed in Section 4.3.6, the adaption rates for the integral and derivative gains are low to avoid unstable and rapid changes in theses parameters as this was found to cause unstable control of the complete system.

The roll angle in Figure 5.3 is seen to have a disturbance from level flight attributed to a slight asymmetry in the wind tunnel model. This is corrected by the autopilot with a slight aileron deflection. The response is somewhat oscillatory for both cases with and without the adaption algorithm enabled indicating that perhaps the proportional gain in the roll direction is slightly too high. However, the oscillations are small and hence modifications to the base PID gains are not warranted and the tests were continued. The adaption algorithm attempts to reduce the high proportional gain while increasing the integral gain slightly, as shown in Figure 5.4 (b).

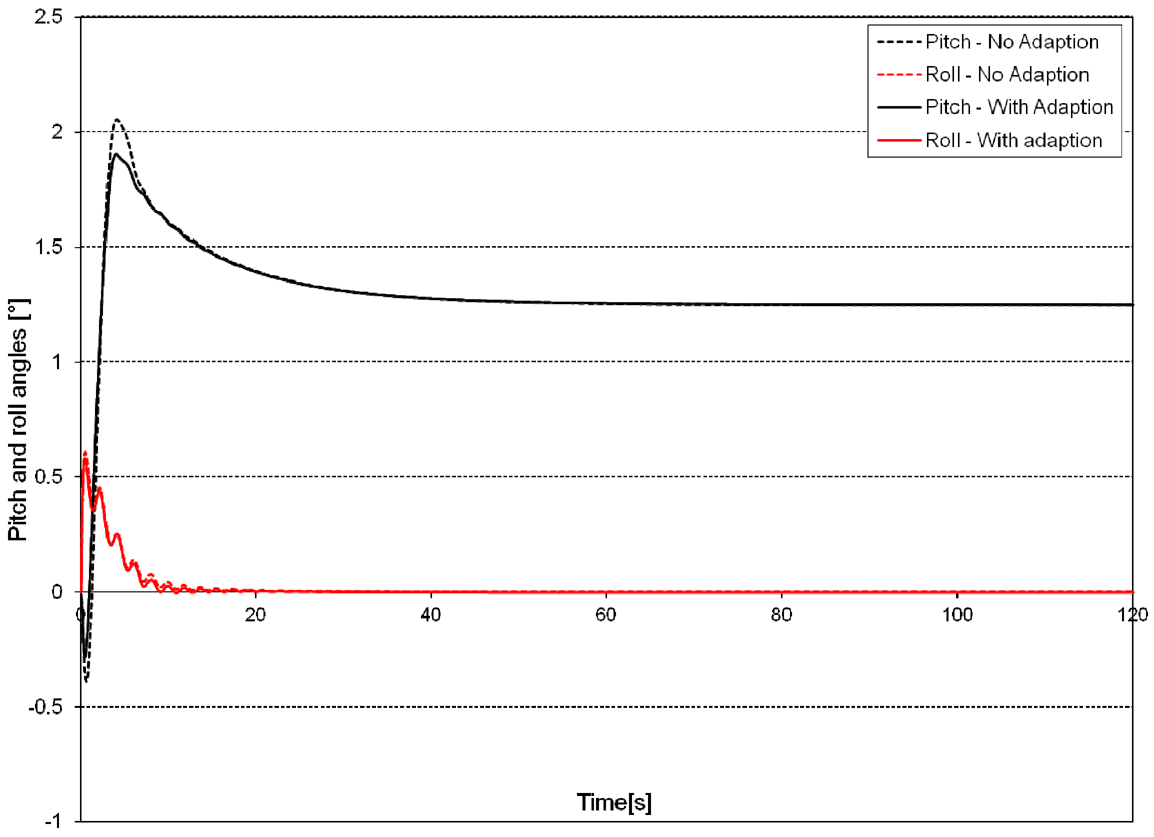


Figure 5.3: Simulated time history of pitch and bank angles (no failures) in level flight

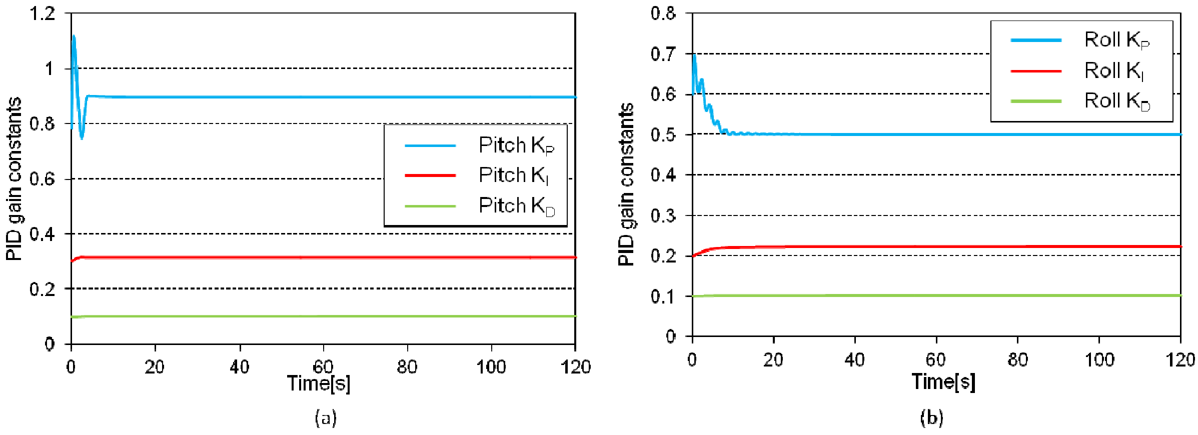


Figure 5.4: Simulated time history of the pitch (a) and roll (b) PID constants with the adaption algorithm enabled (no failures) in level flight

It can be seen from Figure 5.3 that the autopilot achieved the desired pitch and roll angles almost exactly within 40 seconds and 20 seconds respectively and this performance is considered for the purposes of this research to be satisfactory.

The simulation was run again without inducing any failures both with and without the adaption algorithm for constantly banked flight. The time history of pitch and roll angles are presented in Figure 5.5 which shows a similar trend to the level flight case. The pitch angle did, however, exhibit a more oscillatory response but the magnitude of the oscillations was minor. Only a slight difference existed between the case with the adaption algorithm enabled and with it disabled. Figure 5.6 (a) shows the gain adjustment for the pitch PID controller and shows an initial spike in proportional gain followed by a rapid decrease back to base levels. There is then an oscillatory behaviour as the adaption algorithm attempts to track the ideal model until the control set point has been achieved. One may argue that the performance of the pitch control system has degraded somewhat in that the behaviour is more oscillatory; however, it can be argued that the time to reach the desired set point has been slightly reduced and that the increase in oscillations is minor.

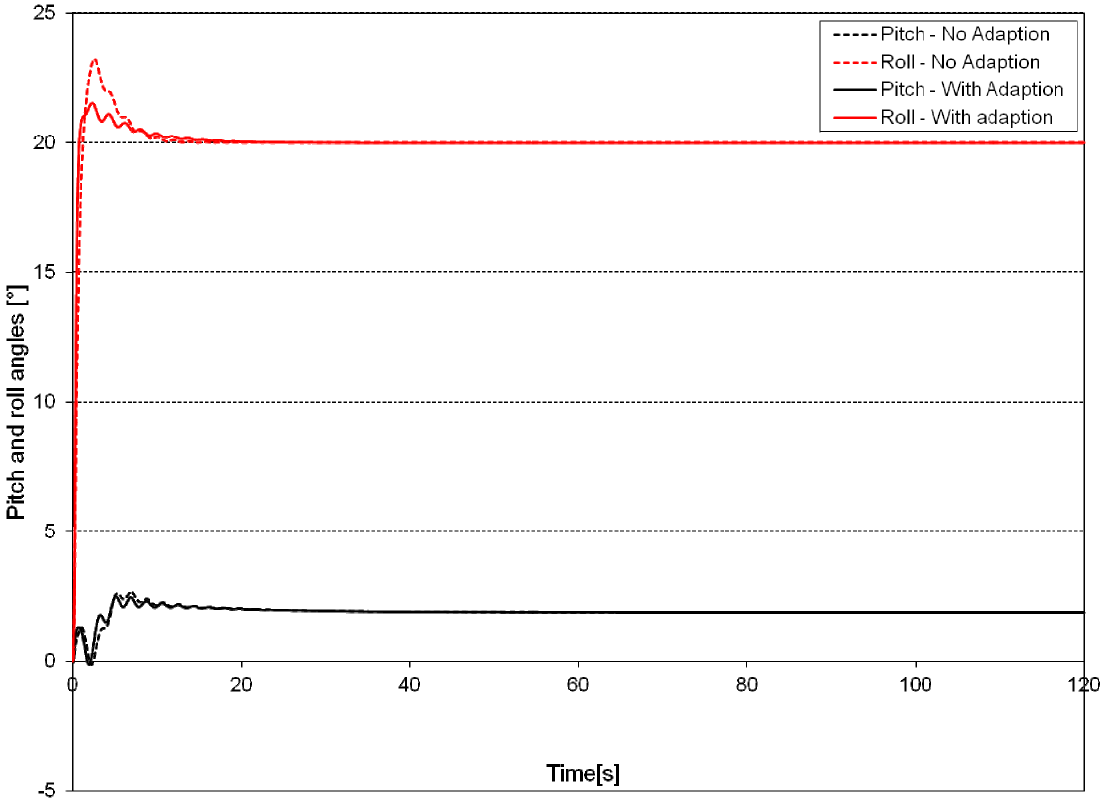


Figure 5.5: Simulated time history of pitch and bank angles (no failures) during a constant banked turn

Figure 5.5 shows more clearly the overshoot of the roll angle from the set point although it is only of the order of 2.5° for the case where the adaption mechanism is disabled. When the adaption mechanism is enabled, this overshoot is reduced by approximately half and the time to reach steady state is approximately the same. The adaption algorithm increased the roll proportional gain while not modifying the integral or derivative gains significantly, as shown in Figure 5.6 (b).

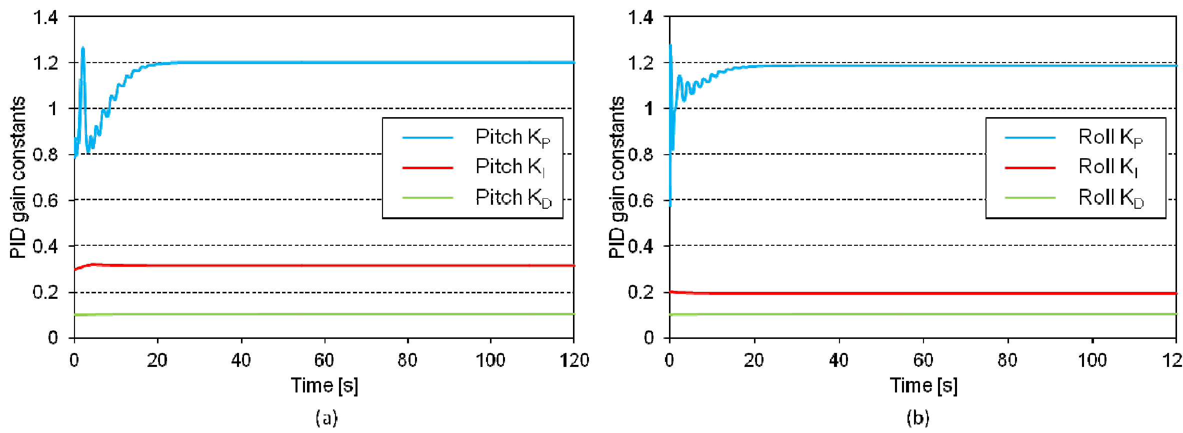


Figure 5.6: Simulated time history of the pitch (a) and roll (b) PID constants with the adaption algorithm enabled (no failures) during a constant banked turn

It can be seen from Figure 5.5 that the autopilot was able to achieve the set pitch and roll angles satisfactorily with a slight performance advantage being gained with the adaption algorithm enabled. This indicates that for constant set point commands, the adaption algorithm appears to be functioning correctly and that it does not inhibit the correct operation of the autopilot system but improves control behaviour very slightly under normal operating circumstances.

Right Elevator Failure - 0° Deflection (Test 1)

The simulation was run with a 0° right elevator deflection failure. That is, the elevator moves from its current position, at the time of failure, to its neutral position. For this flight position and aircraft configuration, a slight down elevator trim is required. Hence, as the elevator returns to its neutral position, a nose up and right roll disturbance is created. Figure 5.7 clearly shows the point at which the elevator failed and shows that the failure causes a nose up pitching moment coupled with a rolling moment to the right which is seen by the positive diversion in both pitch and roll angles.

Figure 5.7 shows that, with the adaption algorithm disabled, the pitch change is of the order of 1°. This pitch change is then followed by a rapid decrease in pitch angle as the autopilot corrects with the remaining elevator. A slight oscillation occurs before the desired set point is re-established. With the adaption algorithm enabled, an improvement in control performance is noted. The initial disturbance causes a smaller change in pitch angle and fewer oscillations exist with a more stable path to the desired set point. Figure 5.8 (a) shows a rapid increase in pitch proportional gain after the failure to aid in reducing the overshoot condition. This is then followed by a minor decrease in proportional gain to prevent the overshoot in the negative sense. Little change was made to the integral and derivative gains in this case.

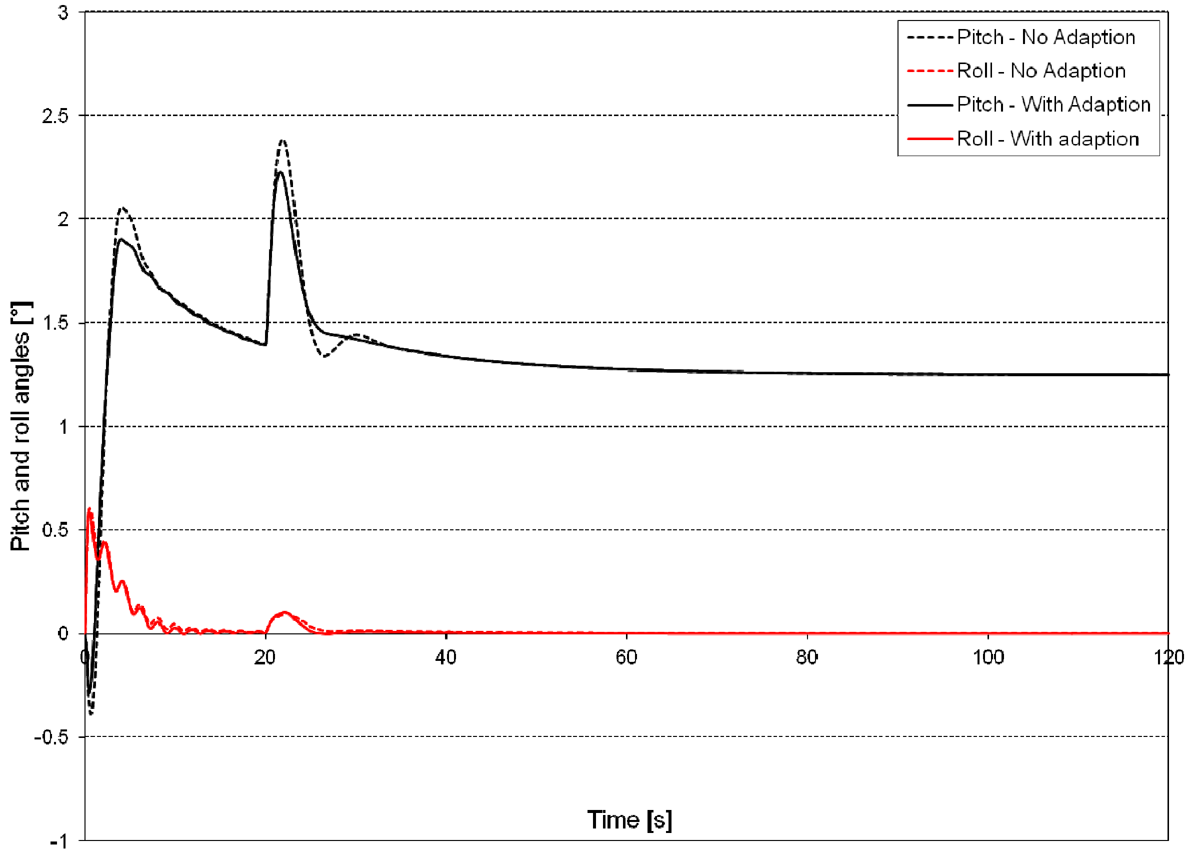


Figure 5.7: Simulated time history of pitch and bank angles (0° right elevator deflection failure) in level flight

Figure 5.7 also shows the roll disturbance created as the elevator moved to the neutral position. The effect was small and little difference can be observed between the case with the adaption algorithm enabled and disabled. Figure 5.8 (b) shows a slight increase in roll proportional gain at the moment of failure. This is reduced back to base levels shortly after the failure. A slight increase in roll integral gain is noted with no increase in derivative gain. It is hypothesised that the integral gain has increased slightly as the fuzzy controller may have interpreted the error as a steady state error due to the size of the disturbance created.

Figure 5.7 shows that the autopilot algorithms and gain tuning is robust enough to tolerate this fault without the use of the adaption algorithms. However, some minor control performance improvement was noted with the adaption algorithm enabled.

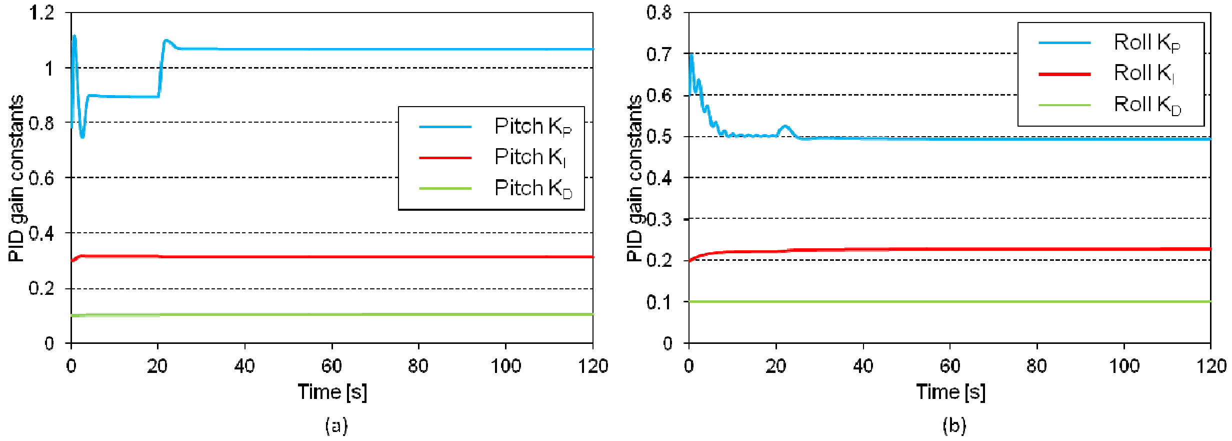


Figure 5.8: Simulated time history of the pitch (a) and roll (b) PID constants with the adaption algorithm enabled (0° right elevator deflection failure) in level flight

Due to the nature of the failure and the direction of the turn, no significant result was obtained for the case of a constant banked turn with 0° elevator deflection as the trim condition had the elevator very close to 0°. The results have not been included here as they offer no insight into the behaviour of the system. They can, however, be found in Appendix C, if desired.

Right Elevator Failure - 50% Hard Over (Test 2)

The simulation was run with a more severe failure, namely a 50% up elevator deflection. Figure 5.9 shows the results of this simulation in the form of a time history plot of pitch and roll angles. It can be seen in Figure 5.9 that the elevator failure caused a large pitch change in the nose up direction coupled with a slight roll to the right. In the case where the adaption algorithm was disabled, the pitch error reached a maximum of approximately 8° after which there were four oscillations around the set point before the set point was reached. This was significantly reduced in the case with the adaptation algorithm enabled to approximately 5.5°, equating to a reduction in error of approximately 37%. This reduction in error was coupled with a lack of oscillations about the set point. Figure 5.10 (a) shows that the adaption algorithm increased the pitch proportional gain rapidly to accommodate the increasing pitch error after which there is a slight decrease in proportional gain until the error has reduced. At this point, the proportional gain remains constant. Little change in the integral and derivative gains was noted.

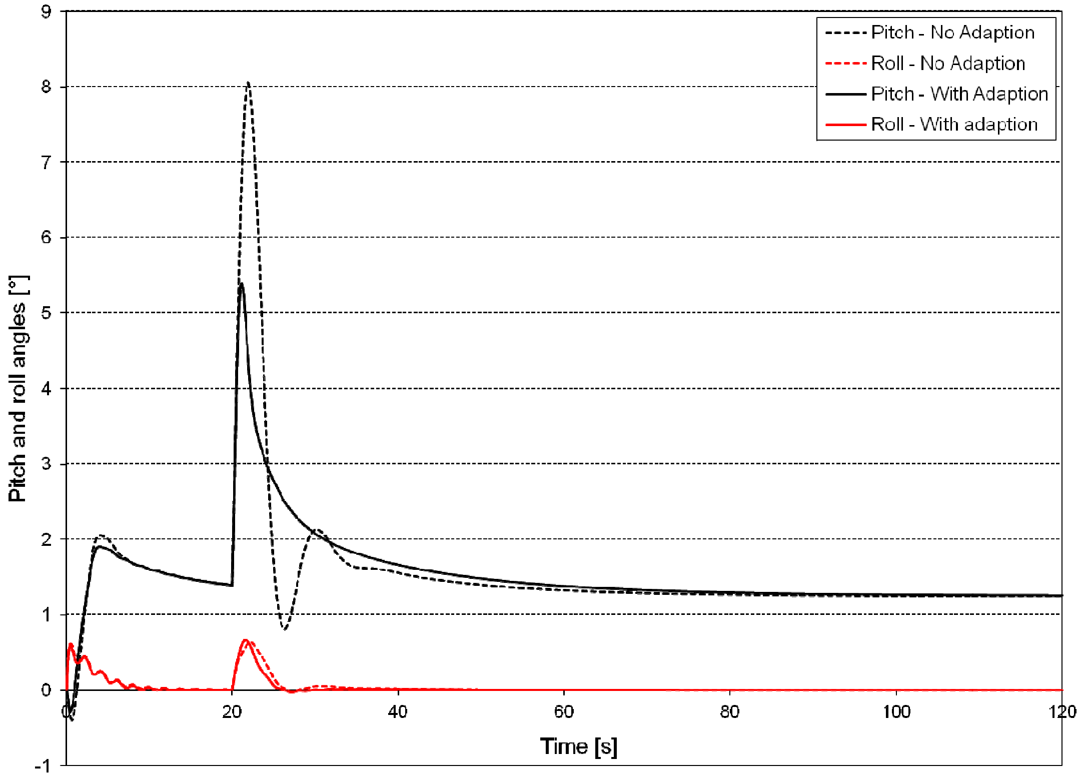


Figure 5.9: Simulated time history of pitch and bank angles (right elevator 50% hard over failure causing nose up pitching moment) in level flight

Figure 5.9 also shows that the autopilot was able to tolerate the roll perturbation without the adaption algorithm but it was noted that there was a marginal increase in control performance. Figure 5.10 (b) shows that the proportional gain rose rapidly, but marginally, with a slight increase in the integral gain. Little change in derivative gain occurred for the roll PID controller.

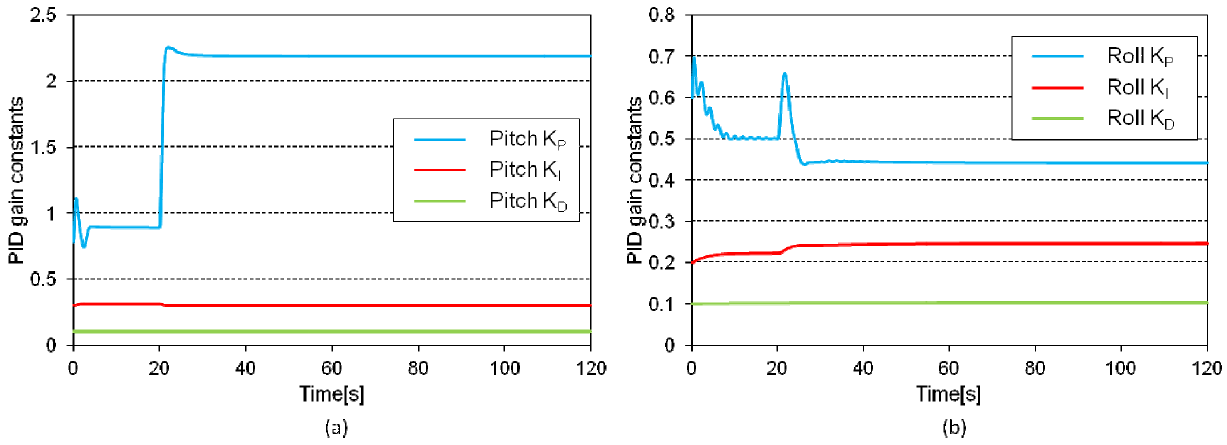


Figure 5.10: Simulated time history of the pitch (a) and roll (b) PID constants with the adaption algorithm enabled (right elevator 50% hard over failure causing a nose up pitching moment) in level flight

Figure 5.11 (a) to (d) shows the time history of the control deflections during the simulation. As can be seen in Figure 5.11 (a), the right elevator moves to an angle of 7.5° (50% of 15°) and remains at that position for the duration of the simulation while the left elevator deflects to reduce the change in pitch angle in the opposite direction. A new equilibrium is reached that is within the control limits set in the autopilot. Figure 5.11 (b) shows a change in aileron deflection as the autopilot has to find a new trim position; however, this is easily achieved with only a minor change in deflection. The slight rudder input seen in Figure 5.11 (c) is as a result of the aileron-rudder mixing used in a feedforward loop to improve the ability of the autopilot to fly coordinated turns.

The results of this simulation clearly show that the autopilot is able to tolerate this failure without the adaption algorithm, although possibly only with a reduced flight envelope. However, great gains in control performance can be achieved with the proposed adaption algorithm.

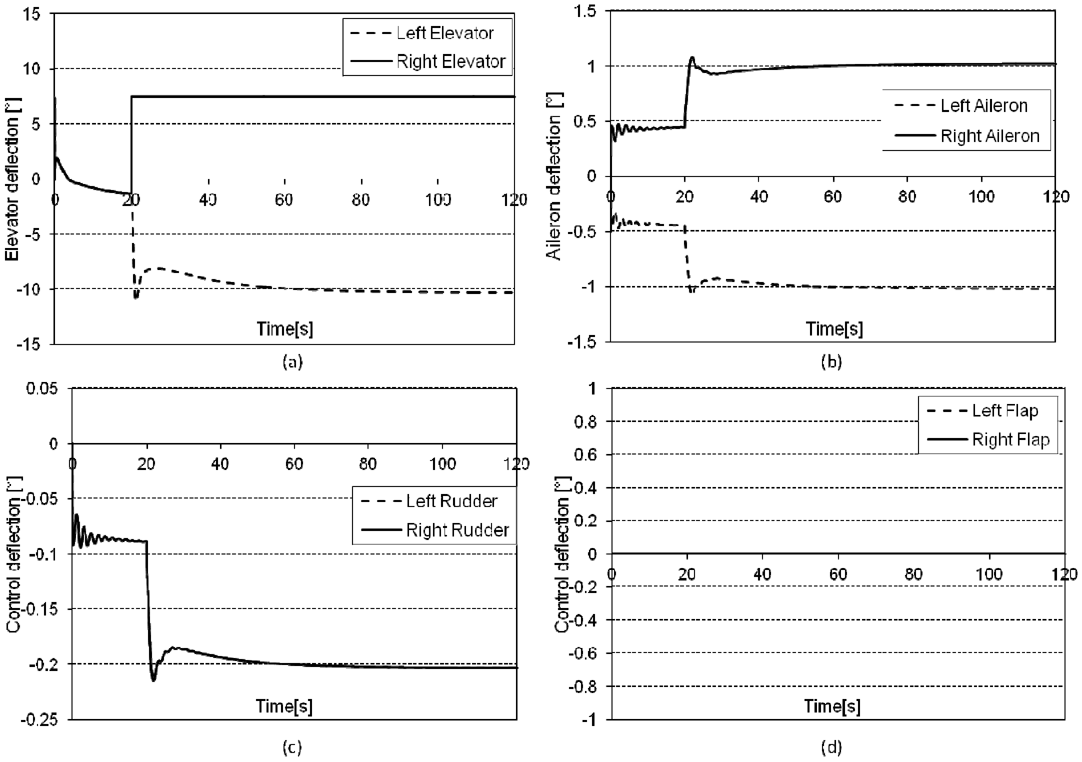


Figure 5.11: Simulated time history of the elevator (a), aileron (b), rudder (c) and flap (d) deflections with the adaption algorithm enabled (right elevator 50% hard over failure causing a nose up pitching moment) in level flight

The simulation was run again but with the aircraft at a constant bank angle of 20°. Figure 5.12 shows the disturbance in both pitch and roll angles and shows a similar trend to the level flight case. Initially, there is a rapid change in pitch angle which is brought under control by the autopilot. The case without adaption, again, shows a higher pitch error with an overshoot on the return to the set point. When the adaption algorithm is enabled, the pitch error is reduced by approximately 35%. In addition, there is a more rapid return to the set point value with no oscillations observed. The roll perturbation was minor but with the adaption algorithm disabled, there were minor oscillations before the set point was reached. With the adaption algorithm enabled, the error was reduced and no oscillation was observed.

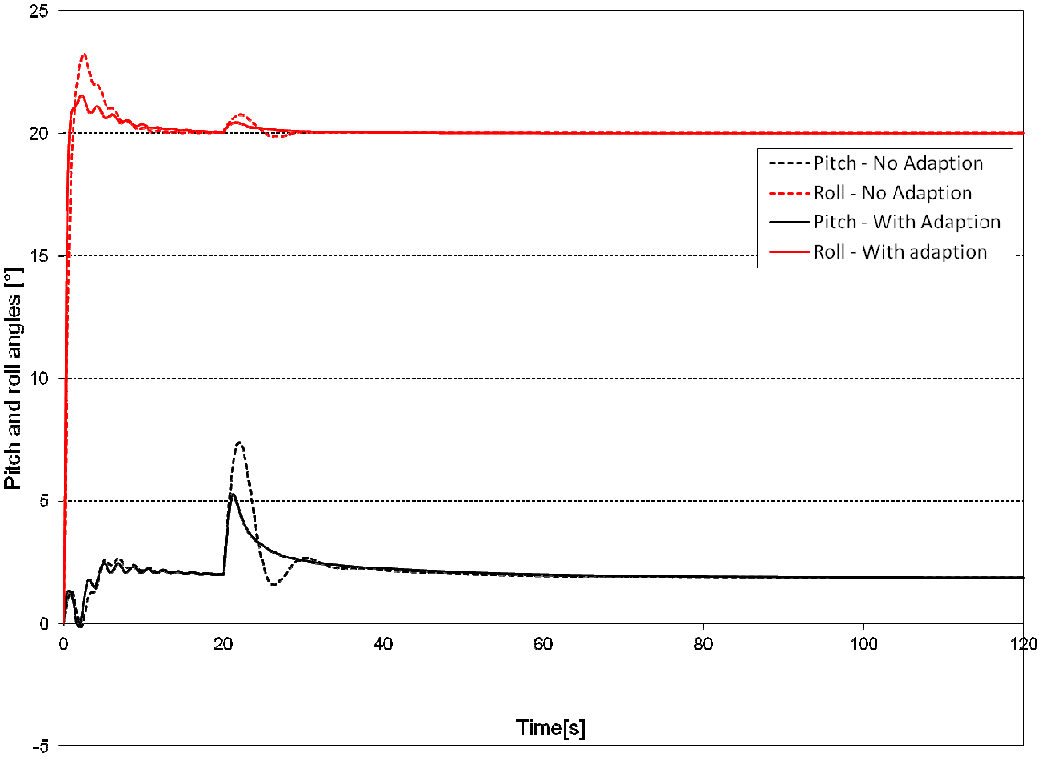


Figure 5.12: Simulated time history of pitch and bank angles (right elevator 50% hard over failure causing nose up pitching moment) during constant banked flight

Similarly to the level flight case, the pitch proportional gain rose rapidly and then dropped slightly before reaching a stable value, as shown in Figure 5.13 (a). The integral and derivative gains were marginally changed. The proportional gain for the roll PID controller was increased slightly after the failure but remained constant shortly thereafter as seen in Figure 5.13 (b). Little to no change was observed for the other gain constants.

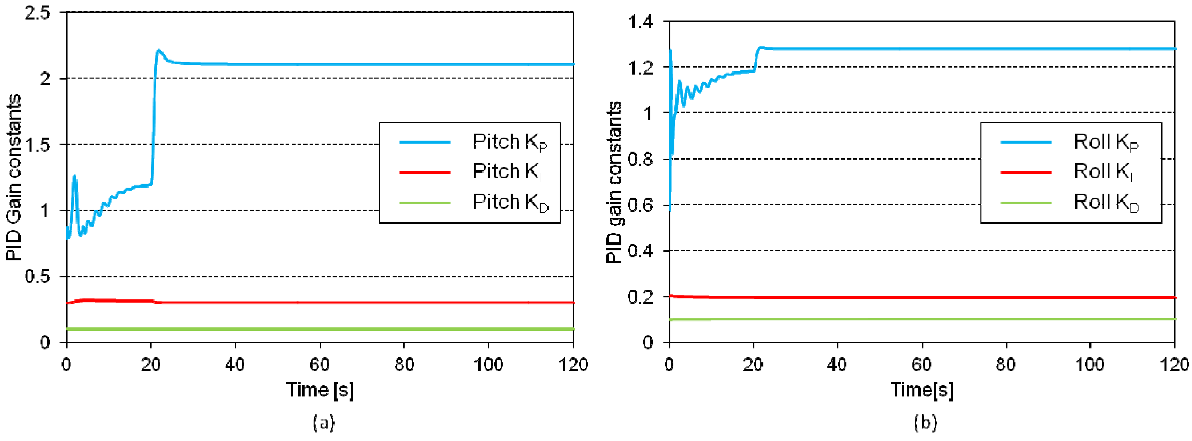


Figure 5.13: Simulated time history of the pitch (a) and roll (b) PID constants with the adaption algorithm enabled (right elevator 50% hard over failure causing a nose up pitching moment) during constant banked flight

Right Elevator Failure - 100% Hard Over (Test 3)

The simulation was run for a 100% hard over elevator failure. Figure 5.14 shows the time history of pitch and roll angles for this simulation. It shows that, for the case where the adaption mechanism was disabled, the maximum pitch angle attained was approximately 13.75°. It was also noted that there was a significant overshoot on the return to the set point value. Again, a number of oscillations occurred about the set point value as was seen in the 50% elevator failure case. However, instead of the autopilot returning the aircraft to the set point value, a state of oscillatory behaviour was seen. The case where the adaption algorithm was enabled resulted in a far lower maximum pitch angle, approximately 9°, with a small overshoot on returning to the set point and a significantly faster return to a pitch angle close to the set point; however, the oscillatory behaviour noted in the non-adapted case was still present. Figure 5.15 (a) shows that there is a rapid increase in pitch proportional gain as the adaption algorithm attempts to prevent the increase in pitch error. This is followed by a slight decrease and then a stepped increase in proportional gains during the oscillatory period.

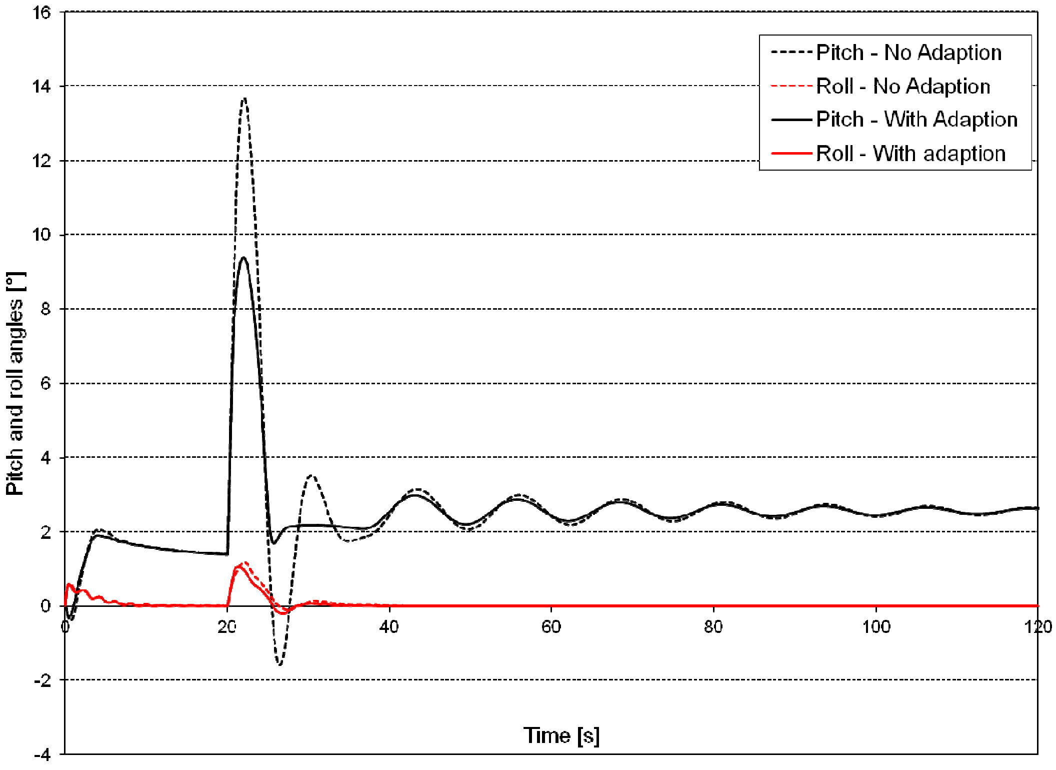


Figure 5.14: Simulated time history of pitch and bank angles (right elevator 100% hard over failure causing nose up pitching moment) in level flight

After the failure of the elevator, there is also corresponding roll to the right as can be seen in Figure 5.14. The case without adaption is slightly more delayed in returning to the set point with a marginally higher overshoot than the case with adaption; however, in practical terms the differences between adaption and non adaption is slight. Figure 5.15 (b) shows that the proportional gain increases as the failure occurs but then reduces to a value lower than the starting base value.

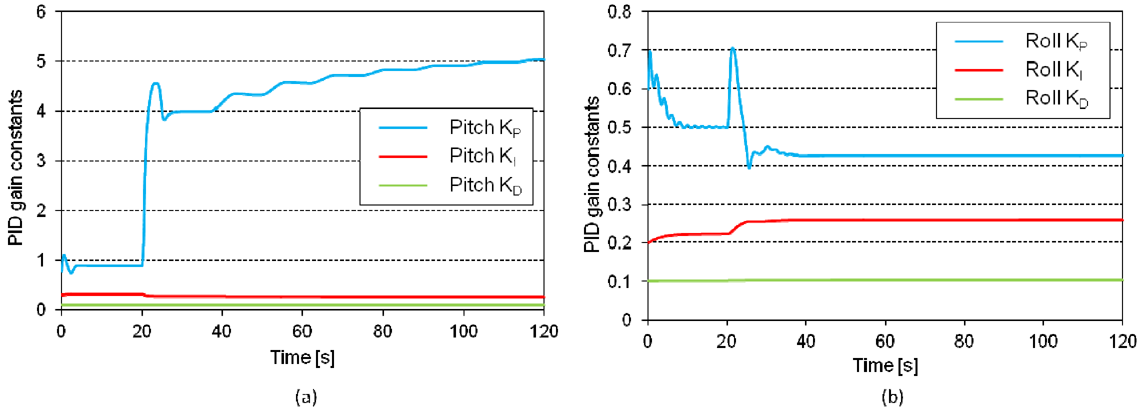


Figure 5.15: Simulated time history of the pitch (a) and roll (b) PID constants with the adaption algorithm enabled (right elevator 100% hard over failure causing a nose up pitching moment) in level flight

The oscillatory pitch behaviour noted in Figure 5.14 can be explained by examining Figure 5.16 (a). In the figure, the right elevator moves to its extreme value of 15°, the left elevator moves in the opposite direction as controlled by the autopilot and quickly reaches -15°. At this point, no remaining control authority in the pitch direction exists and the aircraft is now subject to its own stick-fixed dynamic behaviour which is a long damped phugoid motion, as seen in Figure 5.14. Figure 5.16 (b) to (d) show that no other control limits were exceeded.

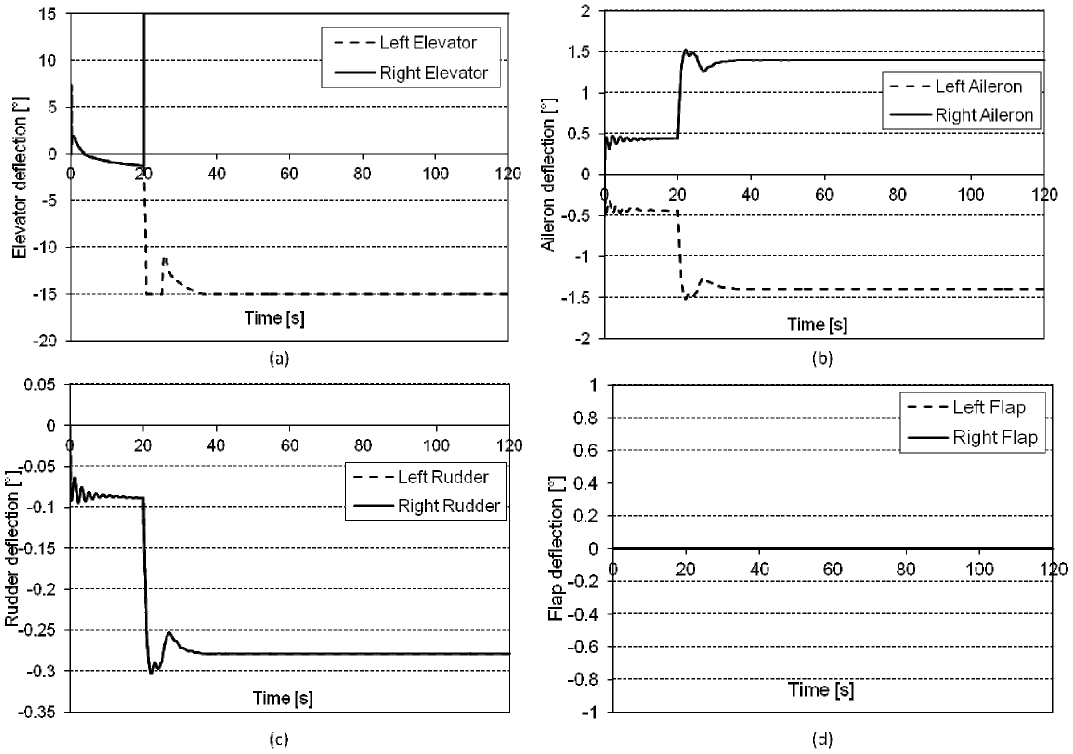


Figure 5.16: Simulated time history of the elevator (a), aileron (b), rudder (c) and flap (d) deflections with the adaption algorithm enabled (right elevator 100% hard over failure causing a nose up pitching moment) during constant banked flight

The simulation was run again but with the aircraft at a constant bank angle of 20°, the results of which are presented in Figure 5.17. Figure 5.17 shows that the pitch angle increases rapidly as the elevator fails, similarly to the level flight case, and again the case with no adaption results in a large pitch error, followed by a number of oscillations before returning gradually to a value close to the set point value. The case with adaption shows that a lower pitch error was achieved (a reduction of 5°) although it is interesting to note a slight behavioural change when compared to the level flight case. The return to the set point value is not as abrupt and a gradual return to the set point occurs. Similarly to the level flight case, control saturation is reached; however, the oscillatory behaviour is minor and occurs significantly later than shown in Figure 5.14 indicating that a different manoeuvre may yield a drastically different result. Again, the oscillatory behaviour that develops towards the end of the simulation can be explained by examining Figure 5.18 (a). The left elevator reaches its maximum value more slowly than the level flight case and at approximately 80 seconds the control surface saturates and the oscillatory behaviour begins. Similarly to the level flight case the other control surfaces did not approach their limits of travel as seen in Figure 5.18 (b) to (d). Figure 5.17 also shows the roll disturbance generated by the elevator failure. As can be seen in the figure, the case with the adaption algorithm enabled shows improved control performance.

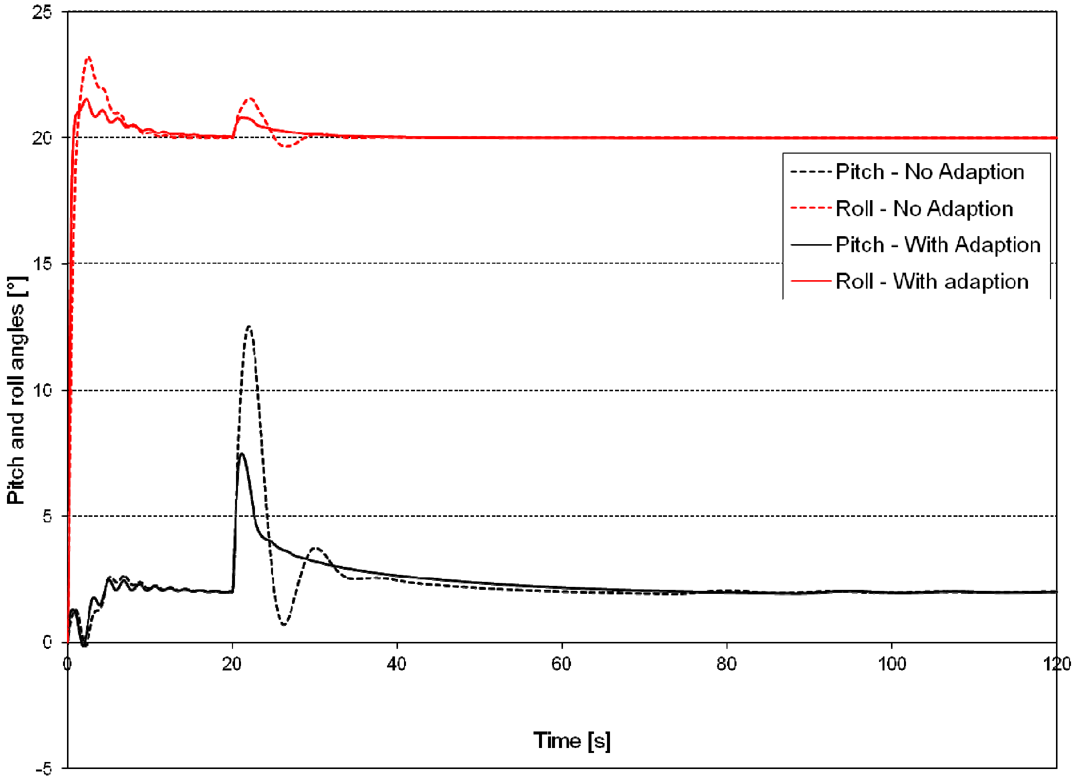


Figure 5.17: Simulated time history of pitch and bank angles (right elevator 100% hard over failure causing nose up pitching moment) during constant banked flight

Figure 5.19 (a) shows a rapid increase in proportional gain which then maintains a reasonably constant value. Again a slight increase is noted towards the end of the simulation as the elevator control saturates. Little change is observed in the integral and derivative gains. At the point of failure, the roll PID proportional gain increases but quickly establishes a constant value, as shown in Figure 5.19 (b).

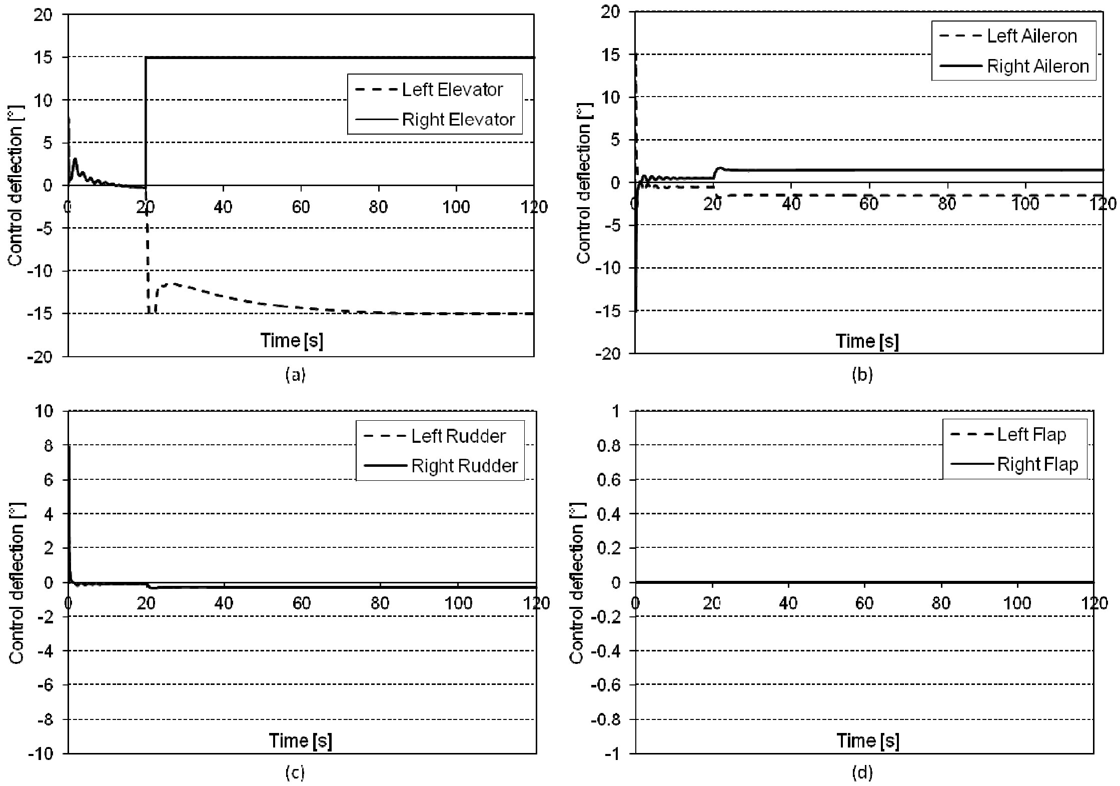


Figure 5.18: Simulated time history of the elevator (a), aileron (b), rudder (c) and flap (d) deflections with the adaption mechanism enabled (right elevator 100% hard over failure causing a nose up pitching moment) during constant banked flight

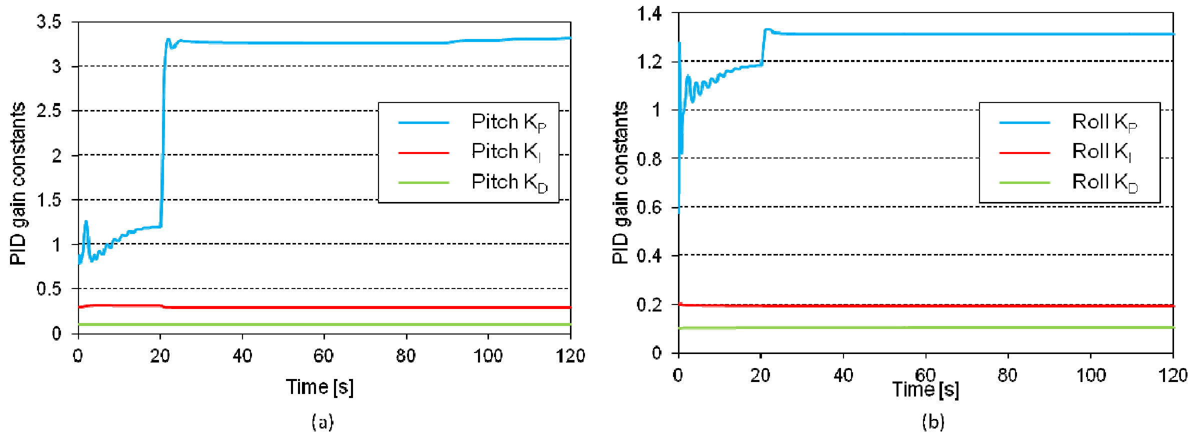


Figure 5.19: Simulated time history of the pitch (a) and roll (b) PID constants with the adaption algorithm enabled (right elevator 100% hard over failure causing a nose up pitching moment) during constant banked flight

Right Elevator Failure - 100% Hard Over (Control allocation enabled) (Test 4)

The simulation was run with a 100% hard over failure of the right elevator as conducted previously; however, in this case, the allocation algorithm was enabled. As can be seen in Figure 5.20, much improved behaviour was observed. The maximum pitch error induced was similar to the previous test; however, the oscillations seen were not evident in this case, as the control allocation algorithm made use of the secondary control surface to provide the remaining pitch control required. The flaps were used to provide a slight nose down pitching moment, as can be seen in Figure 5.21 (d) after the elevators had reached their travel limits, shown in Figure 5.21 (d). In Figure 5.21 (d) an initial flap deflection is observed to aid in reducing the maximum pitch error. This is followed by a rapid reduction in flap deflection as sufficient elevator authority was available and then a subsequent build up in flap deflection as more pitch authority was required. Figure 5.21 (d) shows that the flaps do not provide a great deal of pitch authority and a large flap deflection is required to achieve a small pitch change. There was no lack of roll authority for this test and the results obtained in the roll direction are very similar to those obtained in the previous tests. This is confirmed by Figure 5.21 (b) and (c) which shows that the ailerons and rudder were only slightly deflected.

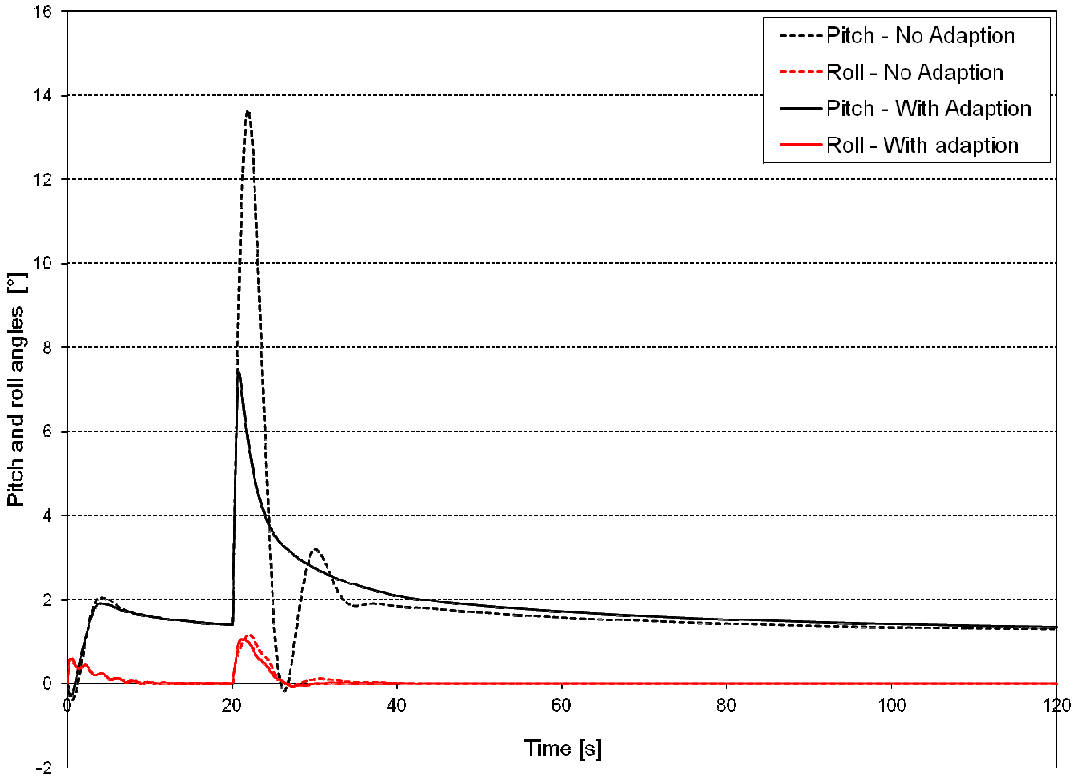


Figure 5.20: Simulated time history of pitch and bank angles (right elevator 100% hard over failure causing nose up pitching moment) in level flight with control allocation enabled

The increase in control authority has improved the stability of the PID gain changes. Figure 5.22 (a), shows a rapid increase in pitch proportional gain which stabilises at a constant level just under 3.5 as opposed to the previous case without the allocation algorithm which showed a steady increase in proportional gain. This indicates the importance of using the allocation algorithm and the mitigation of the potential risk of the adaption algorithm causing unstable behaviour. There were minor differences present in Figure 5.22 (b) when compared to Figure 5.15 (b) in that the increase seen in Figure 5.15 (b) after the failure was not evident in Figure 5.22 (b).

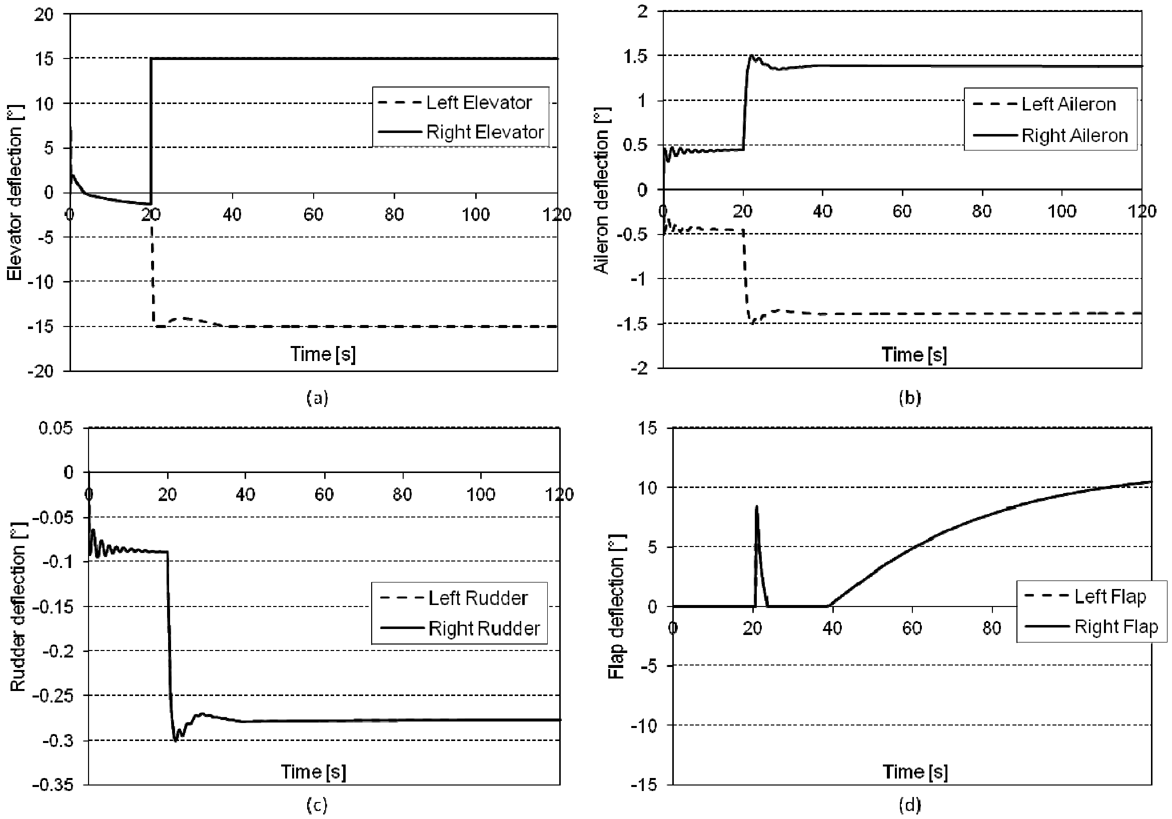


Figure 5.21: Simulated time history of the elevator (a), aileron (b), rudder (c) and flap (d) deflections with the adaption algorithm enabled (right elevator 100% hard over failure causing a nose up pitching moment) during constant banked flight with control allocation enabled

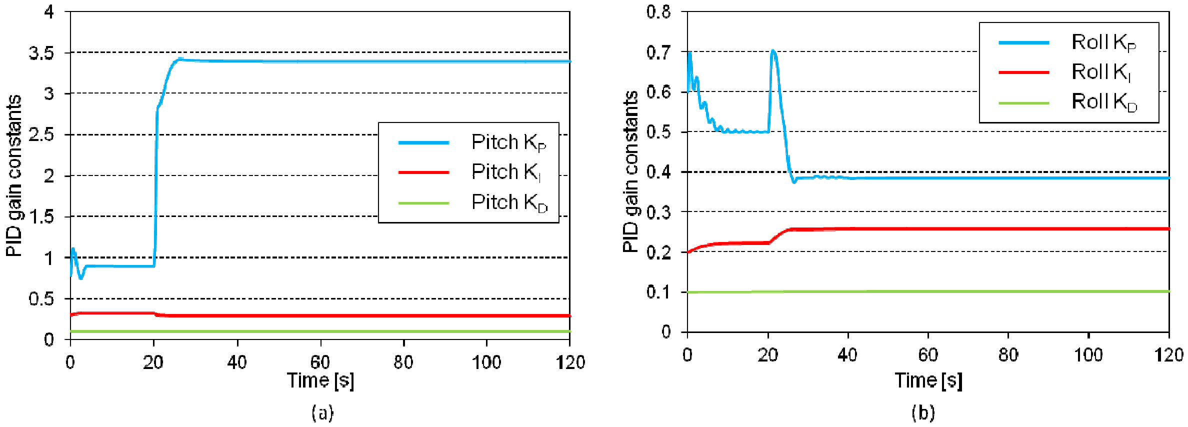


Figure 5.22: Simulated time history of the pitch (a) and roll (b) PID constants with the adaption algorithm enabled (right elevator 100% hard over failure causing a nose up pitching moment) in level flight with control allocation enabled

The simulation was re-run but with the aircraft in a banked turn. This yielded very similar results to those obtained in Figure 5.17. There were minor differences due to the inclusion of flap deflection to assist with the correction of the pitch error. The saturation of the elevator towards the end of the simulation which resulted in a slight oscillation about the set point in Figure 5.17 was not evident when the allocation algorithm was enabled. The results have not been included here as the differences were minor; however they can be found in Appendix C, if desired.

Right Aileron Failure - 0° Deflection (Test 5)

The simulation was run with a 0° aileron deflection failure at 20 seconds after the start of the simulation. Figure 5.23 shows that there was little change in pitch and no discernable difference could be made between the case where the adaption algorithm was enabled or disabled. The effect on roll was more pronounced but only a small bank angle error (0.5°) was induced. This is to be expected for level flight with the only rolling moment being generated by the asymmetries of the aircraft. Figure 5.23 does show a slight difference in control performance in roll with a more rapid response being generated when the adaption algorithm was enabled. This was, however, followed by a slight overshoot, not seen in the case without adaption. In practical terms the difference is negligible.

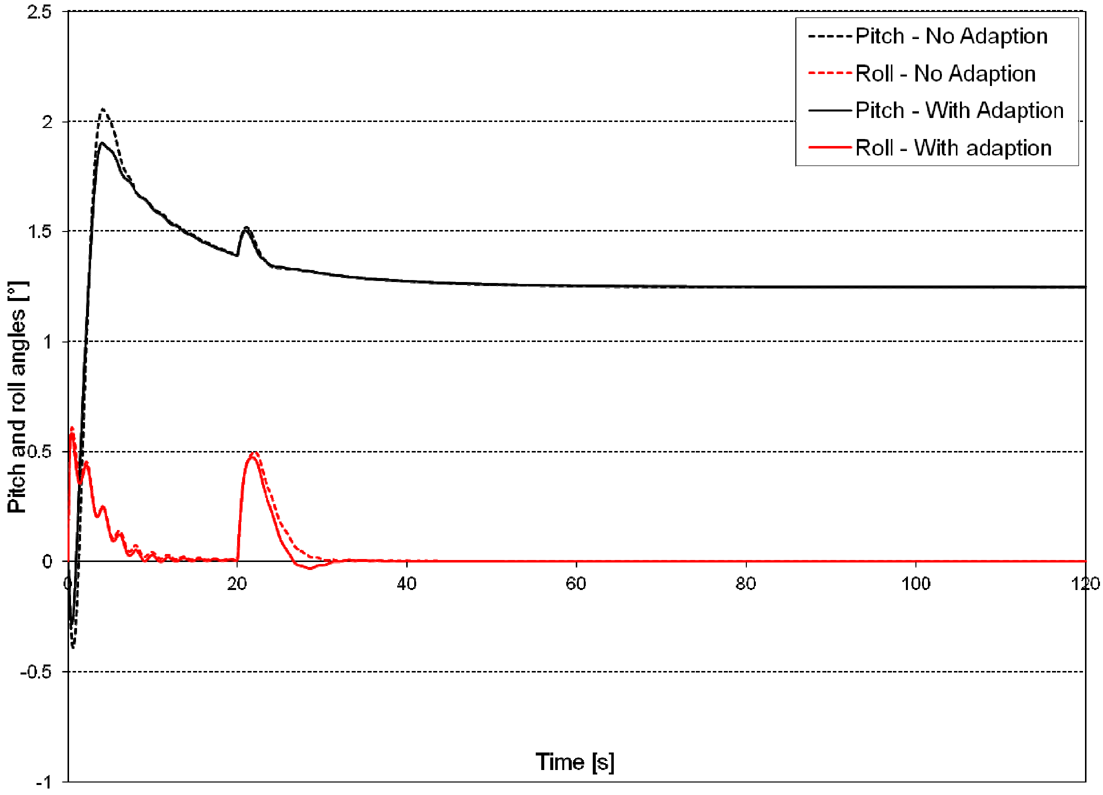


Figure 5.23: Simulated time history of pitch and bank angles (0° right aileron deflection failure) in level flight

Figure 5.24 (a) and (b) show a small change in the proportional gain for both pitch and roll controllers with a slight gain in the integral gain for the roll controller due to the small nature of the errors induced by the failure. Little other change was noted.

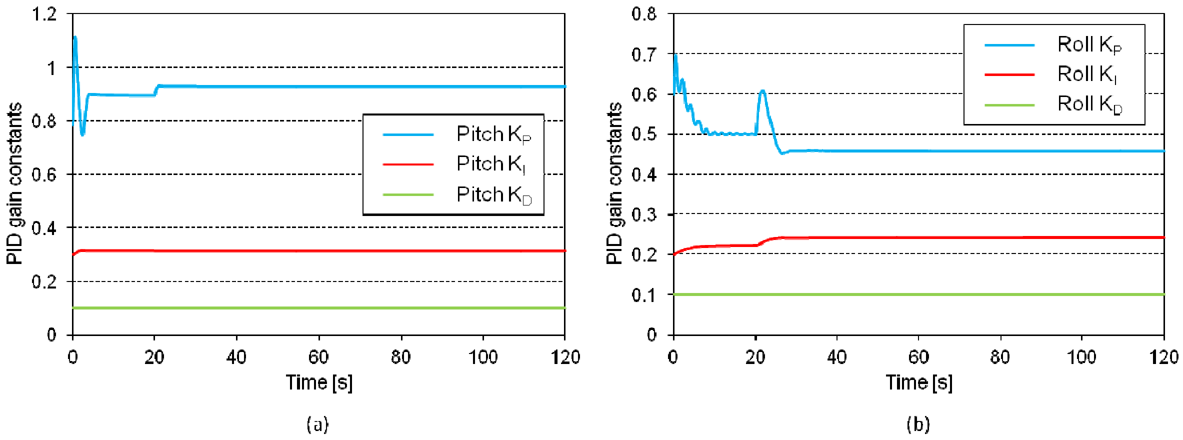


Figure 5.24: Simulated time history of the pitch (a) and roll (b) PID constants with the adaption algorithm enabled (0° right aileron deflection failure) in level flight

Only a small effect was noted during the simulation of a banked turn. The aircraft was at steady state at the time of failure and so only a small aileron change was induced at the moment of failure. Because of this, the remaining results have been excluded as they add little insight into the behaviour of the adaption algorithm. It was noted, however, that the autopilot is tolerant to this failure and there is little need for an adaption algorithm in this case. Should the results of this test be required they can be found in Appendix C.

Right Aileron Failure - 50% Hard Over (Test 6)

A 50 % hard over failure of the right aileron was modelled with and without the adaption algorithm, the results of which can be found in Figure 5.25 (a) and (b). Figure 5.25 (b) shows that, with the adaption algorithm disabled, there is a large and rapid increase in bank angle to approximately 9°. This is followed by a steady and equally as rapid return to level flight as the autopilot compensates for the error. When the adaption algorithm is enabled, the maximum bank angle is reduced to approximately 5°. The return to level flight is also noted to take marginally longer with a slight oscillatory behaviour being present. Upon examination of Figure 5.26 (b), it can be seen that the proportional gain rises rapidly to mitigate the change in bank angle. It then decreases in a slightly stepped manner. It is theorised that the step changes in proportional gain have caused the oscillatory behaviour of the bank angle in Figure 5.25.

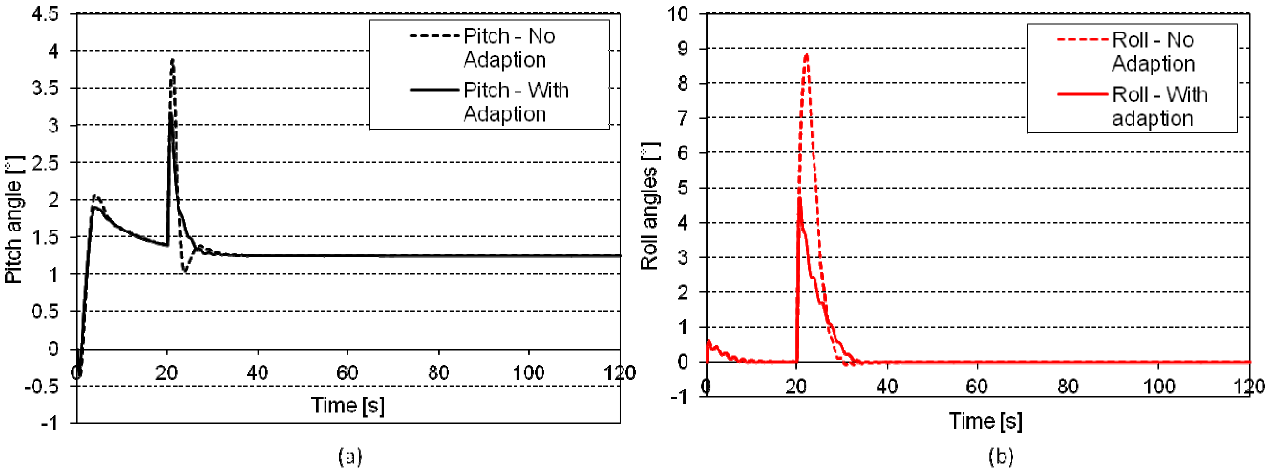


Figure 5.25: Simulated time history of pitch (a) and bank angles (b) (right aileron 50% hard over failure causing rolling moment to the right) in level flight

A pitch error was also induced by the failed elevator, as seen in Figure 5.25 (b), and it was observed that the simulation with the adaption algorithm enabled performed better than the simulation without the adaption algorithm. There was a smaller induced error and a faster, less oscillatory return to the set point value. It was noted in Figure 5.26 (a) that, like many of the other failures, there was a rise in proportional gain to account for the increasing pitch error. The proportional gain was then reduced, as the pitch error reduced. Little change was observed for the integral and derivative gains.

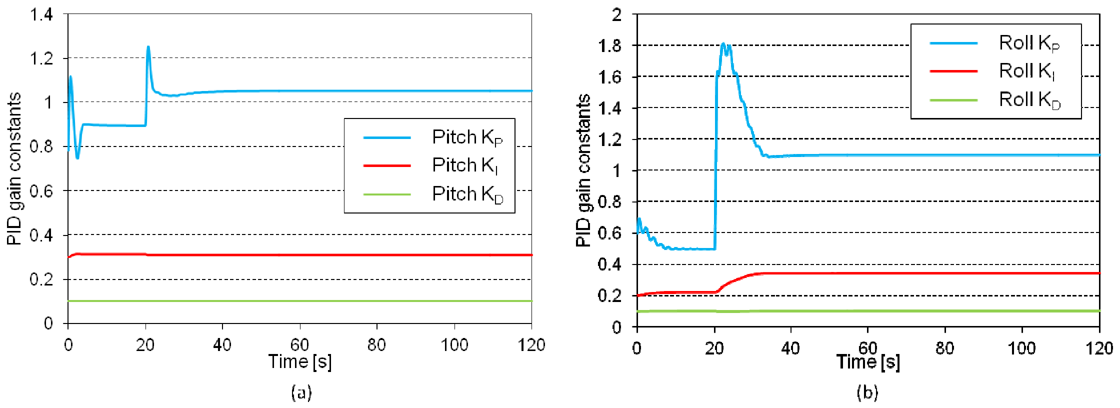


Figure 5.26: Simulated time history of the pitch (a) and roll (b) PID constants with the adaption algorithm enabled (right aileron 50% hard over failure causing a rolling moment to the right) in level flight

Figure 5.27 (b) shows that the right aileron moves to 7.5° and remains at that position until the end of the simulation while the left aileron moves to compensate the induced rolling moment and achieves an angle slightly higher than the right aileron. This is due to the trim offset due to the asymmetries of the wind tunnel model. A small rudder input, shown in Figure 5.27 (c), is also generated due to the feedforward aileron rudder mix used in the autopilot. Of interest is Figure 5.27 (a) where a change in elevator angle is observed as a new trim position has been created with the deflection of the ailerons. No flap deflection was generated as shown in Figure 5.27 (d). It is noted that the autopilot was able to accommodate this failure but the control performance was generally enhanced by the use of the adaption algorithm.

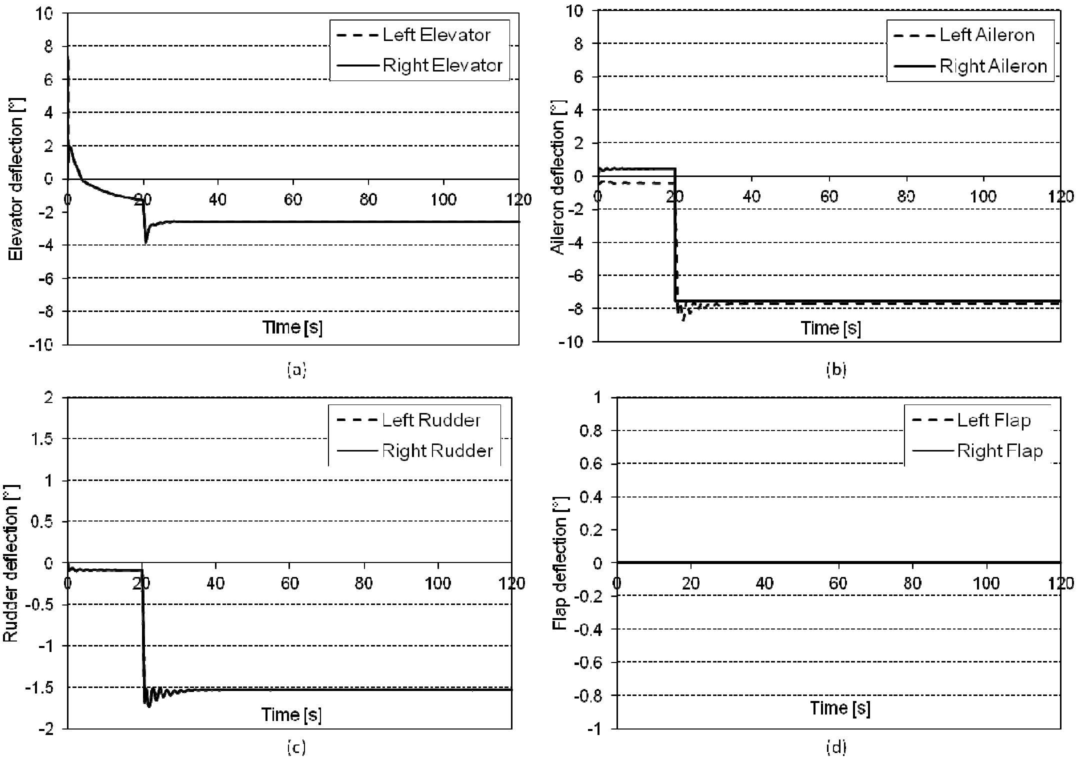


Figure 5.27: Simulated time history of the elevator (a), aileron (b), rudder (c) and flap (d) deflections with the adaption algorithm enabled (right aileron 50% hard over failure causing a rolling moment to the right) in level flight

The simulation was rerun but after having commanded a constant bank angle turn. This resulted in the time history shown in Figure 5.28 which shows that the aircraft, after achieving a steady bank angle of 20°, increased its bank angle at the moment of the induced failure. Without the adaption algorithm the maximum bank angle reached was approximately 28° whereas the maximum bank angle with the adaption algorithm was approximately 24°, a reduction in bank angle error of 50%. It was noted, however, that the adaption algorithm increased the time to return the bank angle to the set point value by a considerable amount. Thus there is a trade off between a quicker response and reduced initial error. A large number of pitch oscillations were observed for both the case with and without adaption and, although the pitch oscillations were slightly reduced with the adaption algorithm, the difference would be likely to be unobservable in flight.

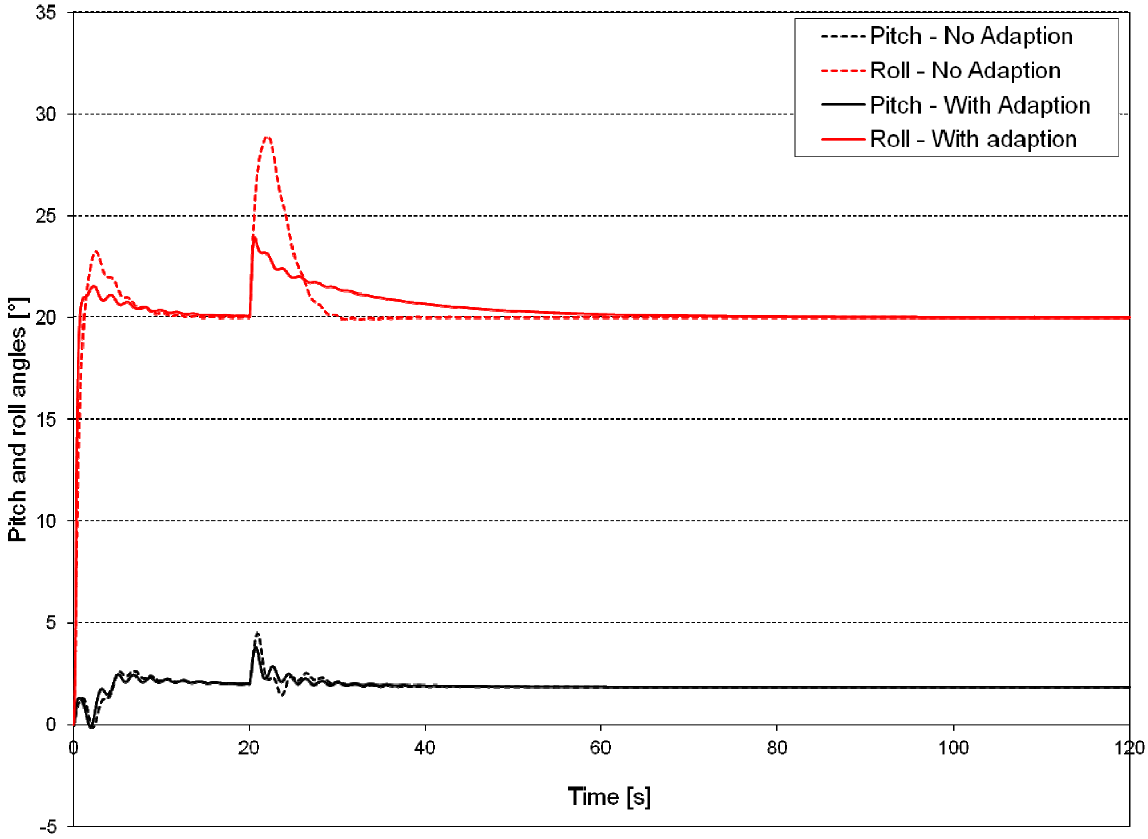
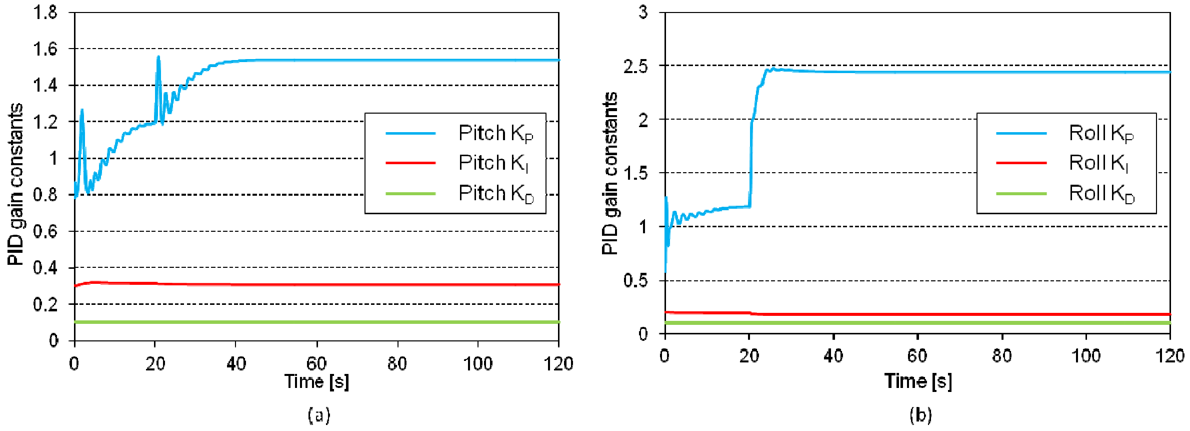


Figure 5.28: Simulated time history of pitch and bank angles (right aileron 50% hard over failure causing a rolling moment to the right) during constant banked flight

Figure 5.29 shows the PID gain changes made by the adaption algorithm and shows the expected trend of an increase in proportional gain with little other change in other gain constants. Of note was the oscillatory behaviour induced in Figure 5.29 (a).



(a)

(b)

Figure 5.29: Simulated time history of the pitch (a) and roll (b) PID constants with the adaption algorithm enabled (right aileron 50% hard over failure causing rolling moment to the right) during constant banked flight

Figure 5.30 (a) to (d), which shows the control deflections of the simulation with the adaption algorithm enabled. It shows a similar response to that of Figure 5.27; however, the elevator response

(Figure 5.30 (a)) is more oscillatory. This is unexpected but explains the oscillations in the pitch time history. The cause of the elevator oscillation, stem from the oscillation seen in the pitch PID proportional gain change; however, the cause of the oscillatory behaviour in the proportional gain is as yet unknown.

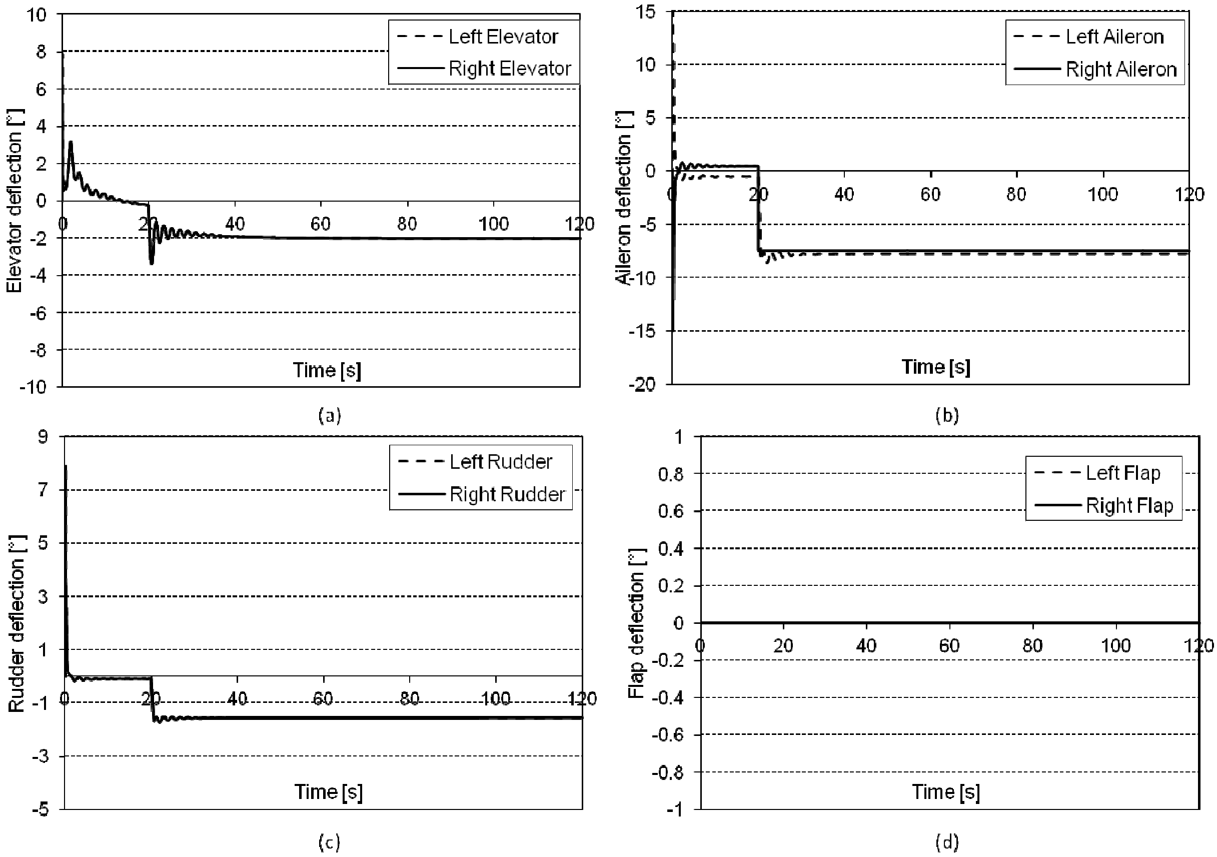


Figure 5.30: Simulated time history of the elevator (a), aileron (b), rudder (c) and flap (d) deflections with the adaption algorithm enabled (right aileron 50% hard over failure causing a rolling moment to the right) during constant banked flight

Right Aileron Failure - 100% Hard Over Failure (Test 7)

The simulation was run for a 100% hard over failure of the right aileron. The failure causes a strong rolling moment to the right with a small nose up pitching moment created due to the change in effective wing camber. Figure 5.31 shows the time history of the pitch and roll angles and shows that, for the case where the adaption algorithm was disabled, there was a large roll error induced. This was then reduced with an overshoot in the return.

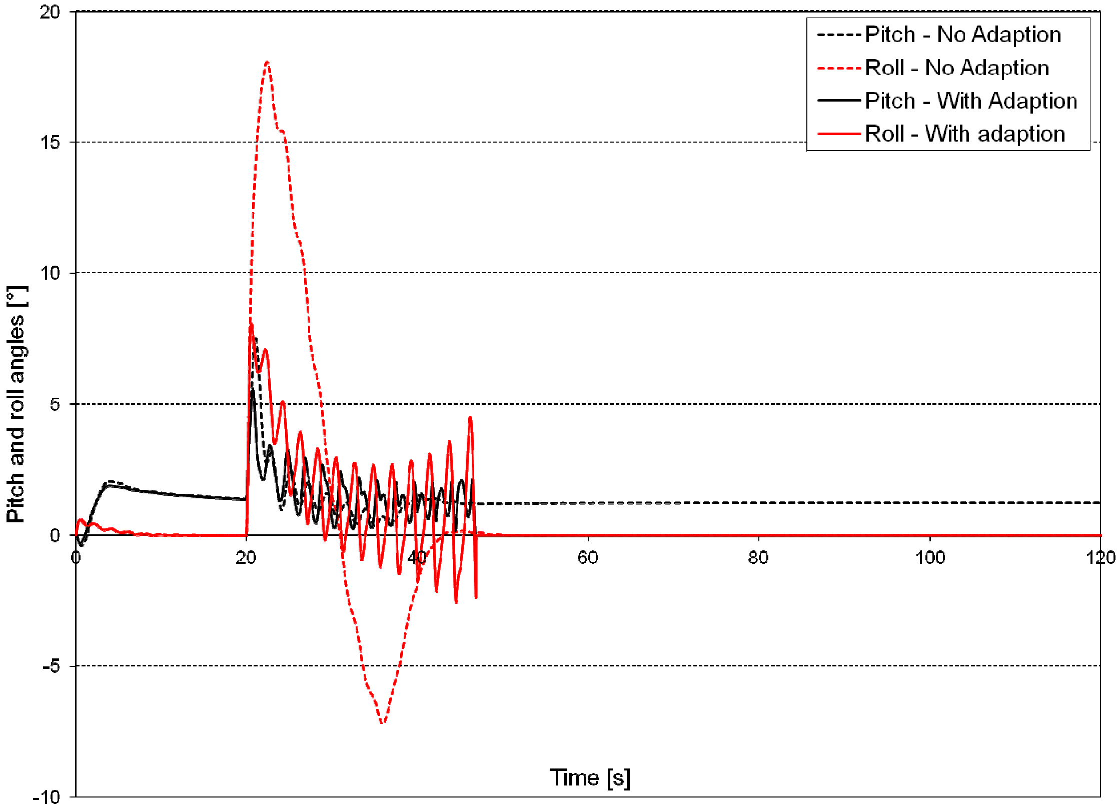


Figure 5.31: Simulated time history of pitch and bank angles (right aileron 100% hard over failure causing rolling moment to the right) in level flight

The ability of the autopilot to tolerate the failure was surprising as it was anticipated that the roll rate might be stopped but that the bank angle would not be returned to a level attitude. It is hypothesised that the autopilot was able to level the wings of the aircraft because of the rudder mixing used. The control authority in the roll sense is supplemented by this rudder mix by the addition of rudder deflection that aids in returning the wings to a level attitude. Of concern was the failure of the simulation with the adaption algorithm enabled. As can be seen in Figure 5.31, the roll angle undergoes an abrupt oscillatory behaviour and, although the maximum roll error is reduced, the roll control is highly unstable. Eventually at approximately 45 seconds the simulation is stopped as the sideslip angle exceeds 30°. The simulation is stopped automatically by the simulation code as the wind tunnel data is only available up to a sideslip angle of 30°. Once this angle is surpassed, an extrapolation of the data would be required. This would be unwise as the aerodynamic effects at these large angles are highly nonlinear. The cause of this instability can be determined by examining Figure 5.32 (b). In the figure it can be seen that there is a rapid but oscillatory increase in proportional gain to the maximum allowable gain of 10. This gain is far too high and causes the roll instability. However, of great concern are the large sideslip angles induced. These large sideslip angles were generated by the rudder feedforward loop. Examining Figure 5.33 (a) to (c), one can see that all of the control surfaces undergo large oscillations. The rudder deflections, in particular, are fairly large and oscillate creating a dynamically increasing sideslip angle as shown in Figure 5.34.

The high rudder deflections seen are as a result of the increase in commanded roll. Currently the limit on aileron command is set by the "Clip to" function described in Section 2.10.2. Thus the PID loops output an ever increasing aileron output command which is fed into the rudder feedforward loop to the point that the rudder induces large and unstable sideslip oscillations. Surprisingly, the pitch behaviour is not as extreme as the roll behaviour but is still oscillatory. This may also be attributed to the rudder deflections causing a change in the aerodynamic characteristics of pitching moment and lift.

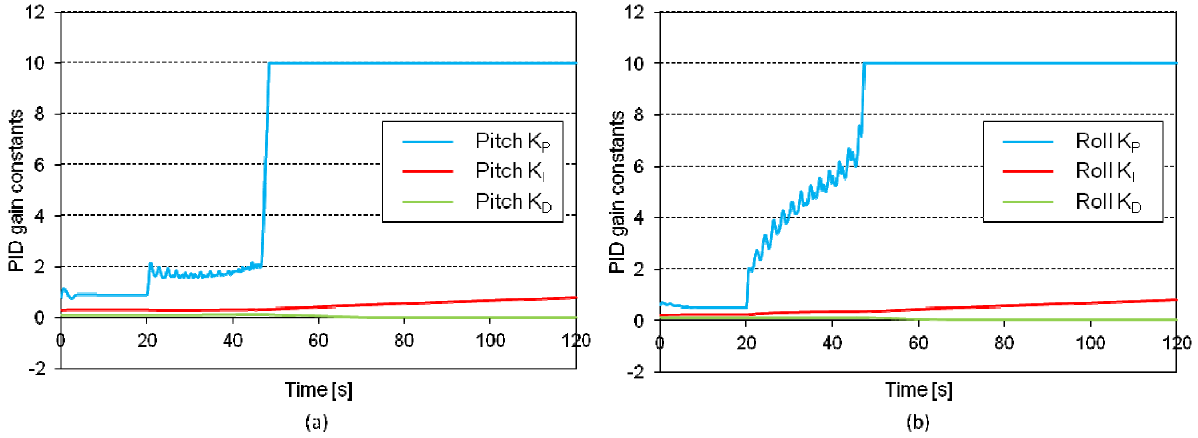


Figure 5.32: Simulated time history of the pitch (a) and roll (b) PID constants with the adaption algorithm enabled (right aileron 100% hard over failure causing a rolling moment to the right) in level flight

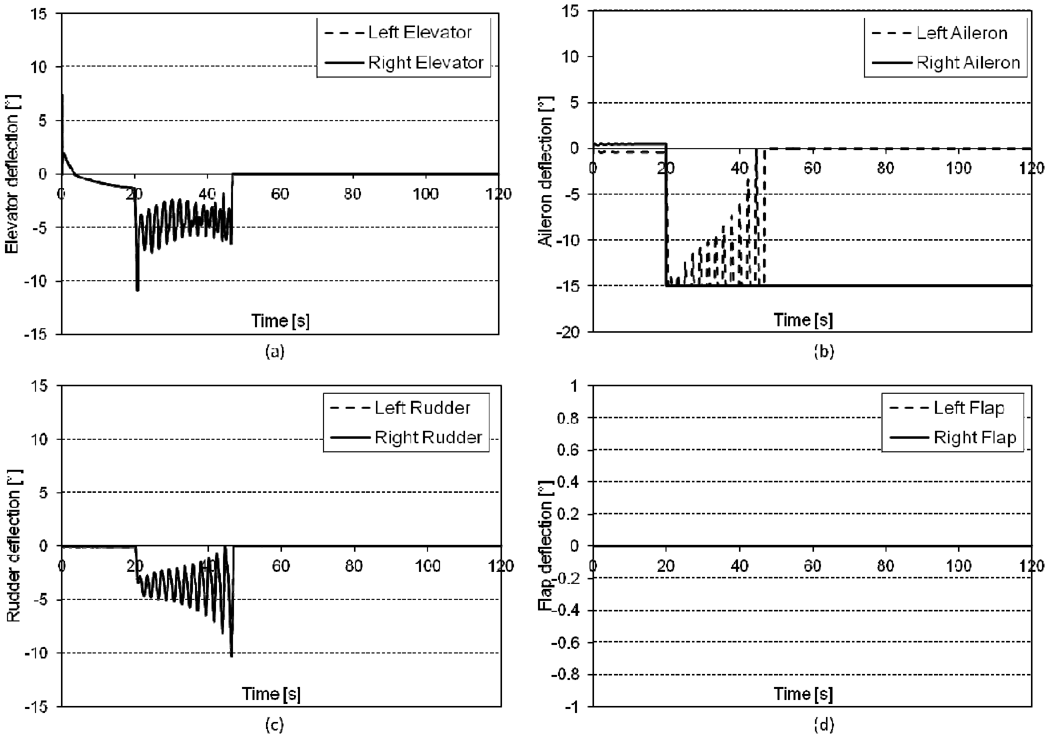


Figure 5.33: Simulated time history of the elevator (a), aileron (b), rudder (c) and flap (d) deflections with adaption algorithm enabled (right aileron 100% hard over failure causing a rolling moment to the right) during constant banked flight

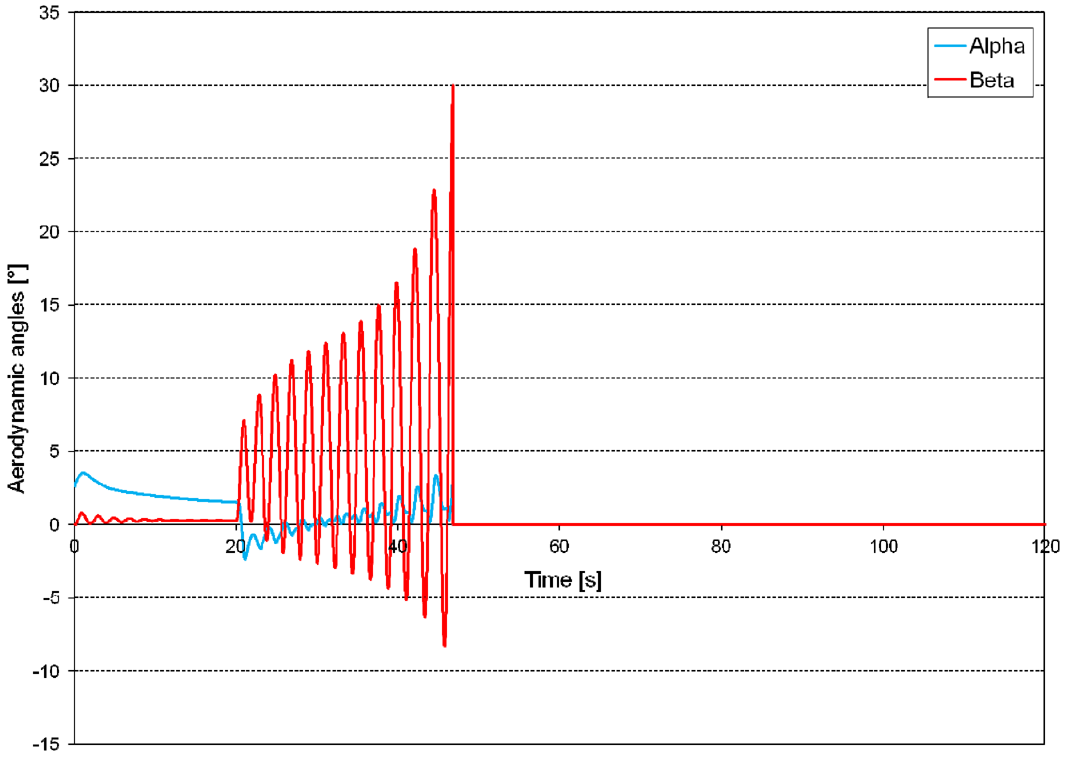


Figure 5.34: Simulated time history of angle of attack and sideslip angle with the adaption algorithm enabled (right aileron 100% hard over failure causing a rolling moment to the right) during constant banked flight

Based on the previous simulation, the rudder-aileron mix was disabled. This test showed that the rudder did indeed create the instabilities seen in the previous simulation. Figure 5.35 shows that the roll angle increases but does not recover as it did previously. In both the case with and without adaption, the angle reaches approximately 70° before recovering slightly to establish a constant bank angle of approximately 67°. The roll angle change was slightly slower with the adaption algorithm enabled. This slightly improved response is a function of the increase in proportional gain (see Figure 5.36 (b)). However, it is also noted that the proportional gain in Figure 5.36 (b) reaches a value of 10 which is the maximum allowable gain set in the program. Thus, left unchecked, this would result in a continuous gain rise. This is as expected and highlights the limitations of an adaption algorithm to cope with this form of failure without some other form of intervention. The aircraft reached an equilibrium bank angle which was not the desired set point and, as such, the adaption algorithm began to increase the integral gain in an attempt to reduce the steady-state error. This obviously was not possible due to insufficient control authority and a runaway condition occurred.

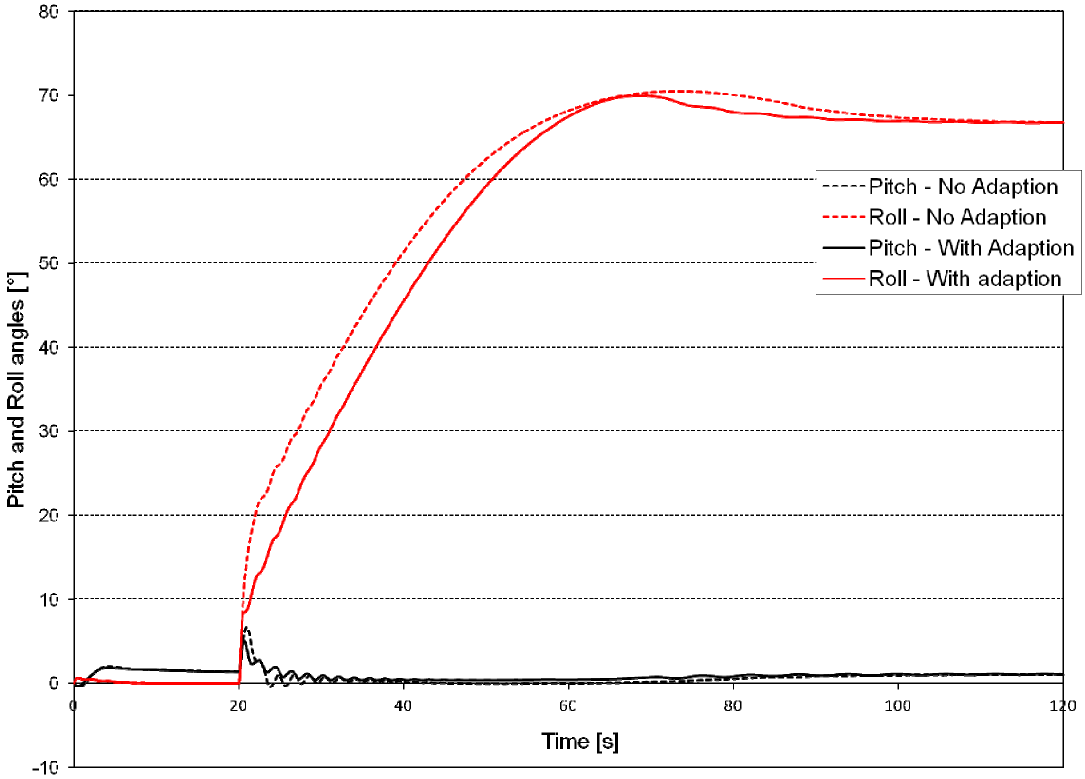


Figure 5.35 : Simulated time history of pitch and bank angles (right aileron 100% hard over failure causing rolling moment to the right) in level flight

The changes in pitch were evident in Figure 5.35 and were relatively minor in comparison to the roll change. However, a small improvement was gained through the use of the adaption algorithm. Of note is the oscillatory behaviour of the proportional gain in Figure 5.36 (a) towards the end of the simulation. This occurs as both elevators reached their maximum deflections, observed in Figure 5.37. The adaption algorithm perceived this to be a steady-state error and subsequently increased the integral gain. Little change was made to the derivative gain in this case.

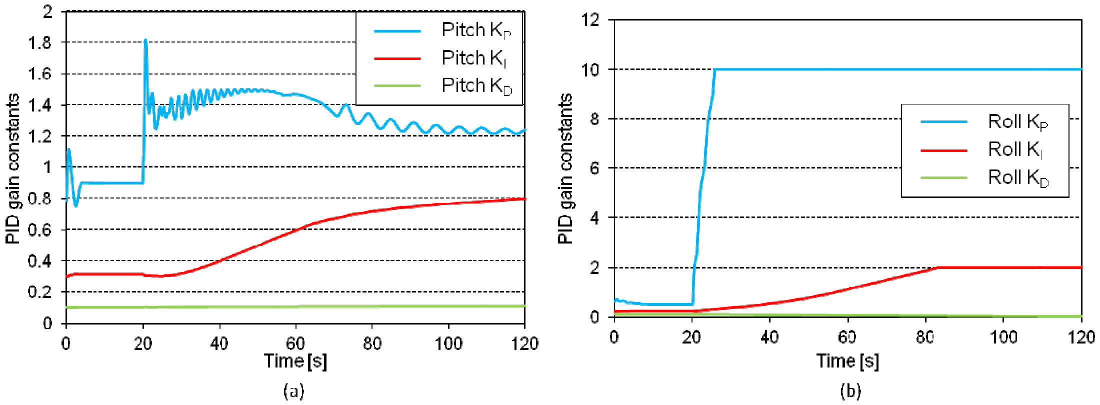


Figure 5.36: Simulated time history of the pitch (a) and roll (b) PID constants with the adaption algorithm enabled (right aileron 100% hard over failure causing a rolling moment to the right) in level flight

Figure 5.37 (a) to (d) shows the time history of the control deflections for the simulation. Figure 5.37 (b), in particular, shows the right aileron failure which is closely followed by left aileron deflection in an attempt to prevent further bank angle. This was unachievable as no more control authority in the roll direction existed. Hence, the aircraft begins an ever increasing spiral. The elevator increased to full deflection, shown in Figure 5.37 (a), to maintain the required pitch angle but is ultimately unsuccessful.

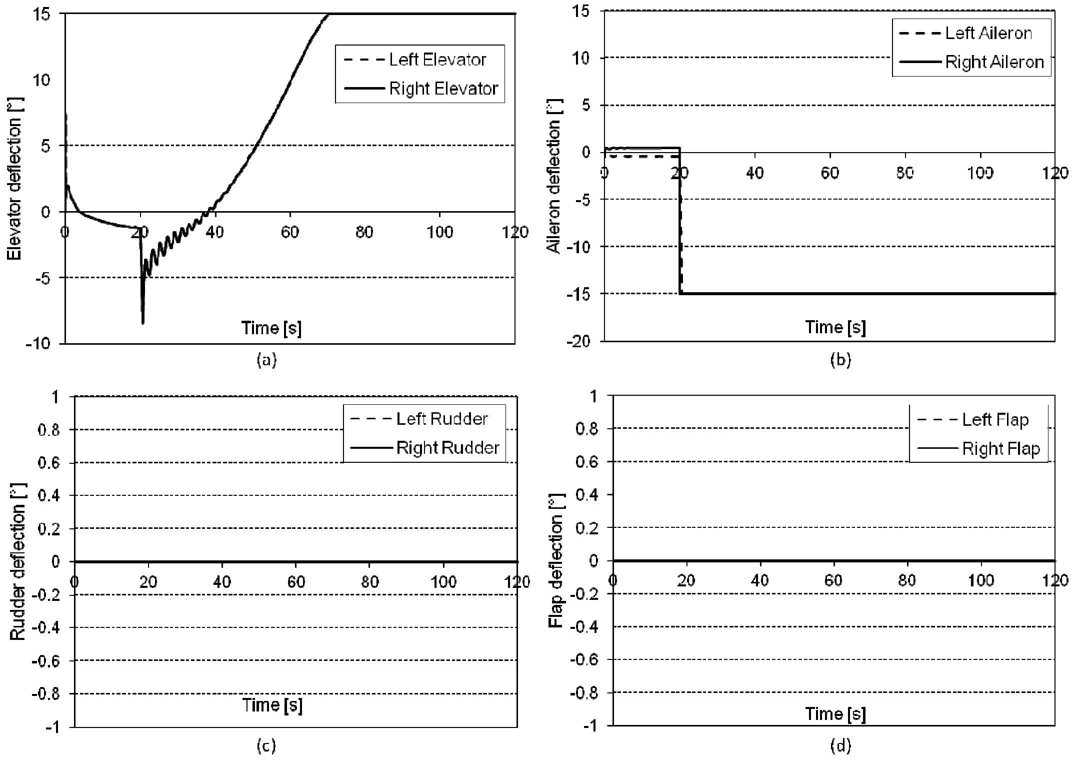


Figure 5.37: Simulated time history of the elevator (a), aileron (b), rudder (c) and flap (d) deflections with the adaption algorithm enabled (right aileron 100% hard over failure causing a rolling moment to the right) in level flight

The results for aileron failure in the banked flight case are not presented as they were found to be almost identical to the level flight case, as insufficient roll authority was unable to return the aircraft to the correct bank angle; however they can be found in Appendix C.

Right Aileron Failure - 100% Hard Over (Control Allocation Enabled - Rudder Mix Enabled) (Test 8)

This simulation was re-run with control allocation enabled. This yielded in a very different result from those of Figure 5.31 to Figure 5.34. In the previous test, with the control adaption algorithm enabled, there was a rapid rise in proportional gain which led to an unstable roll oscillation. This was exacerbated by the aileron-rudder mix that caused large sideslip angles, causing the flight dynamic model to terminate prematurely. In this case, however, this instability has not been created. Figure 5.38 (a) and (b) show that there was an improvement in control performance and that for both the non adapted and adapted cases, there was successful re-establishment of level flight, the adapted case having less of an overshoot in both roll and pitch.

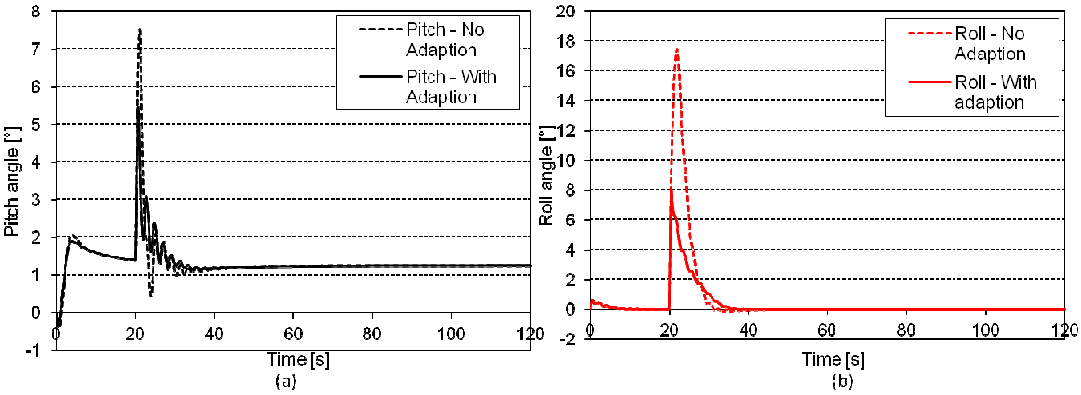


Figure 5.38: Simulated time history of pitch (a) and roll angles (b) (right aileron 100% hard over failure causing rolling moment to the right) in level flight with control allocation enabled

In Figure 5.32, it was observed that there was rapid increase in both pitch and roll PID gains ultimately leading to a runaway condition that caused the oscillatory behaviour and the failure of the simulation. In this simulation, this did not occur, as shown in Figure 5.39. The figure shows a steady rise in the pitch proportional gain closely followed by a large number of oscillations that ultimately decrease the proportional gain to a steady level. The roll proportional gain undergoes a rapid rise followed by a steady decrease, again, until a steady value is attained. There was also an increase in the roll integral gain that attempted to decrease the steady-state error more quickly; however, it is theorised that this was overshadowed by the decrease in proportional gain.

Figure 5.40 (a) to (d) shows the reason for the lack of build up of PID gains. In Figure 5.40 (d), it can be seen that there are momentary applications of differential flap that aid the ailerons in restoring the aircraft to a level attitude. This prevents the build up in commanded aileron input and hence the build up in rudder deflection. As a result, the sideslip angles do not approach the large angles seen in Figure 5.34 but instead remain reasonably small, as seen in Figure 5.41. It can be seen that there is now a sideslip angle offset that is generated by the rudder deflection. Of concern was the observed chattering of the sideslip angle. In essence the controller is reducing the inherent directional stability of the aircraft which is highly undesirable. It is thought that this would improve by removing the Rudder to aileron mixing and controlling the rudder without interference from the aileron command.

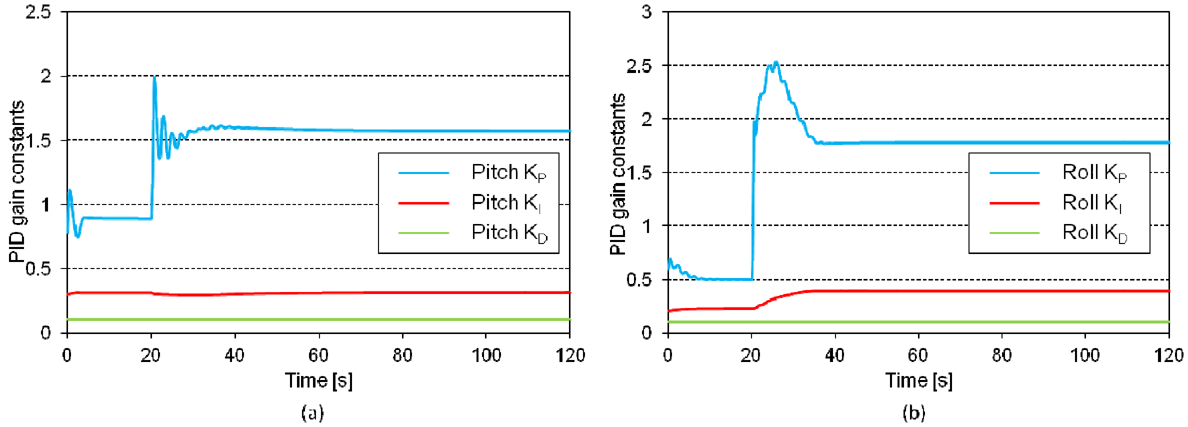


Figure 5.39: Simulated time history of the pitch (a) and roll (b) PID constants with the adaption algorithm enabled (right aileron 100% hard over failure causing a rolling moment to the right) in level flight with control allocation enabled

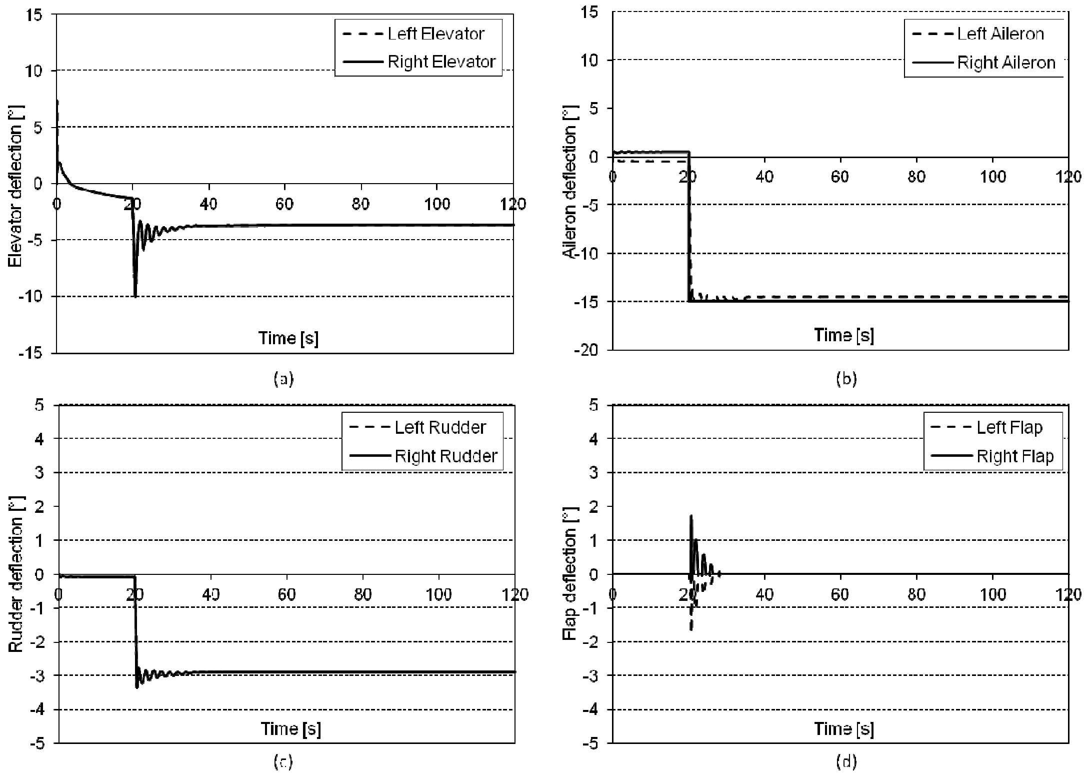


Figure 5.40: Simulated time history of the elevator (a), aileron (b), rudder (c) and flap (d) deflections with the adaption algorithm enabled (right aileron 100% hard over failure causing a rolling moment to the right) in level flight with control allocation enabled

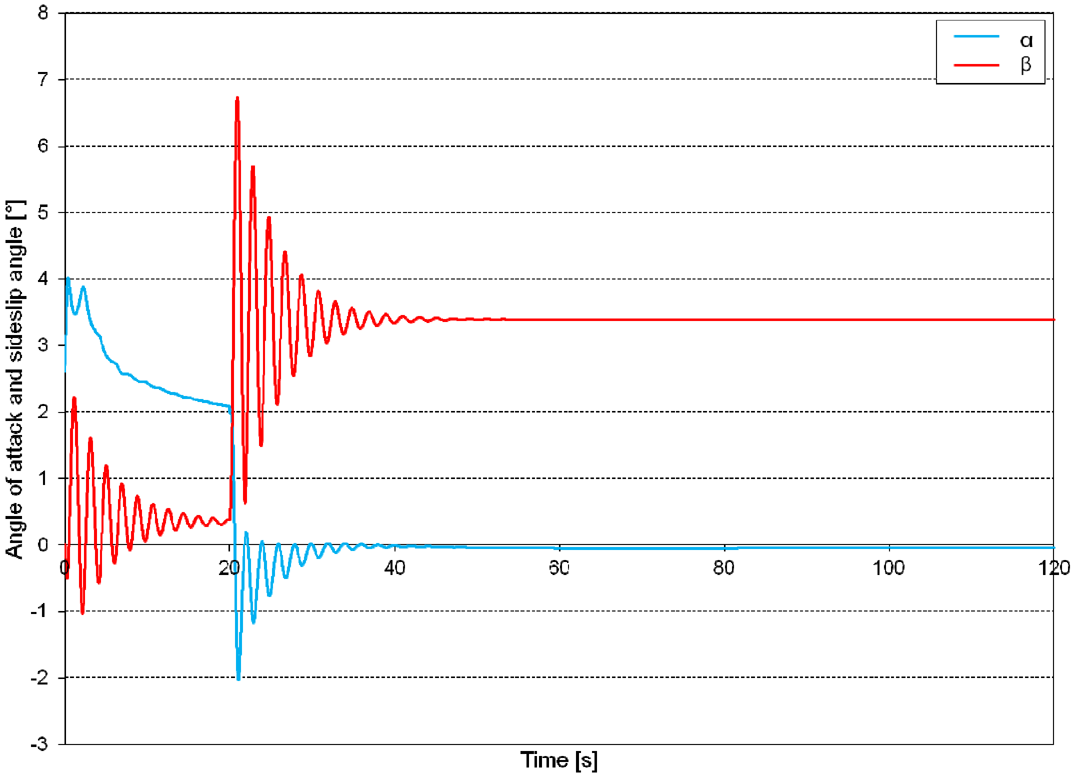


Figure 5.41: Simulated time history of angle of attack and sideslip angle with the adaption algorithm enabled (right aileron 100% hard over failure causing a rolling moment to the right) in level flight with control allocation enabled

The simulation was run for a constant banked turn. Unlike the case without control allocation, the test showed that the autopilot was able to control the aircraft and re-establish the desired set point as shown in Figure 5.42. In the figure, it can be seen that, even without the control adaption, the autopilot is capable of maintaining the required bank angle; however, with the adaption algorithm enabled there is an improvement in the control performance. The maximum error seen was reduced; however, it did take longer to reach the desired bank angle than the non adapted case. Figure 5.43 (a) and (b) show the change in PID gains and show that there is a more pronounced increase in proportional gain that aids in reducing the maximum error. An indication at the cause for the delay in returning the aircraft to the set point value is given in Figure 5.43 (b), which shows very slight reduction in integral gain. This is likely caused by the rule base of the fuzzy logic controller in an attempt to reduce the overshoot and may indicate that a review of the rule base for the adaption algorithm should be undertaken.

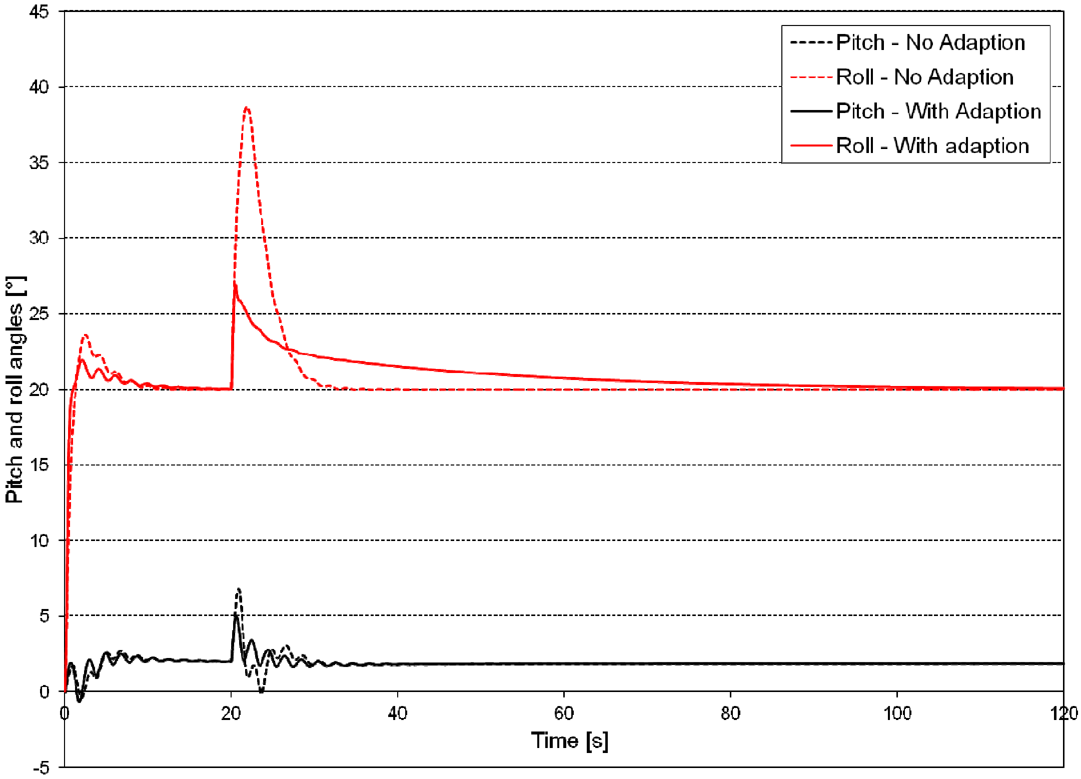


Figure 5.42: Simulated time history of pitch and bank angles (right aileron 100% hard over failure causing rolling moment to the right) during a constantly banked flight with control allocation enabled

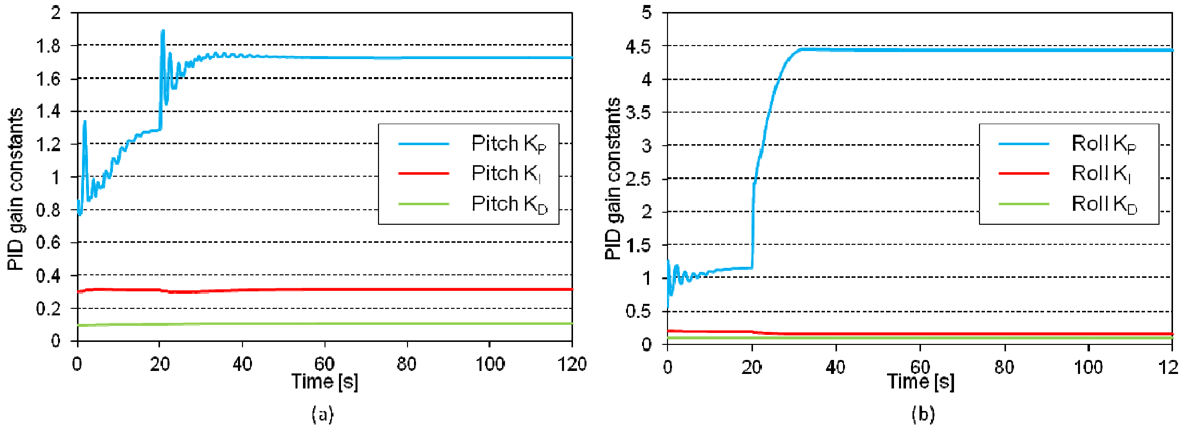


Figure 5.43: Simulated time history of the pitch (a) and roll (b) PID constants with the adaption algorithm enabled (right aileron 100% hard over failure causing a rolling moment to the right) in level flight with control allocation enabled

The simulation for banked flight was re-run with the rudder mix disabled as this was the only way a full result could be attained for the case without control allocation. The previous results, shown in Figure 5.35 to Figure 5.37, showed that the aircraft was unable to achieve level flight although the rolling moment was neutralised. The flight path was a tight descending spiral that was uncontrollable. After the control allocation algorithm was enabled, a different result was obtained. The result showed that the aircraft was able to return the aircraft to the correct bank angle even without the rudder mix. Figure 5.44, which shows the pitch and roll angle time histories, shows a similar result to those obtained in Figure 5.42. However, a subtle difference is noted in that the oscillations seen after the initial failure are less prevalent. The primary difference is however observed when examining Figure 5.45 and comparing it to Figure 5.41. In Figure 5.45, the sideslip angle returns to the initial trim position, albeit with a small oscillation about the trim position, whereas when the rudder mix included, as in Figure 5.41, the sideslip angle finds a new trim position. Hence, the aircraft is in a crabbing condition which, while not unmanageable, is undesirable.

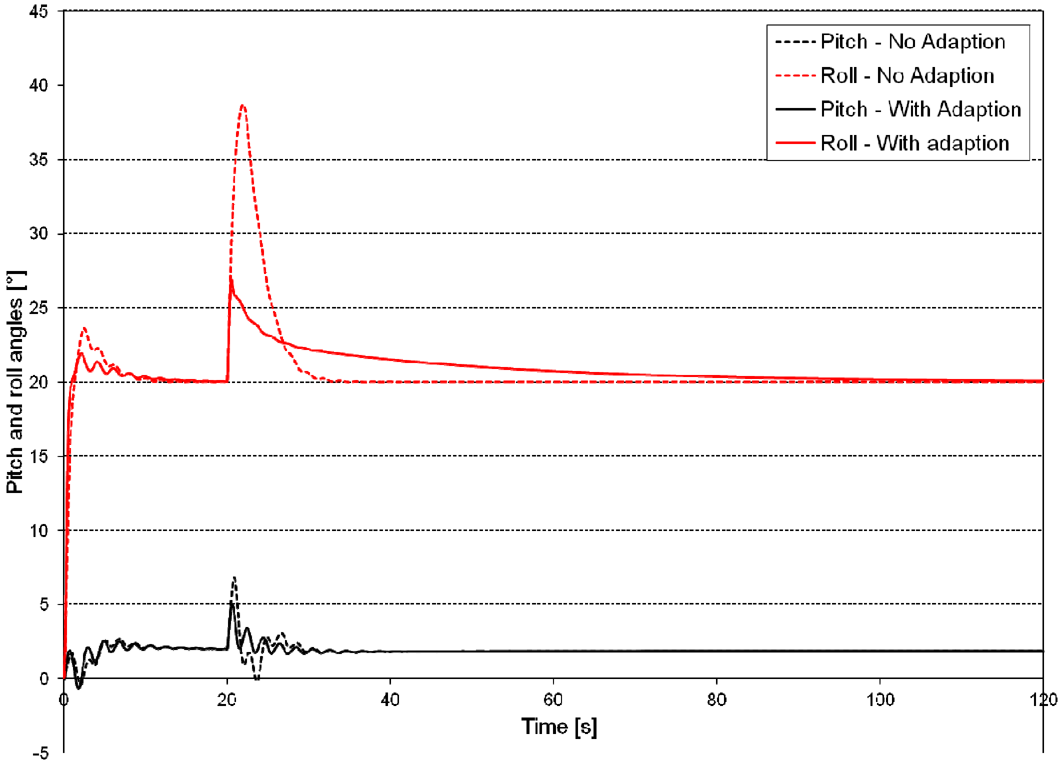


Figure 5.44: Simulated time history of pitch and bank angles (right aileron 100% hard over failure causing rolling moment to the right) during a constantly banked flight with control allocation enabled but without the aileron rudder mixing

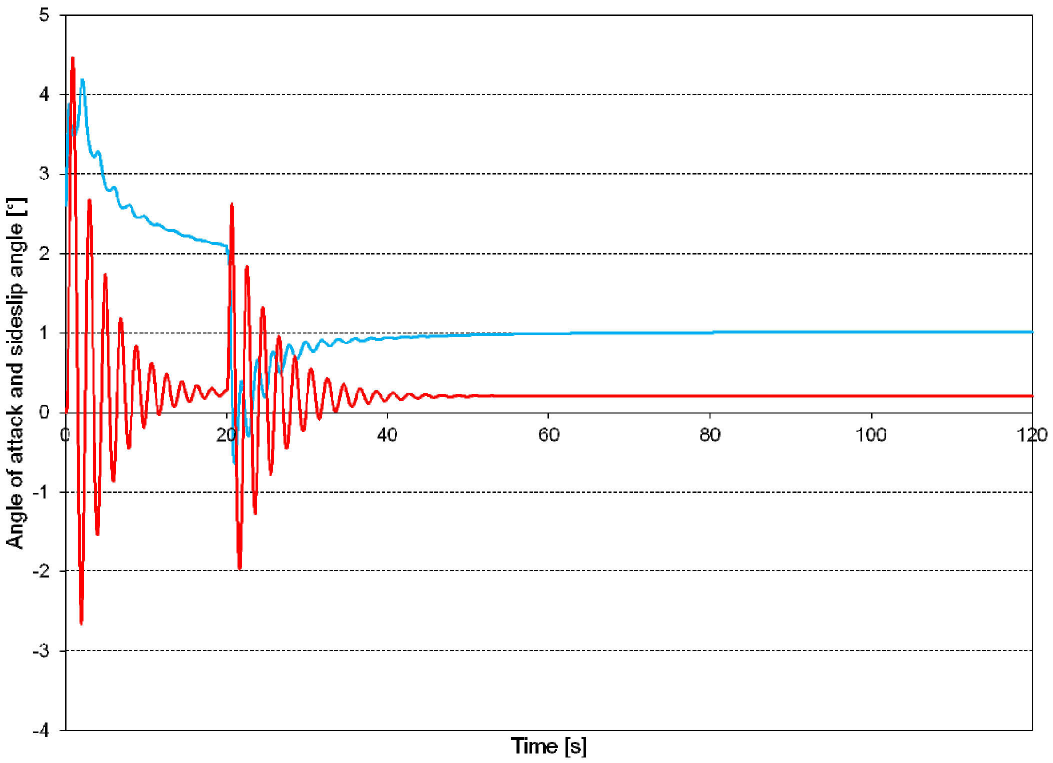


Figure 5.45: Simulated time history of angle of attack and sideslip angle with the adaption algorithm enabled (right aileron 100% hard over failure causing a rolling moment to the right) in banked flight with control allocation enabled with no rudder mix

Elevator and aileron combined failure - 0° deflection (Test 9)

The 0° deflection failure was again induced but in this case failing both the right aileron and right elevator. The result of this simulation was a combination of the two previous cases: there was a change in both pitch and roll as seen in Figure 5.46. Similarly to the cases above, the aircraft was returned to the desired set point without the aid of the adaption algorithm; however, the performance was improved by the adaption algorithm. Again, the proportional gain for the pitch PID rose sharply with a slight decrease to a stable level as shown in Figure 5.47 (a). The response to multiple control surface failure was almost identical to the case where the elevator alone was failed. The roll PID followed the same trends as the aileron failure case; however, the magnitudes of the gains differed slightly. It is observed that the autopilot was tolerant to this failure but improved performance was shown when the adaption algorithm was enabled.

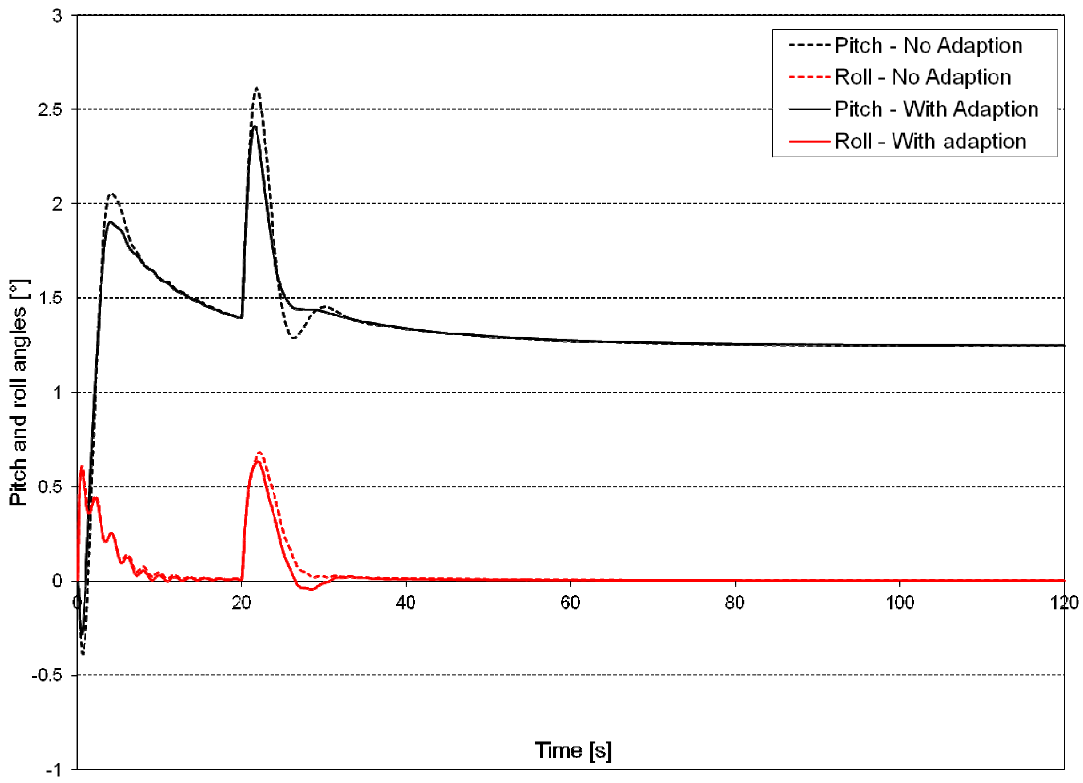


Figure 5.46: Simulated time history of pitch and bank angles (0° right elevator and right aileron deflection failure) in level flight

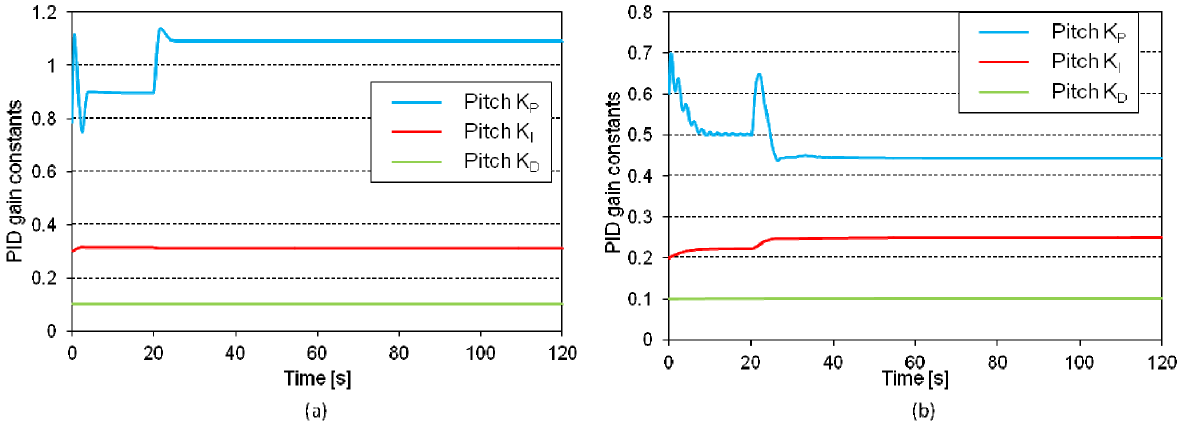


Figure 5.47: Simulated time history of the pitch (a) and roll (b) PID constants with the adaption algorithm enabled (0° right elevator and right aileron deflection failure) in level flight

Little difference was observed for the banked flight case, and these results, can be found in Appendix C.

Elevator and aileron combined failure - 50% hard over (Test 10)

The simulation was run, but with a 50% hard over failure of both the right aileron and right elevator. Figure 5.48 (a) and (b) shows the time history of the pitch and roll angles for the level flight case and shows a large pitch and roll disturbance. The pitch disturbance (Figure 5.48 (a)) was found to be larger than the elevator failure case (11.5° vs. 7°) for the case without the adaption algorithm running. There are also a number of oscillations that occurred before reaching the set point value. The roll angle (Figure 5.48 (b)) shows that the adaption algorithm was able to curb the maximum bank angle disturbance by approximately 50%; however, the return to level flight was more gradual than the case without the adaption algorithm.

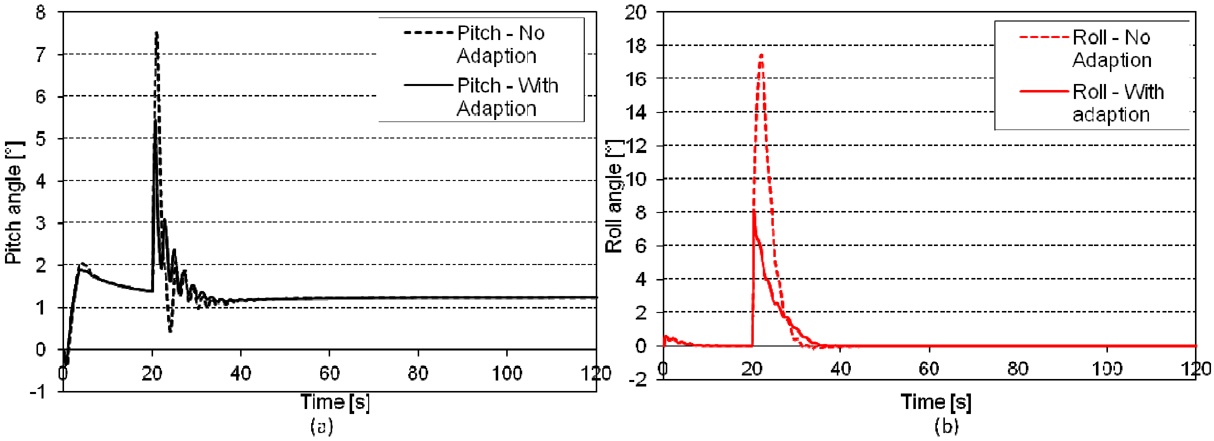
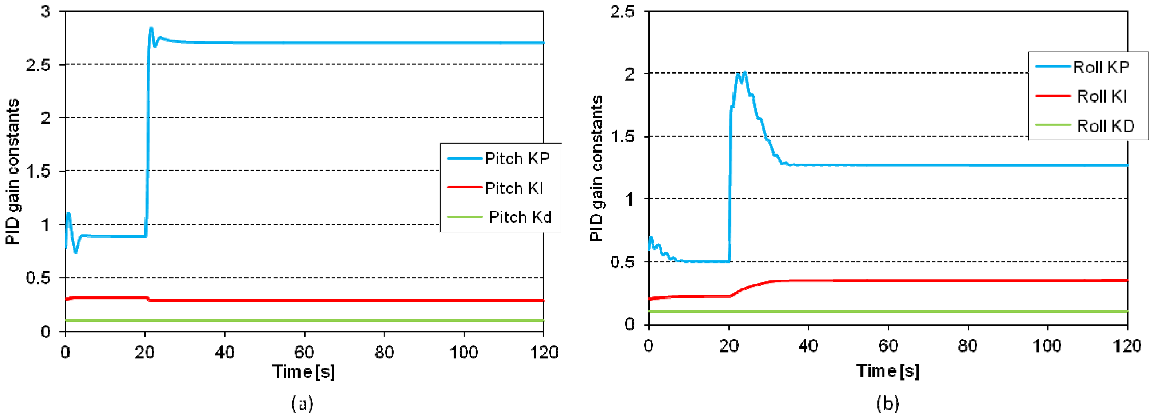


Figure 5.48: Simulated time history of pitch (a) and bank angles (b) (right elevator and aileron 50% hard over failure causing pitching moment in the nose up direction and a rolling moment to the right) in level flight

Figure 5.49 (a) and (b) show the changes in PID gains for the pitch and roll PID loops and show that, for the pitch PID the proportional gain rose sharply with a slight oscillation down to a marginally lower value. This was then stable for the rest of the simulation; however, this was the largest change yet seen in the simulations. The integral and derivatives gains were hardly affected. The roll proportional gain was also increased rapidly but reduced to a greater extent after the initial disturbance. This may account for the ability of the system to curb the maximum bank angle error but return more slowly to the set point. An increase in the integral gain was also noted.

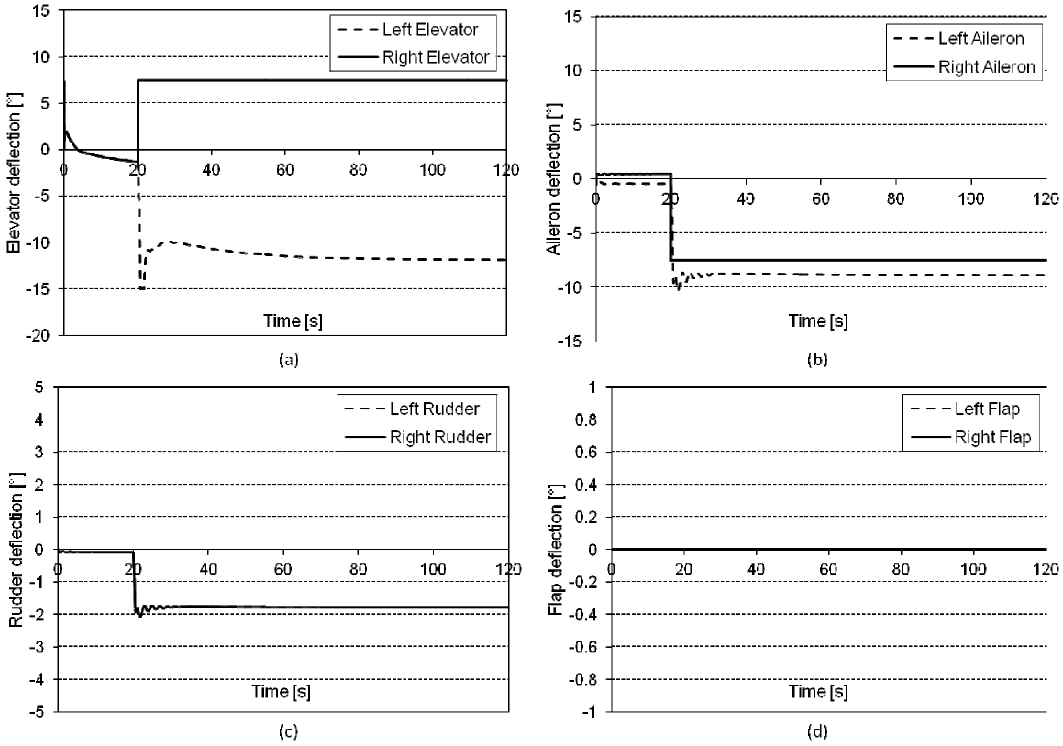


(a)

(b)

Figure 5.49: Simulated time history of the pitch (a) and roll (b) PID constants with the adaption algorithm enabled (right elevator and aileron 50% hard over failure causing a nose up pitching moment and a rolling moment to the right) in level flight

Figure 5.50 (a) to (d) shows the control deflections during the simulation and shows the point at which the elevator and ailerons failed. It also shows a slight oscillatory response in the remaining aileron with a positive elevator corrective action. Again the presence of the rudder deflection stems from the feedforward rudder-aileron mix.



(a)

(b)

(c)

(d)

Figure 5.50:Simulated time history of the elevator (a), aileron (b), rudder (c) and flap (d) deflections with the adaption algorithm enabled (right elevator and aileron 50% hard over failure causing nose up pitching moment and a rolling moment to the right) in level flight

The simulation was run again this time in the banked condition. This yielded a different set of results, shown in Figure 5.51, but with similar trends to those of Figure 5.48 (a) and 9B). The pitch without adaption reached a higher maximum error and again underwent a number of oscillations before returning to the set point value. The pitch with the adaption showed a lower maximum pitch error but for this case there were a number of small oscillations induced in the return to the set point value. The cause of this can be seen in Figure 5.52 (a) where a number of oscillations in the proportional gain are evident. The results for bank angle show the ability of the adaption algorithm to induce a faster corrective action; however, Figure 5.51, shows more clearly the delay in the return of the bank angle to the desired set point value. This is attributed to the slight reduction in integral gain as seen in Figure 5.52 (b). The figure also shows a large increase in proportional gain with no subsequent reduction, as seen in the other simulations.

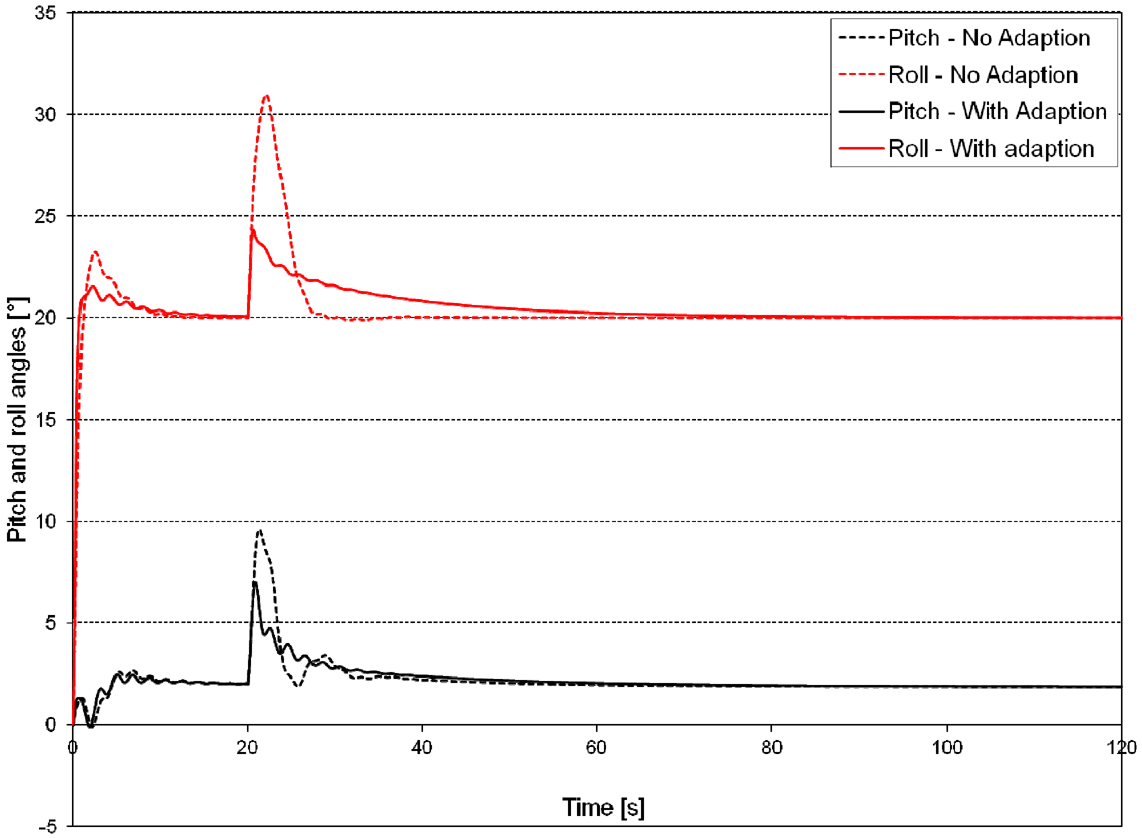


Figure 5.51: Simulated time history of pitch and bank angles (right elevator and aileron 50% hard over failure causing pitching moment in the nose up direction and a rolling moment to the right) during constant banked flight

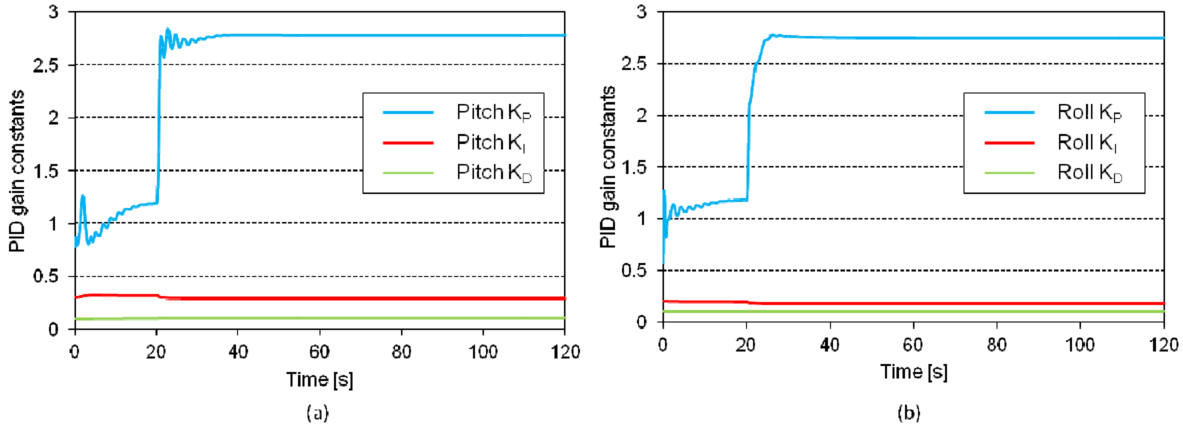


Figure 5.52: Simulated time history of the pitch (a) and roll (b) PID constants with the adaption algorithm enabled (right elevator and aileron 50% hard over failure causing a nose up pitching moment and a rolling moment to the right) during constant banked flight

Elevator and aileron combined failure - 100% hard over (Test 11)

The simulation was run on this occasion inducing a 100% control deflection failure for both the right aileron and right elevator. A similar trend to the previous 100% hard over failure cases was noted in that the autopilot was unable to re-establish control with the adaption algorithm enabled or disabled, as shown in Figure 5.53. It was noted that the case with the adaption algorithm enabled was less oscillatory and, in the case of pitch, the initial response was very much better than the case without the adaption algorithm.

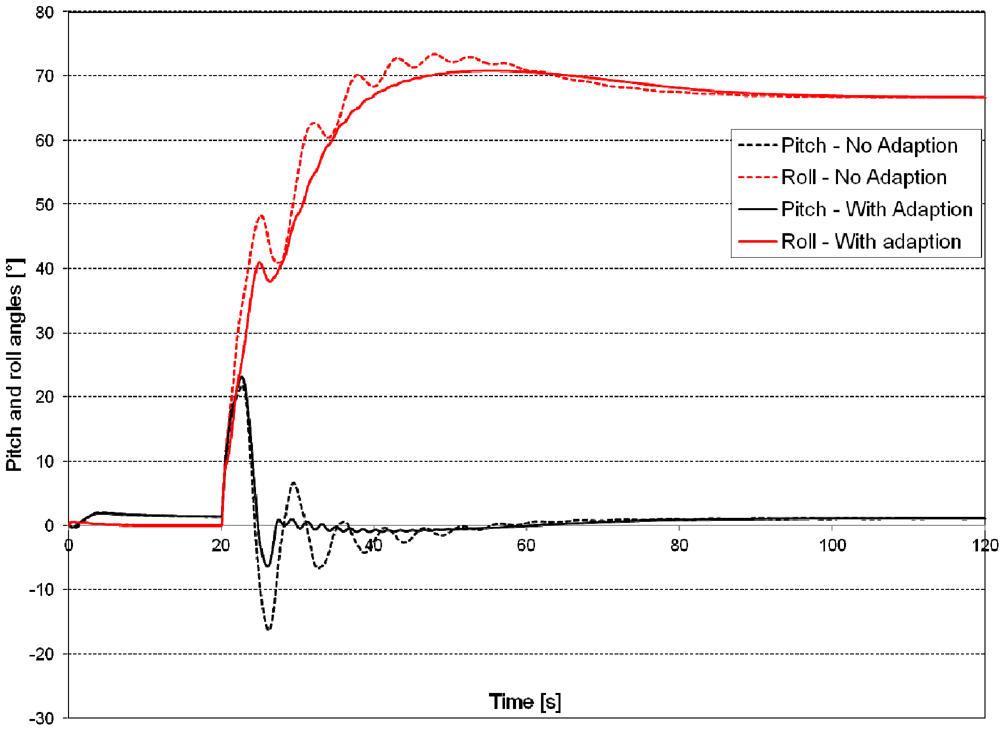


Figure 5.53: Simulated time history of pitch and bank angles (right elevator and aileron 100% hard over failure causing pitching moment in the nose up direction and a rolling moment to the right) in level flight

Figure 5.54 shows the rapid increase of proportional gain for the pitch and roll PID controllers. This is in response to the insufficient roll authority. Once the roll angle has been stopped, it can be seen that the adaption algorithm perceives this as a steady-state error and tries to increase the roll integral gain. However, this is a futile attempt to remedy the situation and the resulting flight path is a tight spiral dive. Figure 5.55 (a) interestingly shows a number of aggressive elevator oscillations that eventually reach the maximum allowable elevator deflection. This corresponds to the rapid pitch oscillations seen in Figure 5.53. Once the remaining elevator saturates, the pitch response is that of the natural dynamics of the aircraft.

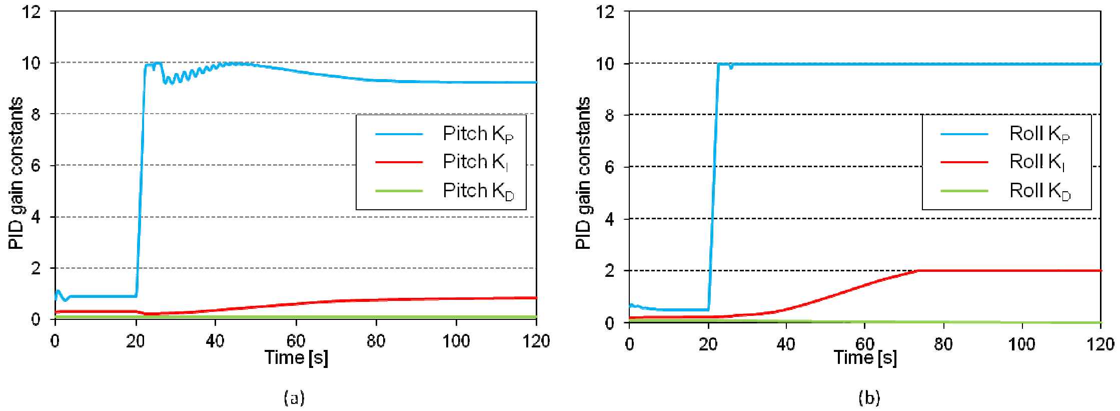


Figure 5.54: Simulated time history of the pitch (a) and roll (b) PID constants with the adaption algorithm enabled (right elevator and aileron 100% hard over failure causing a nose up pitching moment and a rolling moment to the right) in level flight

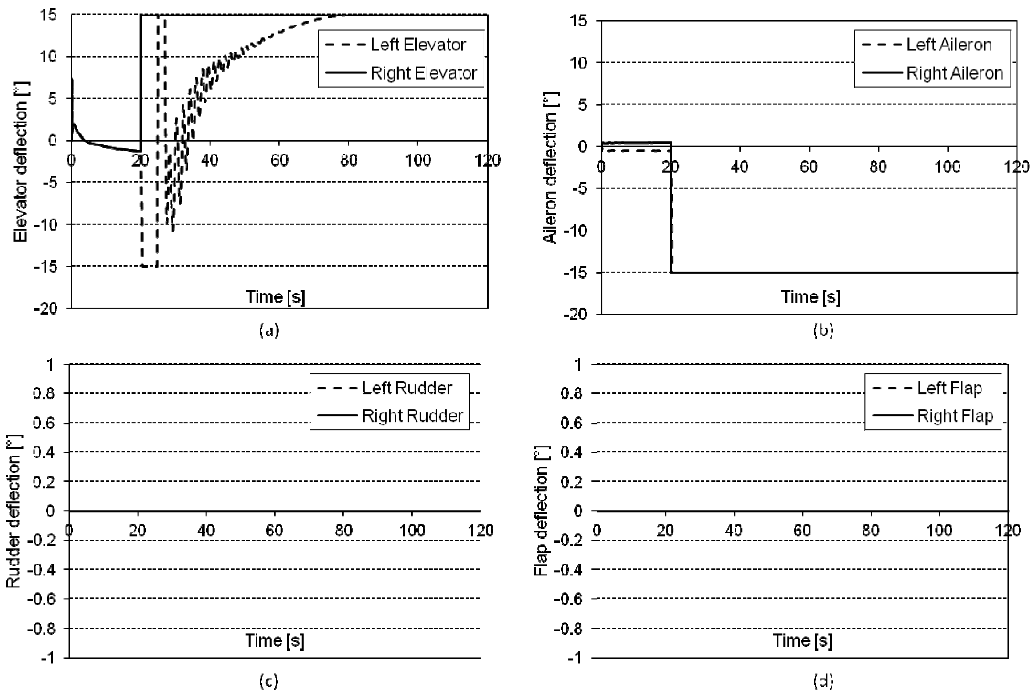


Figure 5.55: Simulated time history of the elevator (a), aileron (b), rudder (c) and flap (d) deflections with adaption algorithm enabled (right elevator and aileron 50% hard over failure causing nose up pitching moment and a rolling moment to the right) in level flight

A similar result was obtained after the simulation was re-run with a constant bank turn set as the manoeuvre to fly. The results provide little insight into the behaviour of the adaption algorithm and, as such, are not presented here. Suffice to say that the resulting flight path resulted in the same tight spiral dive seen in the previous simulation, the results of which, if required can be found in Appendix C.

Elevator and aileron combined failure - 100% hard over (control allocation enabled) (Test 12)

The inability of the aircraft to tolerate the 100% hard over failures successfully is unsurprising and was the primary reason that the control allocation algorithm was developed. Accordingly, the 100% hard over failures were repeated with the control allocation algorithm enabled to demonstrate the necessity of having control adaption coupled with control allocation. Figure 5.56 (a) and (b) shows that the performance is greatly improved with the addition of the allocation algorithm. The bank angle (Figure 5.56 (b)) is easily returned to level flight with and without the adaption algorithm and there was a general improvement in control performance in the case with adaption. However, it is noted that an uncontrolled oscillatory response of the pitch angle (Figure 5.56 (a)) developed. The cause of this is seen in Figure 5.57 (d), which shows the use of flaps in the control of both roll and pitch. The figure shows that the flaps while enough to level the wings, cannot provide the required control authority to maintain the required pitch angle. A slight offset is seen in the flap deflections to maintain the wing-level attitude. The aircraft undergoes a phugoid motion due to the lack of control authority in pitch. The saturation of the flaps and the inability of the system cause the adaption algorithm to increase the pitch proportional gain to its limit; however, the roll PID gains are reasonably well behaved as shown in Figure 5.58.

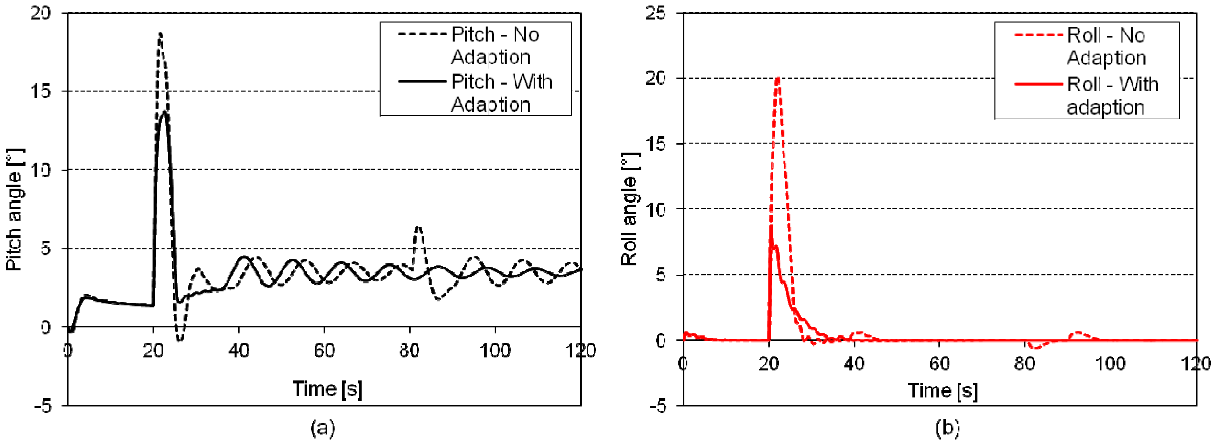


Figure 5.56: Simulated time history of pitch (a) and bank angles (b) (right elevator and aileron 100% hard over failure causing pitching moment in the nose up direction and a rolling moment to the right) in level flight with the control allocation algorithm enabled

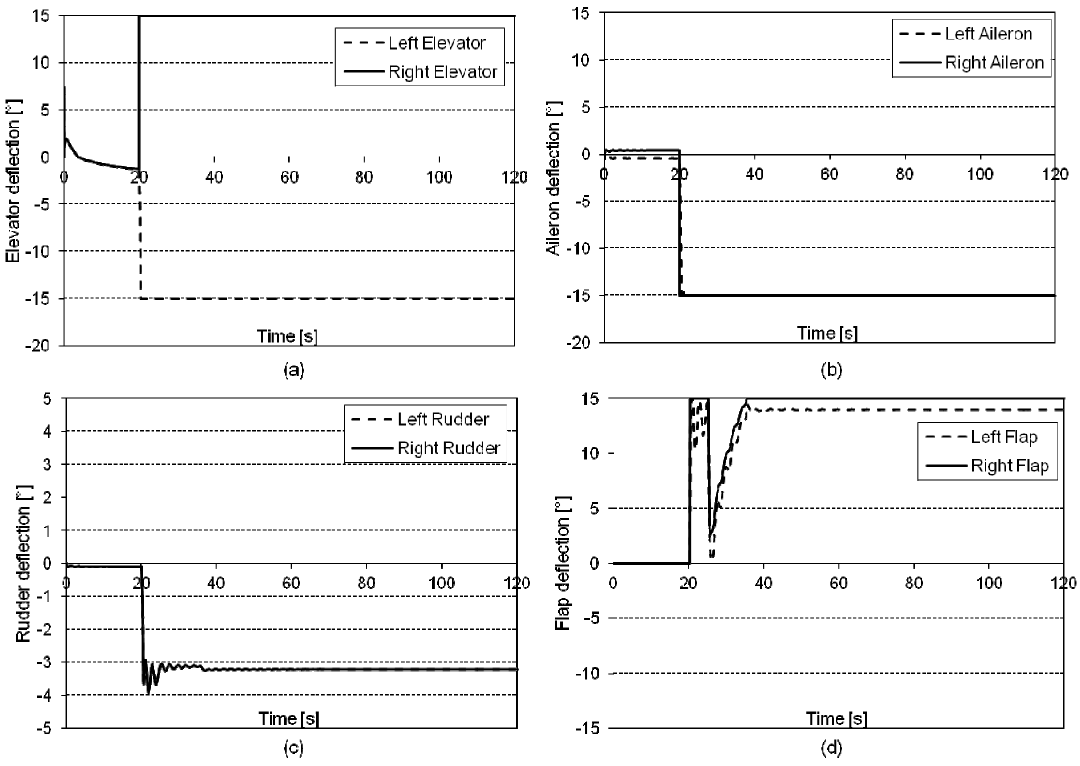


Figure 5.57: Simulated time history of the elevator (a), aileron (b), rudder (c) and flap (d) deflections with the adaption algorithm enabled (right elevator and aileron 100% hard over failure causing a nose up pitching moment and a rolling moment to the right) in level flight with control allocation enabled

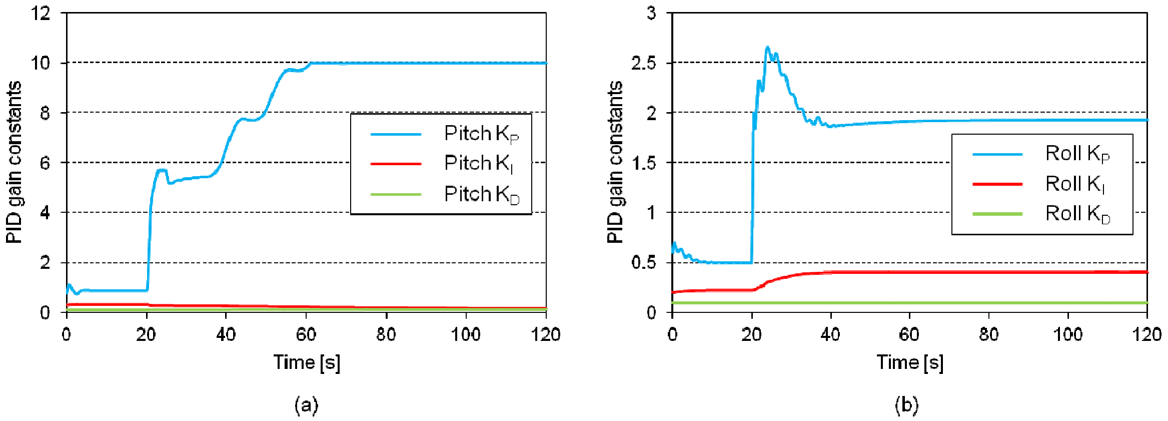


Figure 5.58: Simulated time history of the pitch (a) and roll (b) PID constants with the adaption and allocation algorithms enabled (right elevator and aileron 100% hard over failure causing a nose up pitching moment and a rolling moment to the right) in level flight

The simulation was also run for the constant banked turn and showed a similar set of results. The rolling moment caused by the failed aileron is easily overcome, as seen in Figure 5.59, by the application of differential flap shown in Figure 5.60 (d). This is also aided by the rudder-aileron mix, shown by the rudder deflection in Figure 5.60 (d). There is still an oscillatory response in pitch as the flap control reaches its physical limits. It is noted in Figure 5.59 that the control performance was generally improved by the adaption algorithm. Of concern, however, was the delay in returning the bank angle to the desired set point. It was also noted that there was a large offset and a large amount of chattering in sideslip angle due to the application of rudder, as shown in Figure 5.61. This resulted in an undesirable flight condition that was resolved by removing the aileron to rudder mix. The results of this simulation showed that the sideslip angle was restored to the trim position as expected; however, it was noted that there was a slightly improved performance of the system. Figure 5.62, shows that the oscillations in pitch have been resolved and no reduction in roll performance between the two cases is noted. It is hypothesised that this stems from a more stable control of roll without the influence of the rudder; hence, more control authority remains for the pitch control. While promising, Figure 5.63 (d) shows that all that has occurred is a delay in flap saturation. The figure shows a more gradual increase in flap deflection that, shortly after the simulation ends, will reach saturation and the oscillations will again become apparent.

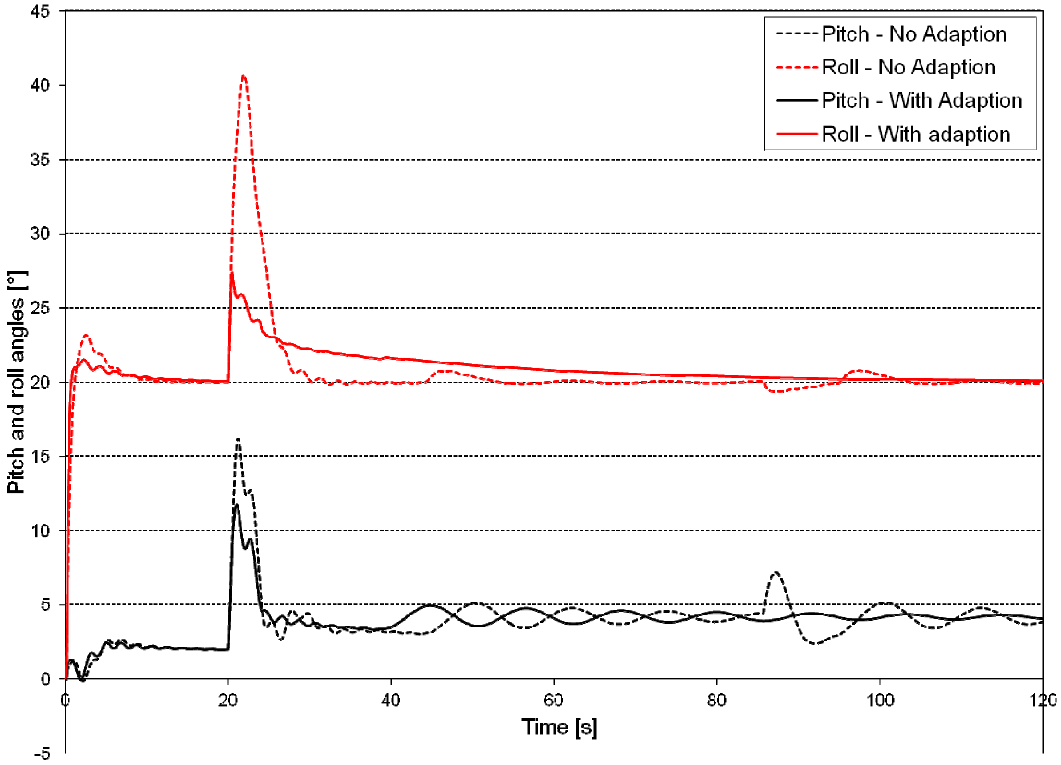


Figure 5.59: Simulated time history of pitch and bank angles (right elevator and aileron 100% hard over failure causing pitching moment in the nose up direction and a rolling moment to the right) with the aircraft in a constant bank turn with the control allocation algorithm enabled

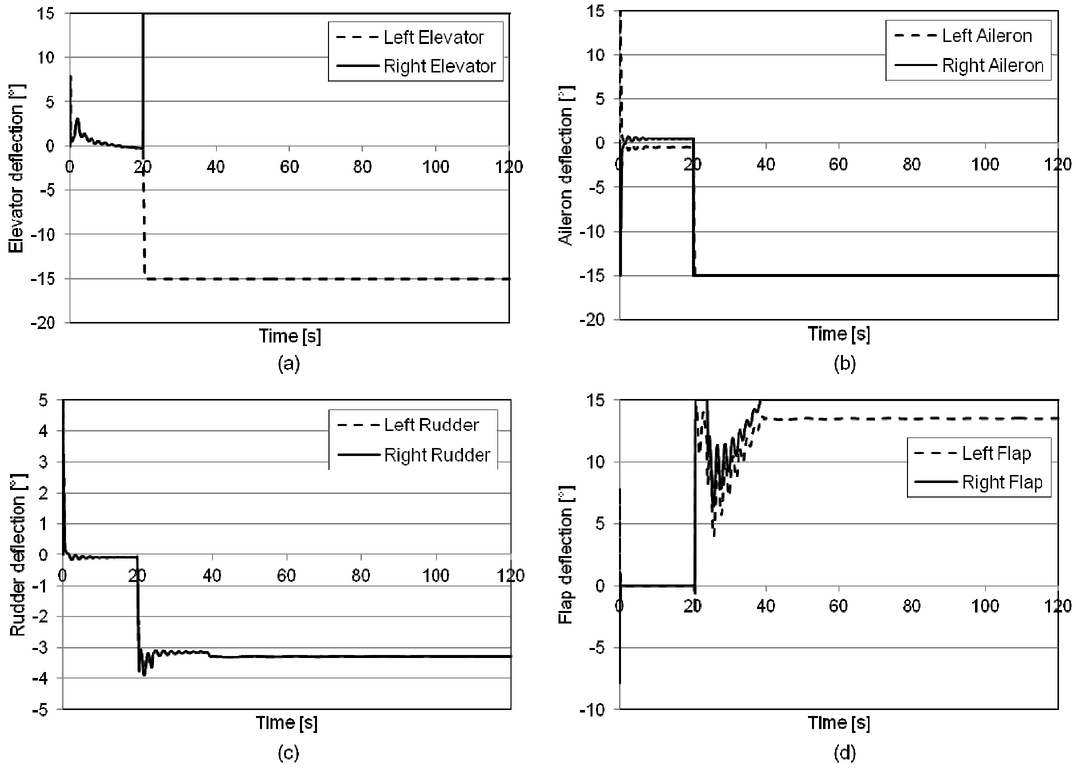


Figure 5.60: Simulated time history of the elevator (a), aileron (b), rudder (c) and flap (d) deflections with the adaption and allocation algorithms enabled (right elevator and aileron 100% hard over failure causing a nose up pitching moment and a rolling moment to the right) with the aircraft in a constant bank turn

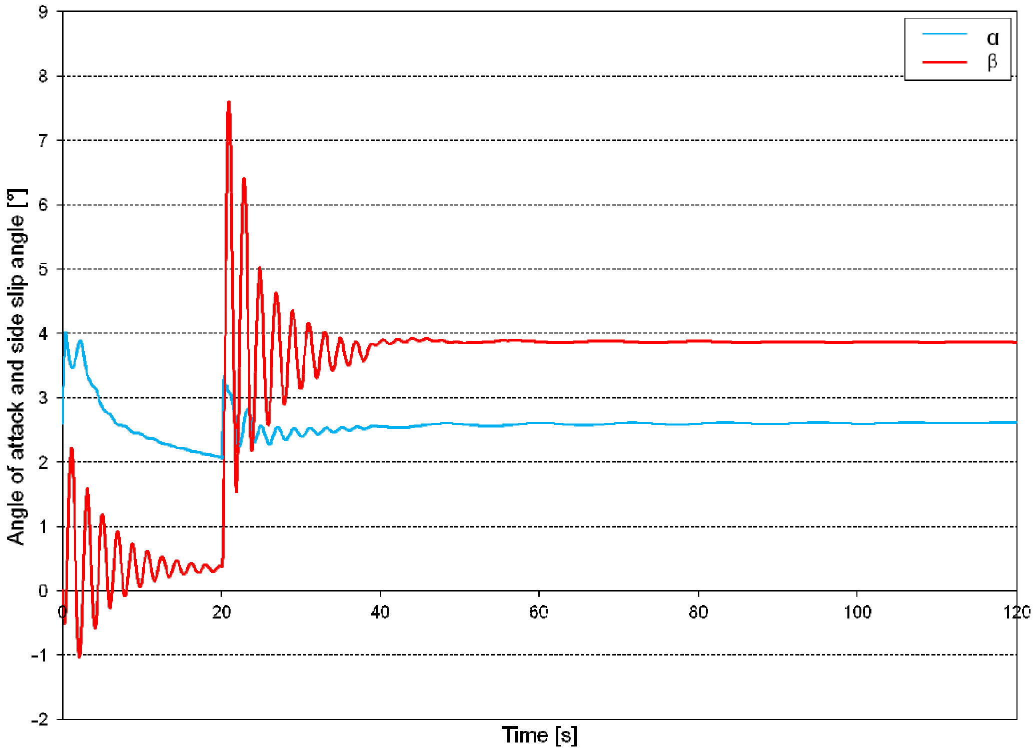


Figure 5.61: Simulated time history of angle of attack and sideslip angle with the adaption and allocation algorithms enabled (right elevator and aileron 100% hard over failure causing a nose up pitching moment and a rolling moment to the right) for the aircraft in a constant bank turn

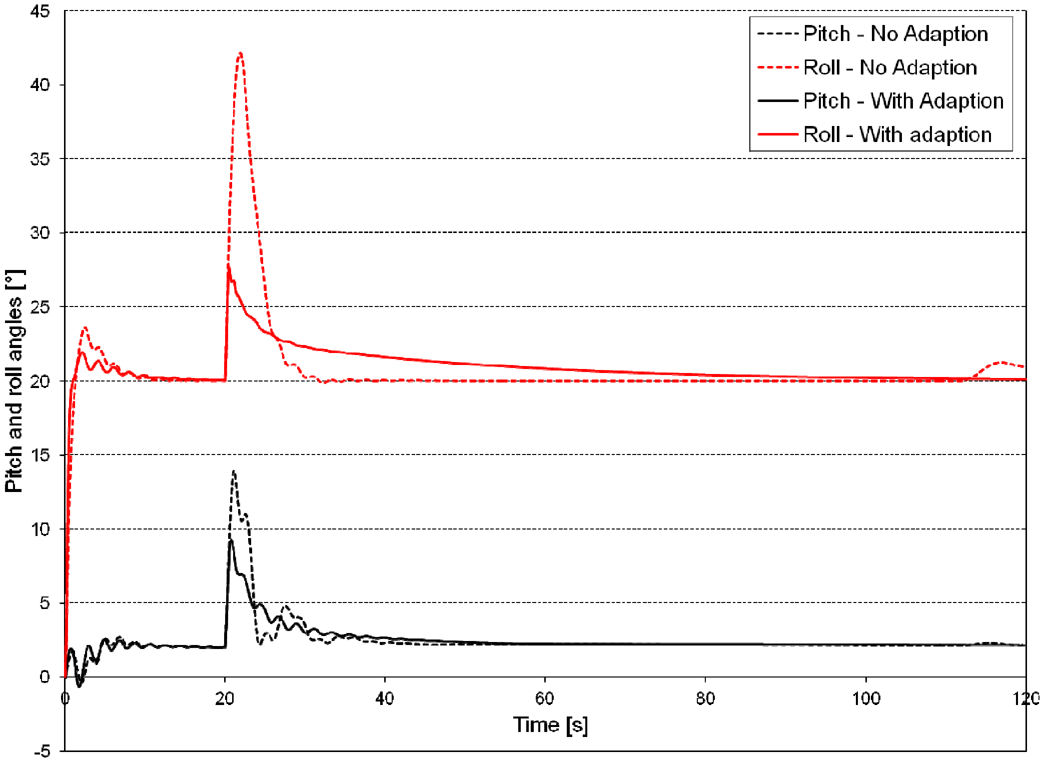


Figure 5.62: Simulated time history of pitch and bank angles (right elevator and aileron 100% hard over failure causing pitching moment in the nose up direction and a rolling moment to the right) with the aircraft in a constant bank turn with the control allocation enabled and no rudder mix

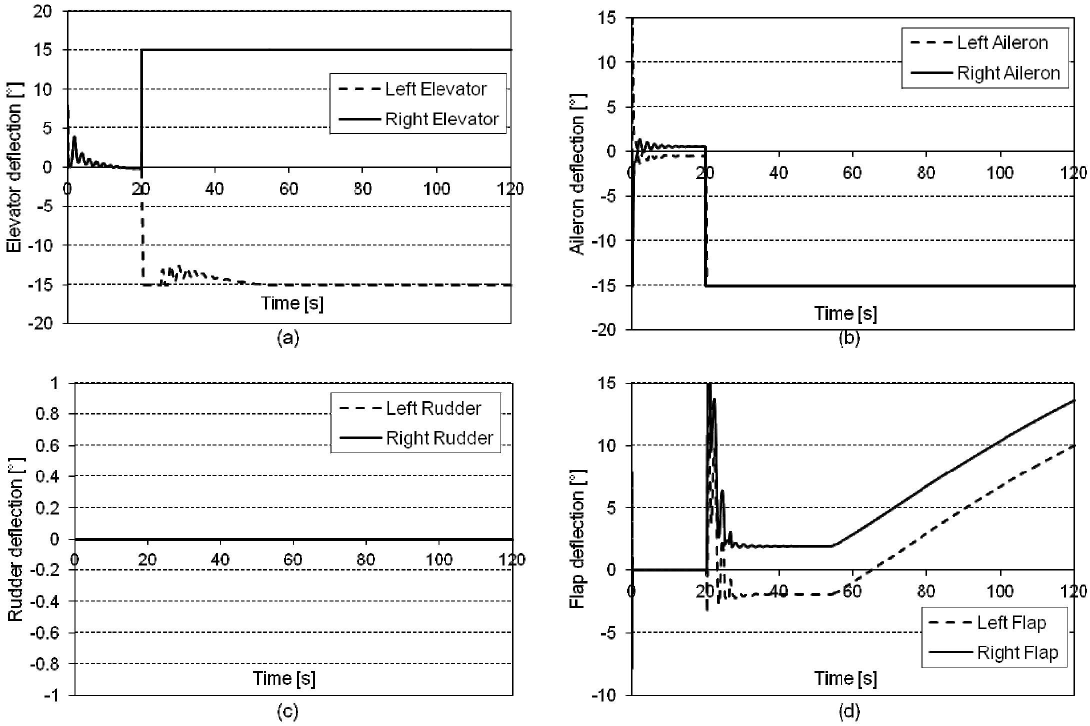


Figure 5.63: Simulated time history of the elevator (a), aileron (b), rudder (c) and flap (d) deflections with the adaption and allocation algorithms enabled (right elevator and aileron 100% hard over failure causing a nose up pitching moment and a rolling moment to the right) with the aircraft in a constant bank turn with no rudder mixing

Comparison between Unmodified Autopilot and Autopilot with all Proposed Control Algorithms Enabled (Test 13)

In order to show the full effect of the proposed control system, a simulation was run with a full aileron deflection failure and a 50% failure of the right elevator with the unmodified autopilot system and then again with the adaption algorithm and allocation algorithm enabled. A dramatic improvement is seen between these two simulations as shown in Figure 5.64. The figure shows that the unmodified autopilot fails to level the aircraft but instead the bank angle settles at approximately 65°. When all fault-tolerant algorithms are enabled, the simulation shows that the autopilot is able to return the aircraft to level flight quickly and maintain the required pitch angle. Figure 5.65 shows the change in PID constants and shows an initial increase in proportional gain for both pitch and roll PID controllers. The proportional gain for the pitch PID controller then undergoes a number of small oscillations but stabilises at a constant level thereafter. The roll PID proportional gain decreases at a steady rate and at the same time, the integral gain rises. This is due to a perceived steady-state error. However, at approximately 35 seconds all gains have stabilised.

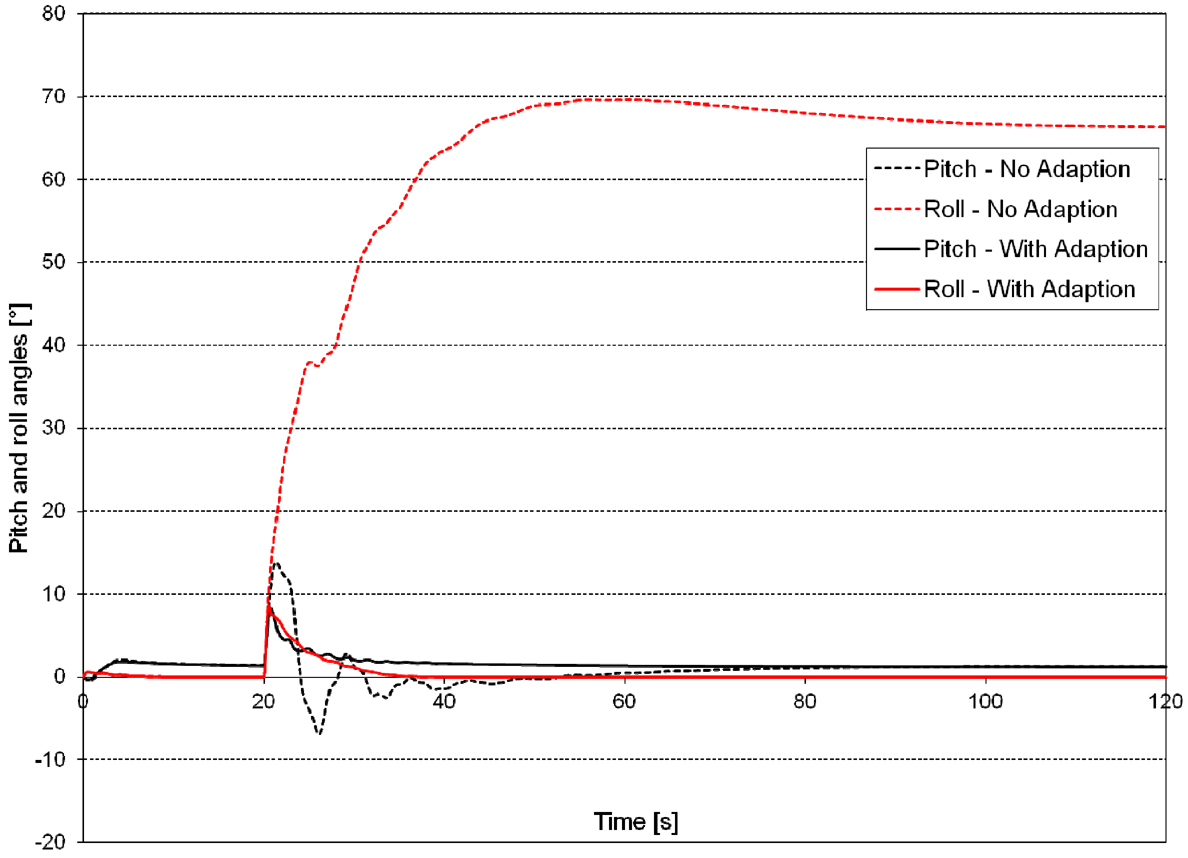


Figure 5.64: Simulated time history of pitch and bank angles (right aileron 100% hard over failure and right elevator 50% deflection failure causing rolling moment to the right and a pitching moment nose up) in level flight, with the autopilot unmodified and with the adaption and allocation algorithms enabled

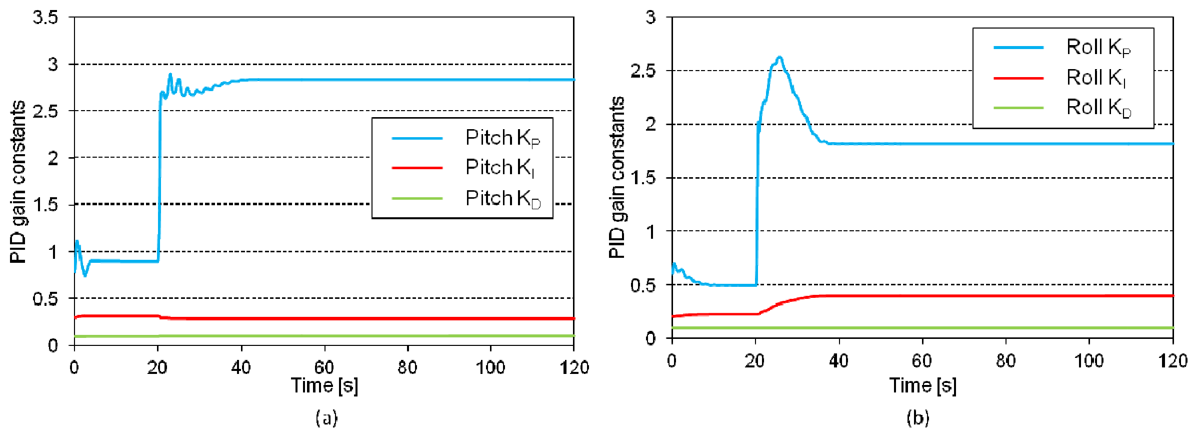


Figure 5.65: Simulated time history of the pitch (a) and roll (b) PID constants with the adaption and allocation algorithms enabled (right aileron 100% hard over failure and right elevator 50% deflection failure causing rolling moment to the right and a pitching moment nose up) in level flight

The success of this simulation is due to the control allocation algorithm. In Figure 5.66 (a) to (d), it was observed that additional pitch and roll authority was required at the moment of control failure. This can be seen as the elevator, Figure 5.66 (a), and aileron, Figure 5.66 (b), both saturate briefly after the failure. The control allocation algorithm compensates through the use of flaps, Figure 5.66 (d), which provide an additional pitching moment and rolling moment. The additional pitching moment required is observed in the simultaneous deflection of the flaps, moments after the control failure. However, as the pitch angle is regained, the elevators are able to provide enough control authority to return the aircraft to a level attitude.

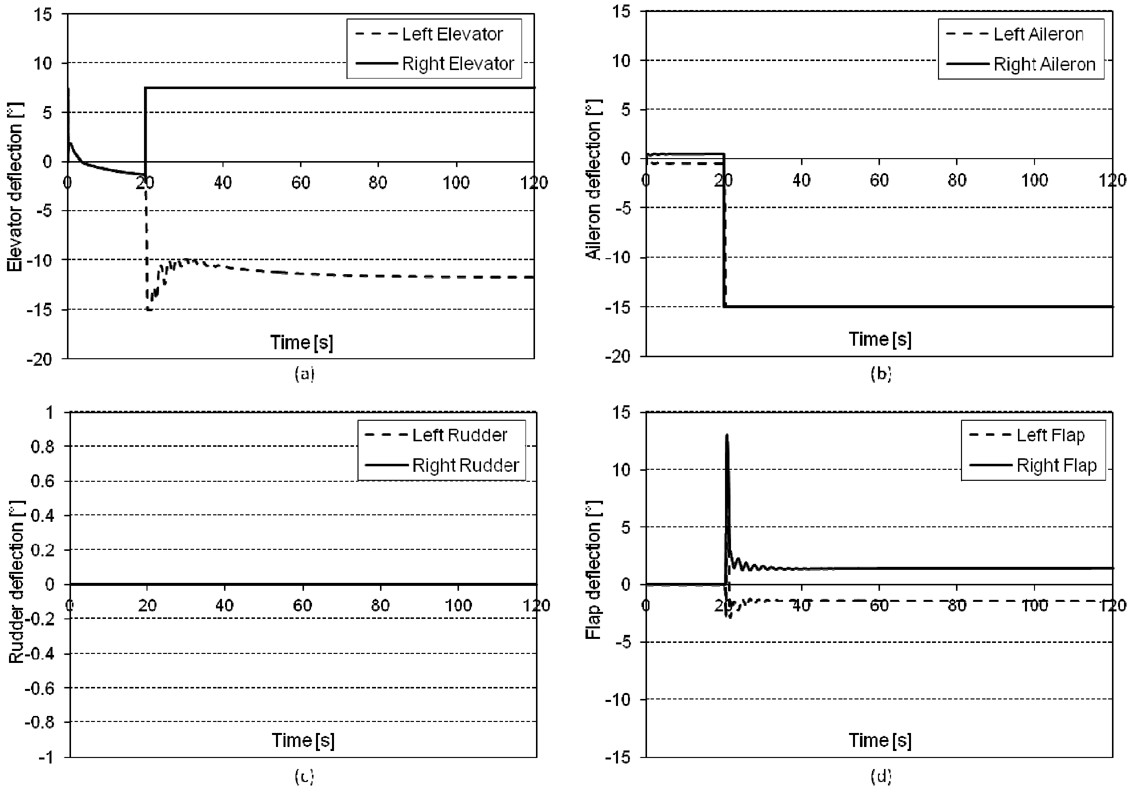


Figure 5.66: Simulated time history of elevator (a), aileron (b), rudder (c) and flap (d) deflections with the adaption mechanism and control allocation enabled (right aileron 100% hard over failure causing a rolling moment to the right) in level flight with no rudder mix

5.5.2 Middle Control Loop Simulation

The middle control loop was tested in the "FBW B" flight mode. This flight mode controls airspeed through the use of pitch angle and altitude through a form of total energy error as described in more detail in Section 3.7. For these tests, the airspeed was set to a value of 25m/s and the altitude set to 1524m above mean sea level. This altitude and airspeed was chosen as it was anticipated that this would be the primary flight condition should experimental testing be conducted. The bank angle is still controlled by the pilot in this flight mode. For these tests, two cases were run, namely level flight and 20° bank angle turn. For the cases without adaption, all adaptive algorithms were disabled, that is, the inner and middle loops were disabled. When the adaptive algorithm was enabled, all adaptive loops were enabled, namely pitch, roll, airspeed and energy.

Control - No Failure

A control simulation was run to determine the response of the aircraft under these conditions, with no failures. The simulation was run with and without the adaption algorithms, the results of which are promising. Figure 5.67 (a) shows that, without the adaption algorithms enabled, the airspeed follows a damped oscillation to the set point value. The decay is fairly rapid but a number of oscillations exist. When the adaption algorithm is enabled, the adaption algorithm tends to increase the initial overshoot in airspeed, but brings the airspeed back to the desired set point more quickly, with fewer oscillations.

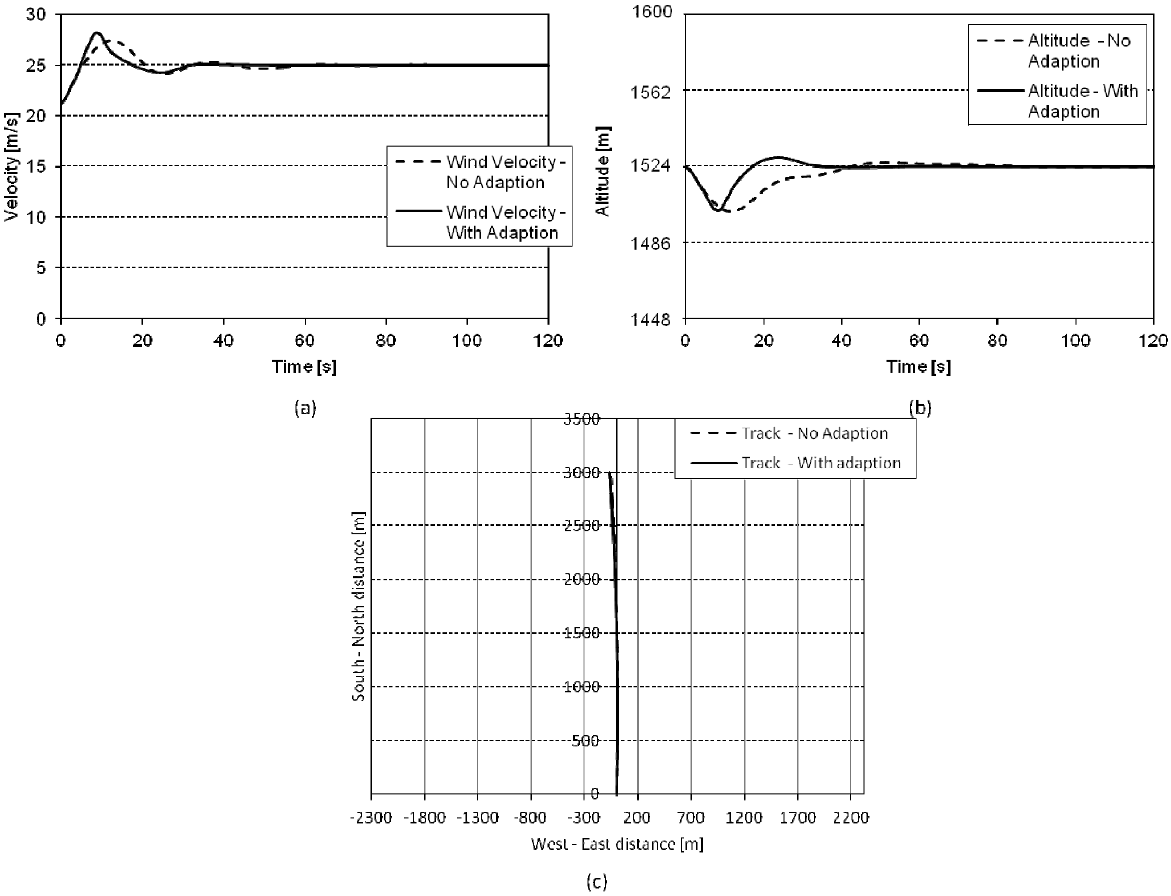


Figure 5.67: Simulated time history of airspeed (a), altitude (b) and the ground track (c) of the aircraft, with the adaption algorithm enabled and disabled with no failures (control)

Figure 5.67 (b) shows far better control performance with regard to altitude control. The altitude is controlled more tightly with the adaption algorithm enabled than disabled. It must be kept in mind that airspeed and altitude controllers are strongly coupled and their influence on each other is significant. The ability of the adaption algorithm to improve the control performance of the airspeed and altitude tracking is encouraging as it shows that initial concerns about the interaction of two adaption algorithms were not founded. Figure 5.67 (c) shows that the aircraft follows a fairly straight flight path with no noticeable differences observed between the two cases.

Figure 5.68 shows the variation in PID gain constants for all control loops. Figure 5.68 (a) shows the change in PID gains for the NavPitch (or airspeed) controller. This control loop controls airspeed by lowering or raising the nose of the aircraft. It can be seen in Figure 5.68 (a) that there is a rapid increase in proportional gain which levels off quickly to a stable value. There is also an increase in the energy gain, shown in Figure 5.68 (b). This is as one would expect as, because the aircraft is too slow initially and the autopilot tries to lower the nose to attain the correct airspeed, this in turn causes a loss of altitude which is catered for by the energy PID which will detect a negative energy error and increase the throttle. Some increase in integral gain for both the NavPitch and energy PID controllers is noted and this is thought to occur because of the small oscillations around the set point. Thus the adaption algorithm is detecting this as a steady state error. Figure 5.68 (c) shows a different pattern compared to behaviour observed in the inner loop simulation control. There is a spike in proportional gain followed by a decrease and then increase to a stable level. The reason for these changes becomes apparent in Figure 5.69.

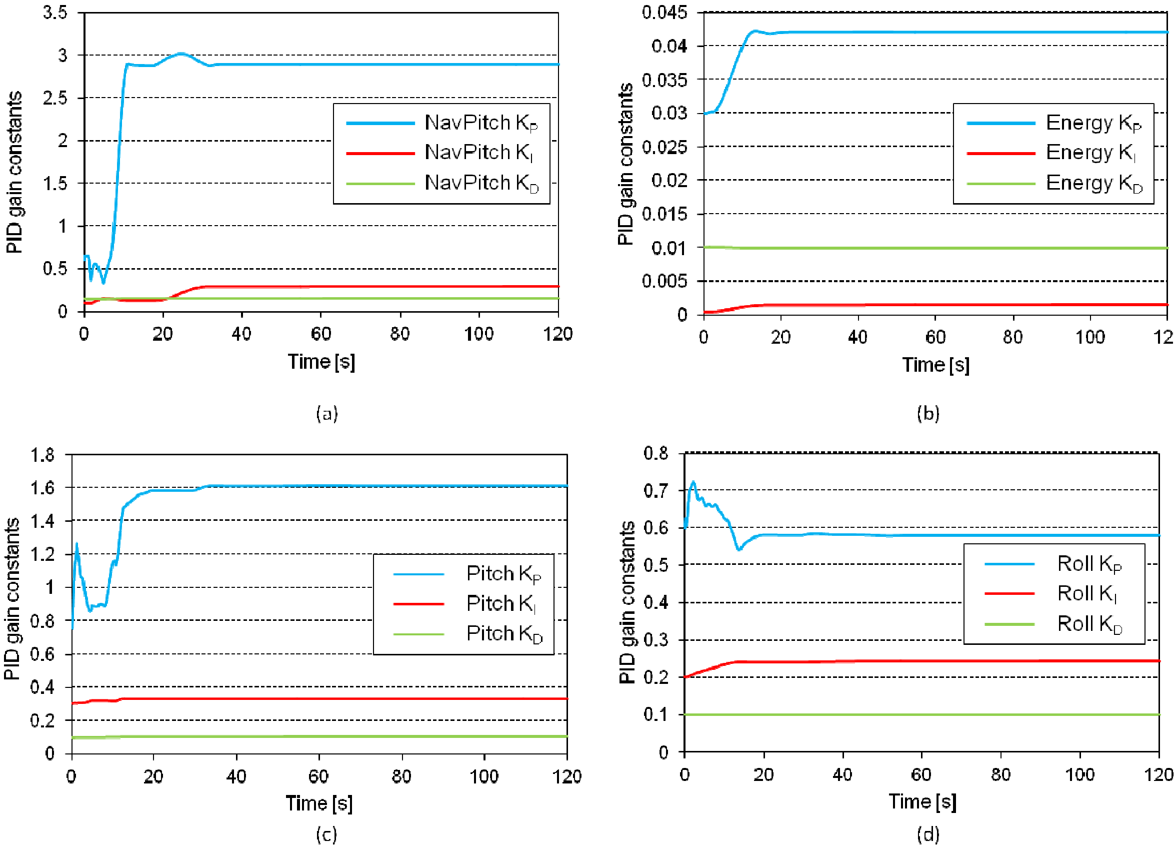


Figure 5.68: Simulated time history of the PID gain changes with the adaption mechanism enabled, for level flight with no failures.

Figure 5.69 shows the commanded pitch angle (NavPitch), the ideal behaviour, which tracks the NavPitch command very closely, the actual pitch of the aircraft and the pitch error. At the start of the simulation the pitch angle briefly undershoots the desired performance. This accounts for the increase in proportional gain. It then overshoots during the next peak, but is quickly returned to the ideal pitch angle at which point, after some minor modifications to the integral and proportional gain the ideal pitch behaviour is indeed well tracked. This is an important result. In the inner loop simulations, a constant NavPitch was set and the ability of the adaption algorithm to track a transient value was not seen. Figure 5.69 shows clearly that the adaptive algorithm is able to track a moving set point value to an acceptable level.

The simulation was run for a banked flight condition; however, the only significant differences in the results occurred for the roll angles. As this has been tested extensively in the inner loops, the results offer no added value as the middle loop has no major influence on the roll angle and no coupling exists. Accordingly the results of the simulations for banked flight, while provided in Appendix C, are not displayed or discussed here further.

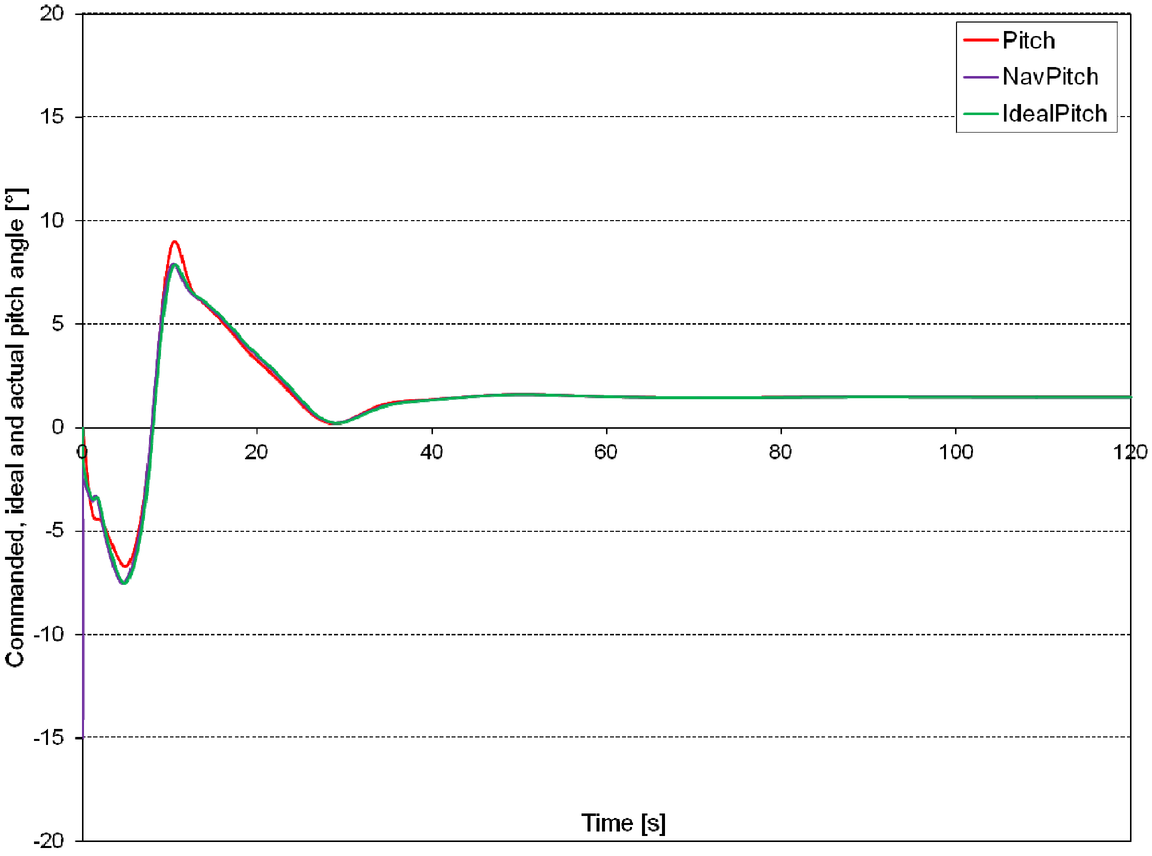


Figure 5.69: Simulated time history of the commanded pitch, ideal pitch and actual pitch for the aircraft in level flight with no failures.

Right Elevator Failure - 0° Deflection (Test 1)

The simulation was run inducing a 0° right elevator failure. To maintain consistency with the inner loop simulations, the failure was induced at 20 seconds. The results for the wings level case are presented in the figures that follow. Figure 5.70 (a) shows the time history of the airspeed, the results of which are almost identical to that of the control case. A similar result was obtained for the altitude plot and track of the aircraft, shown in Figure 5.70 (b) and (c) respectively. In fact, the only real change in the results was a subtle change in the pitch angle, shown in Figure 5.71, and, subsequently, a minor change in the proportional gain of the pitch PID controller, shown in Figure 5.72 (c). The reason for this lack of change may have been due in part to the timing of the failure and that the main controlled parameters of airspeed and altitude are reasonably tolerant to small changes in elevator angle.

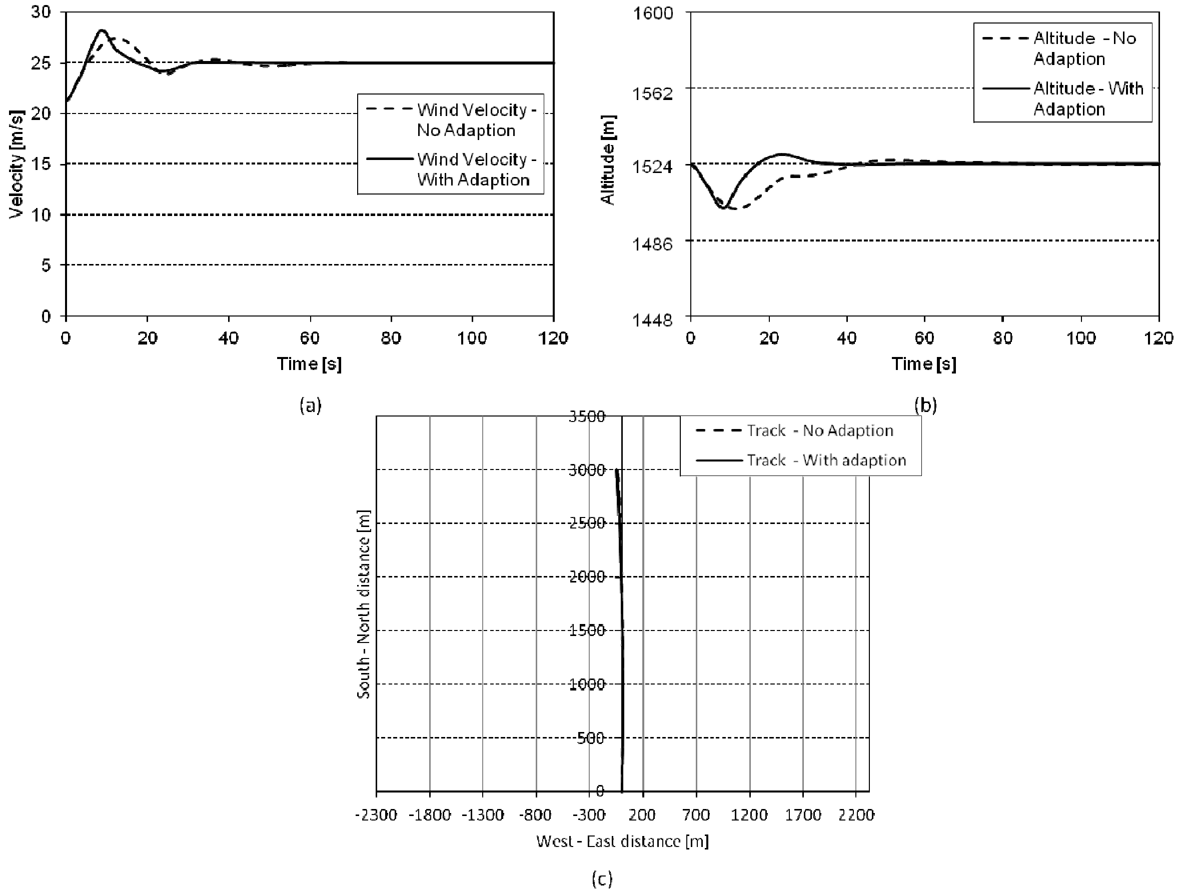


Figure 5.70: Simulated time history of airspeed (a), altitude (b) and the ground track (c) of the aircraft, with the adaption algorithm enabled and disabled, with a 0° failure of the right elevator

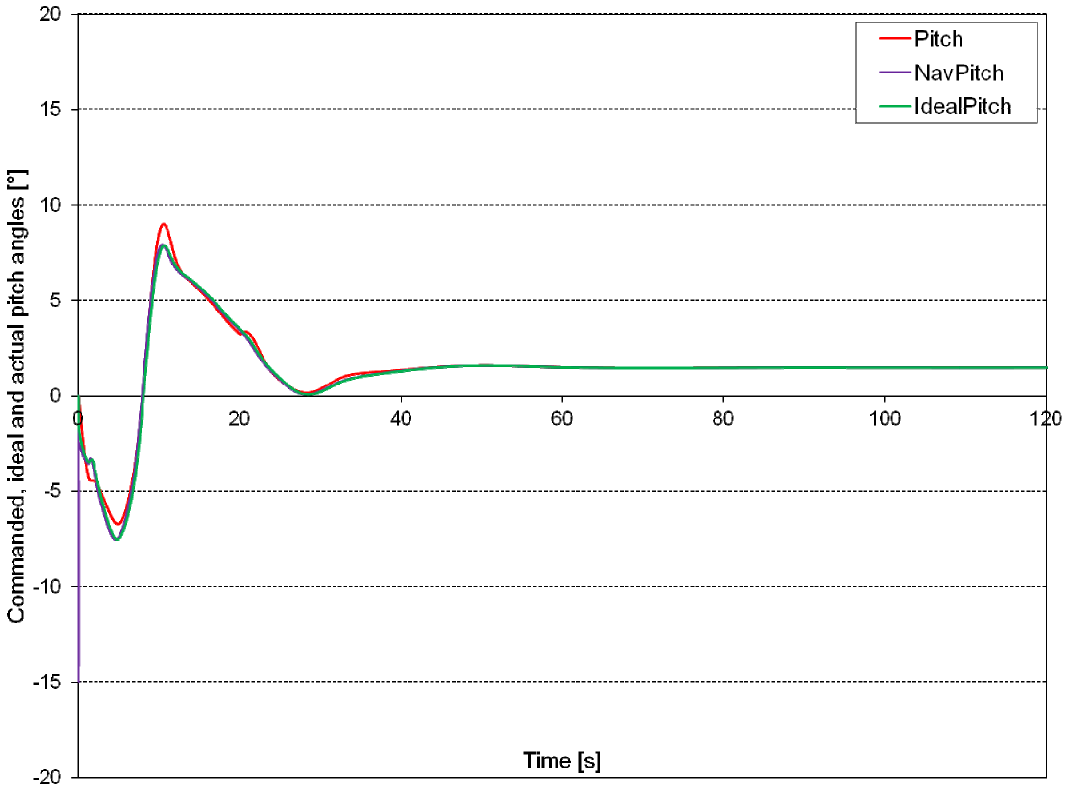
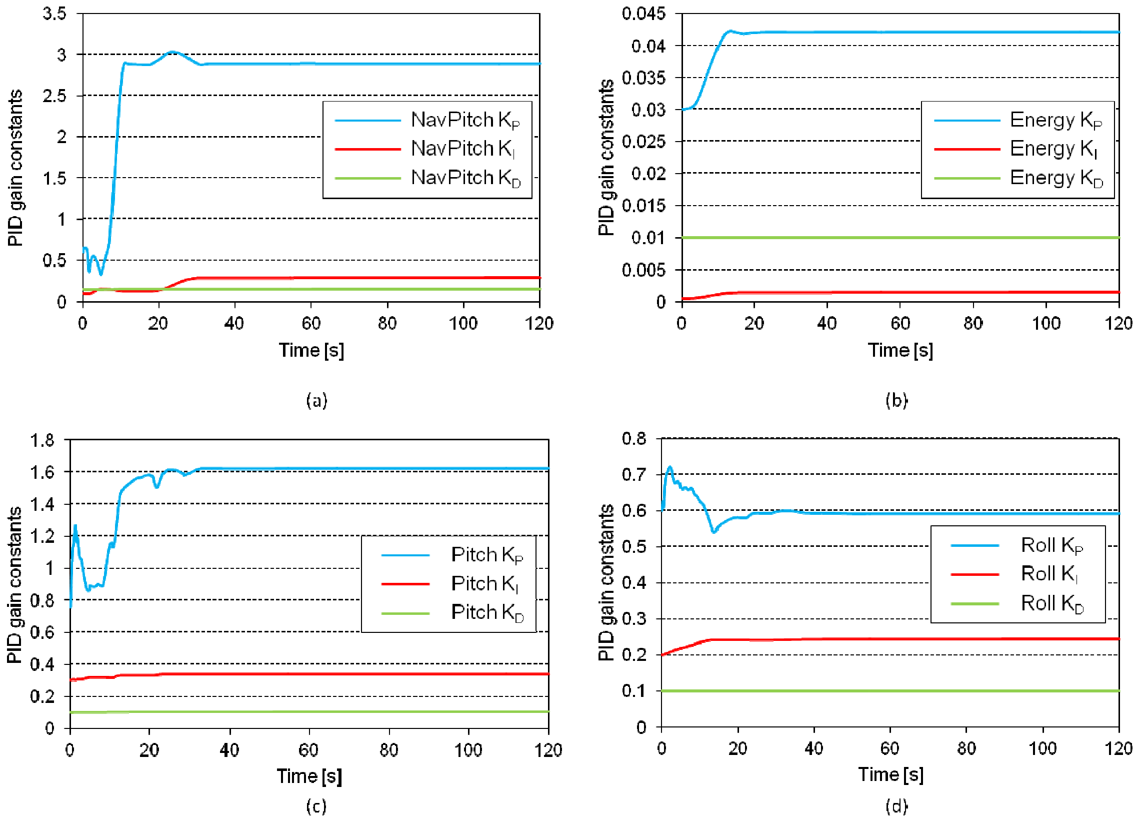


Figure 5.71: Simulated time history of the commanded pitch, ideal pitch and actual pitch for the aircraft in level flight with a 0° failure of the right elevator



(a)

(b)

(c)

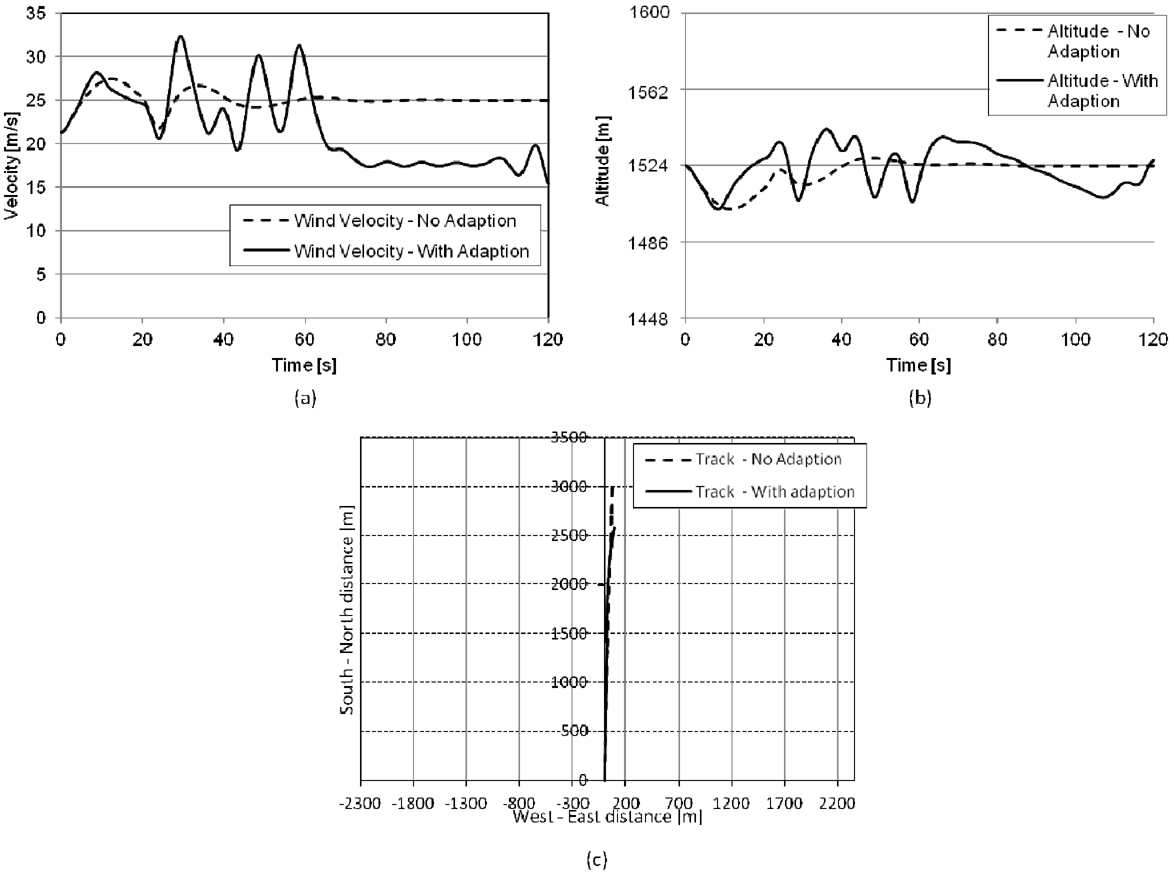
(d)

.

Figure 5.72: Simulated time history of the PID gain changes with the adaption mechanism enabled for level flight with a 0° failure of the right elevator.

Right Elevator Failure - 50% Hard Over (Test 2)

The simulation was run inducing a 50% hard over failure of the right elevator causing a nose-up pitching moment coupled with a roll to the right. Figure 5.73 (a) shows the failure and shows the inability of the adaption algorithm to tolerate this fault whereas the simulation without adaption was able to maintain stable flight. Figure 5.73 (b) shows a similar result for altitude control in that an unstable and erratic altitude control occurs with the adaption algorithm enabled. The track of the aircraft, shown in Figure 5.73 (c), shows a slight change in flight path which is reasonably straight.



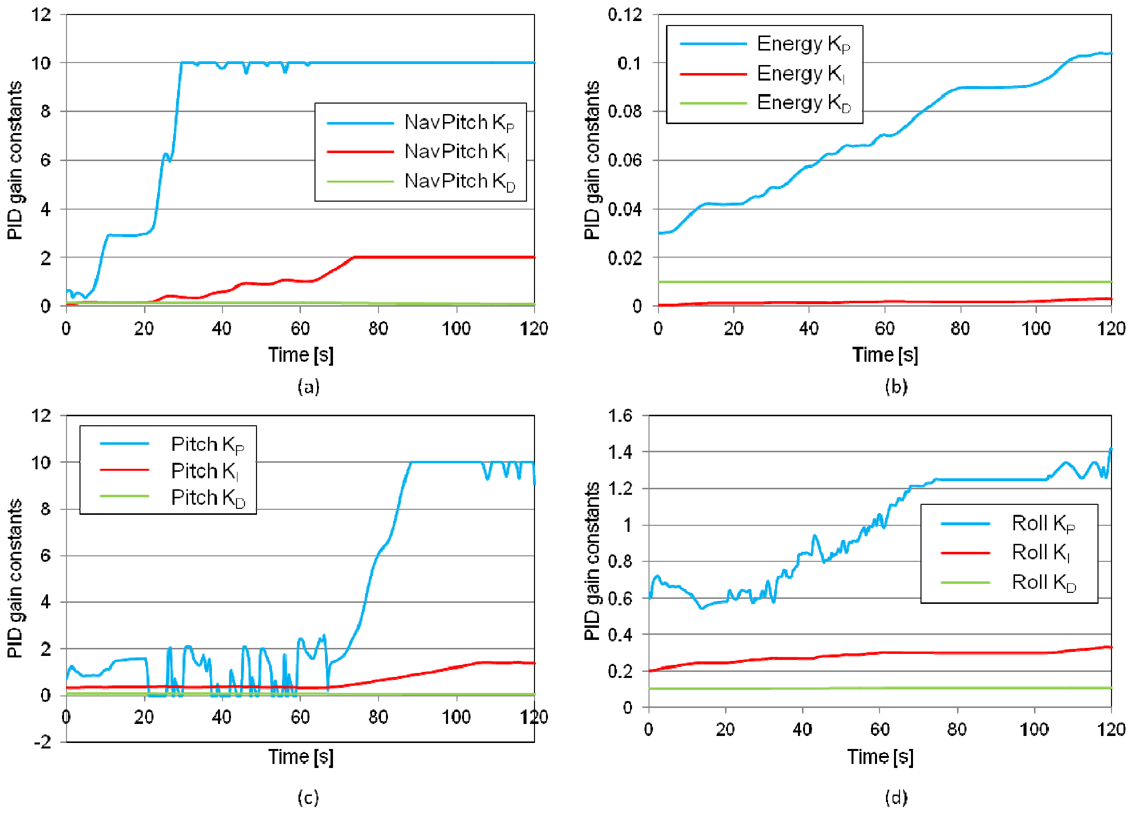
(a)

(b)

(c)

Figure 5.73: Simulated time history of airspeed (a), altitude (b) and the ground track (c) of the aircraft with the adaption algorithm enabled and disabled with a 50% hard over failure of the right elevator

The cause of this failure of the adaption algorithm is largely unknown, but some insight is gained when examining Figure 5.74 (a) which shows the PID gain constant variation of the PID controller responsible for airspeed control. The proportional gain rose rapidly after the failure to the maximum allowed level set in the programme which explains the erratic behaviour of the system. The integral gain also increases which is surprising as a steady-state error is not observed. The oscillations in the airspeed would have induced the oscillations seen in the altitude plot of Figure 5.73 (b) in two ways. First, the attitude is being commanded to the extreme limits of pitch angle thus causing the loss and gain in altitude but, secondly, the oscillations also cause the proportional gain of the energy PID to increase, shown in Figure 5.74 (b), thus exacerbating the altitude fluctuations. Consequently, the inner loop PID gains also undergo a series of fluctuations that do not help to mitigate the erratic result obtained. The roll PID, which is not as strongly coupled to the other PID controller response, is somewhat better behaved although in comparison to the inner loop tests, the change in gains are far more aggressive and oscillatory. Figure 5.75 sheds some light on the failure of the system to resolve the failure. It is noted that the NavPitch command reached the prescribed maximum and minimum values. These were initially set in the autopilot to curb excessive changes in pitch. But as seen in the inner loop cases, when the method of control saturated, a runaway condition of the proportional PID gain was observed. It was theorised that increasing the allowable pitch limits may resolve some of the problems seen; however, this was subsequently found to create greater instability within the system. Another contributing factor could be the size of the adaption rates; thus, the airspeed adaption rate was reduced by a factor of 10. The results of this simulation are presented in Figure 5.76.



(a)

(b)

(c)

(d)

Figure 5.74: Simulated time history of the PID gain changes with the adaption mechanism enabled for level flight with a 50% hard over failure of the right elevator

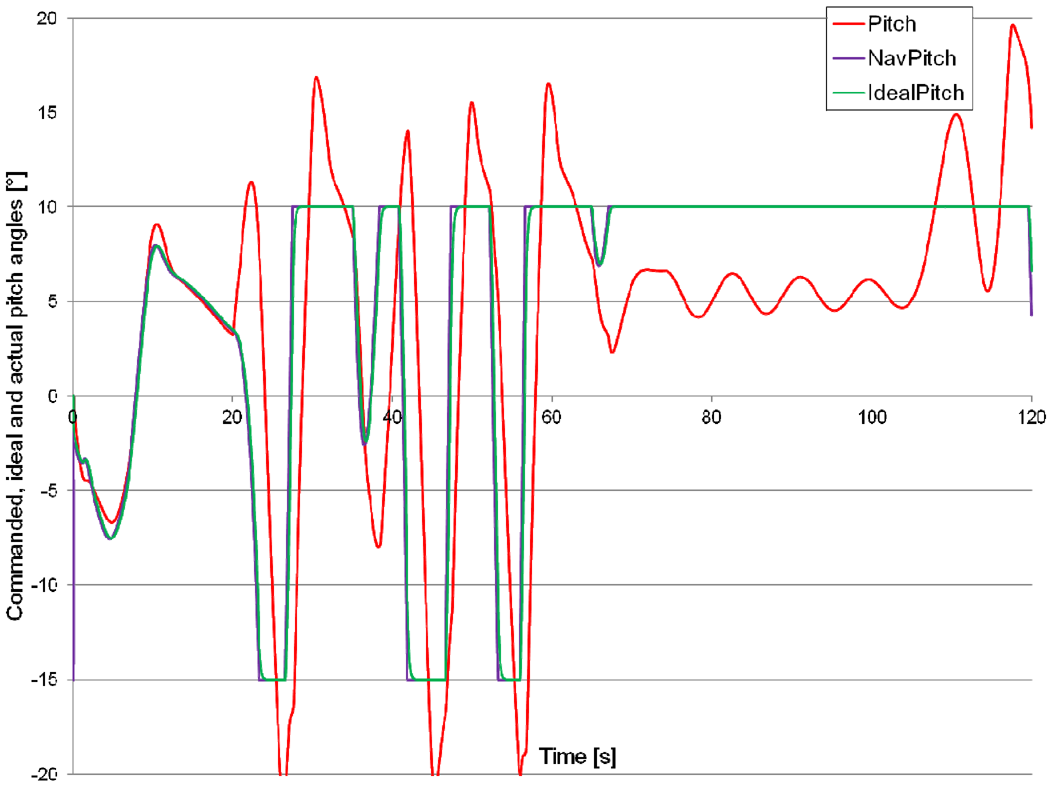


Figure 5.75: Simulated time history of the commanded pitch, ideal pitch and actual pitch for the aircraft in level flight, with a 50% hard over failure of the right elevator

The results presented in Figure 5.76 show that the instabilities have been removed to a large extent and now the adaption algorithm is able to tolerate this failure. However, the control performance in controlling airspeed has reduced to some degree, as shown in Figure 5.76 (a). The airspeed has a larger overshoot when compared to the simulation without adaption although the time to return to the set point value is perhaps marginally quicker. The altitude plot, shown in Figure 5.76 (b), shows improved performance although the gains are small. Little change was observed in the ground track of Figure 5.76 (c).

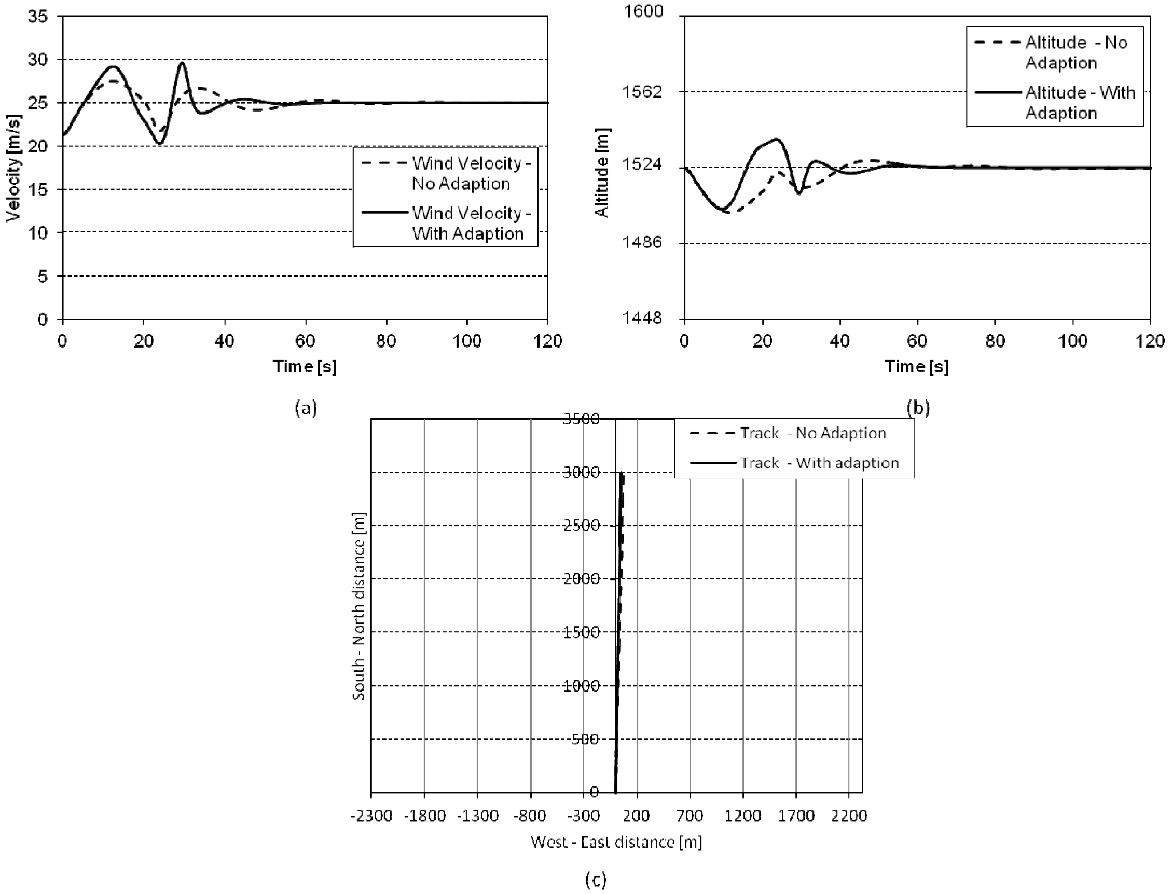


Figure 5.76: Simulated time history of airspeed (a), altitude (b) and the ground track (c) of the aircraft with the adaption algorithm enabled and disabled, with a 50% hard over failure of the right elevator but with a reduced airspeed PID adaption rate

Improved results are observed in the change in PID gains for this simulation as shown in Figure 5.77. In all of the figures, (a) to (d), there is a far more stable change in the gain values. The gains do not rise to the extent seen in Figure 5.74 and there are fewer oscillations. It is noted that a large overshoot is observed for the pitch trace (Figure 5.78) in trying to return the measured pitch to the ideal pitch. This is likely to be the cause of the reduced performance in tolerating this failure. Hence, a modification of pitch PID adaption rate was tested and it was found that improved performance was attained when increasing the adaption rate by a factor of 10. Decreasing the adaption rate yielded a reduction in performance. Interestingly, increasing the adaption rate of the airspeed PID controller back to the levels of the first simulation yielded a greater improvement in control performance. As a result, the simulation was run with the original airspeed adaption rate but with an increased pitch PID adaption rate, the results of which can be found in Figure 5.79.

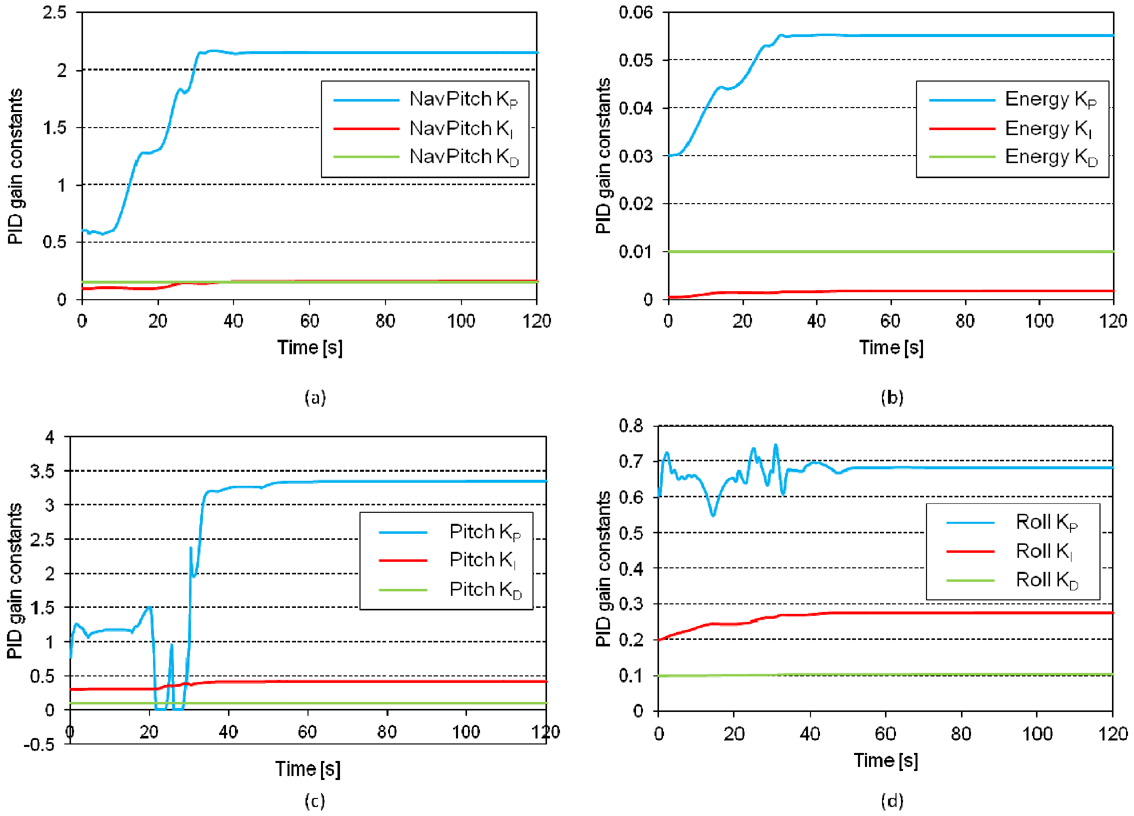


Figure 5.77 Simulated time history of the PID gain changes with the adaption mechanism enabled for level flight with a 50% hard over failure of the right elevator but with a reduced airspeed PID adaption rate

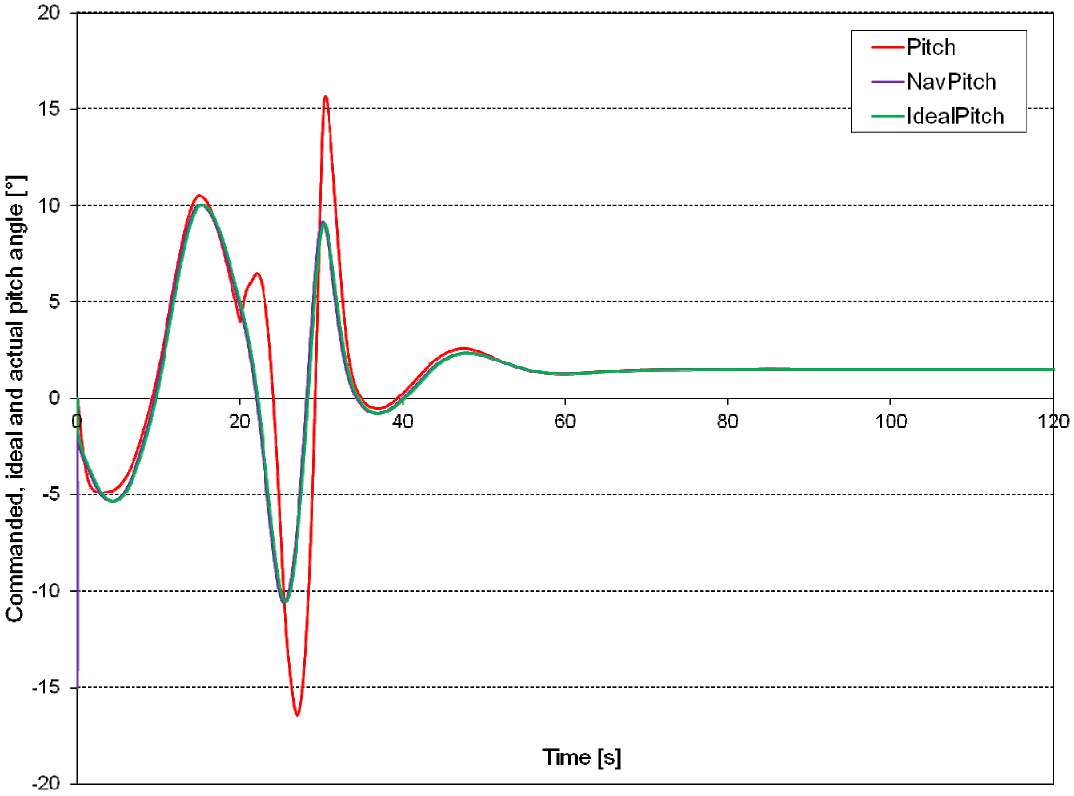


Figure 5.78: Simulated time history of the commanded pitch, ideal pitch and actual pitch for the aircraft in level flight with a 50% hard over failure of the right elevator but with reduced airspeed PID adaption rate

Figure 5.79 (a) shows that the response of the airspeed has improved from that of Figure 5.76 (a) in that the overshoot and time to return to the set point value is now better than the non adapted case. Similarly, Figure 5.76 (b) shows that the altitude response is also improved when compared to the non adapted case. Figure 5.76 (c) again shows little difference in ground track. Some concern lies in the high values of PID gains seen for the airspeed and pitch controllers, as seen in Figure 5.80 (a) and (c) respectively. In particular, the pitch PID gain achieves a maximum value above eight which was found that, in general, for the inner flight modes, resulted in poor performance of the system. Of note was the improved tracking performance of the measured pitch compared to the ideal pitch, as seen in Figure 5.81, when compared to the results of Figure 5.78. After the failure has occurred, there is no overshoot in the measured pitch compared to ideal pitch and the ideal pitch is tracked very well.

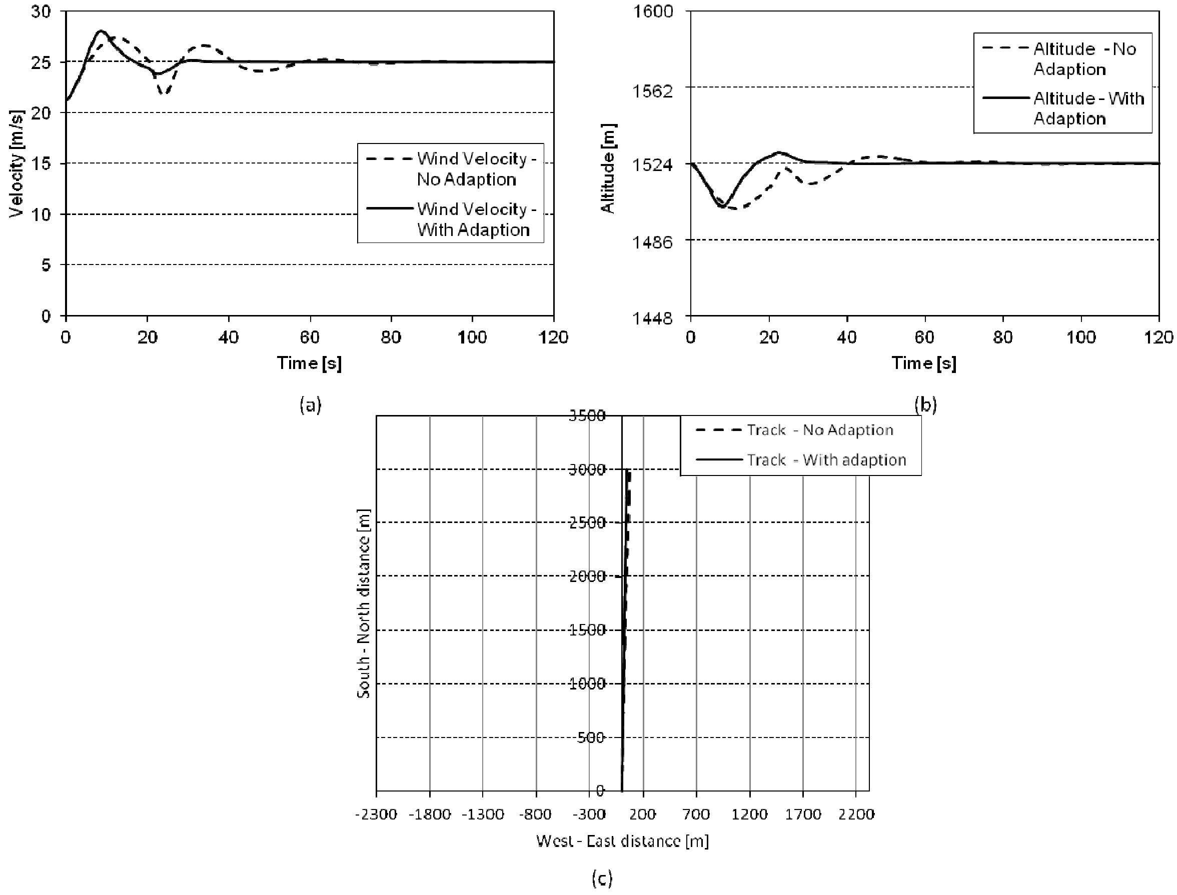


Figure 5.79: Simulated time history of airspeed (a), altitude (b) and the ground track (c) of the aircraft with the adaption algorithm enabled and disabled, with a 50% hard over failure of the right elevator but with an increased pitch PID adaption rate

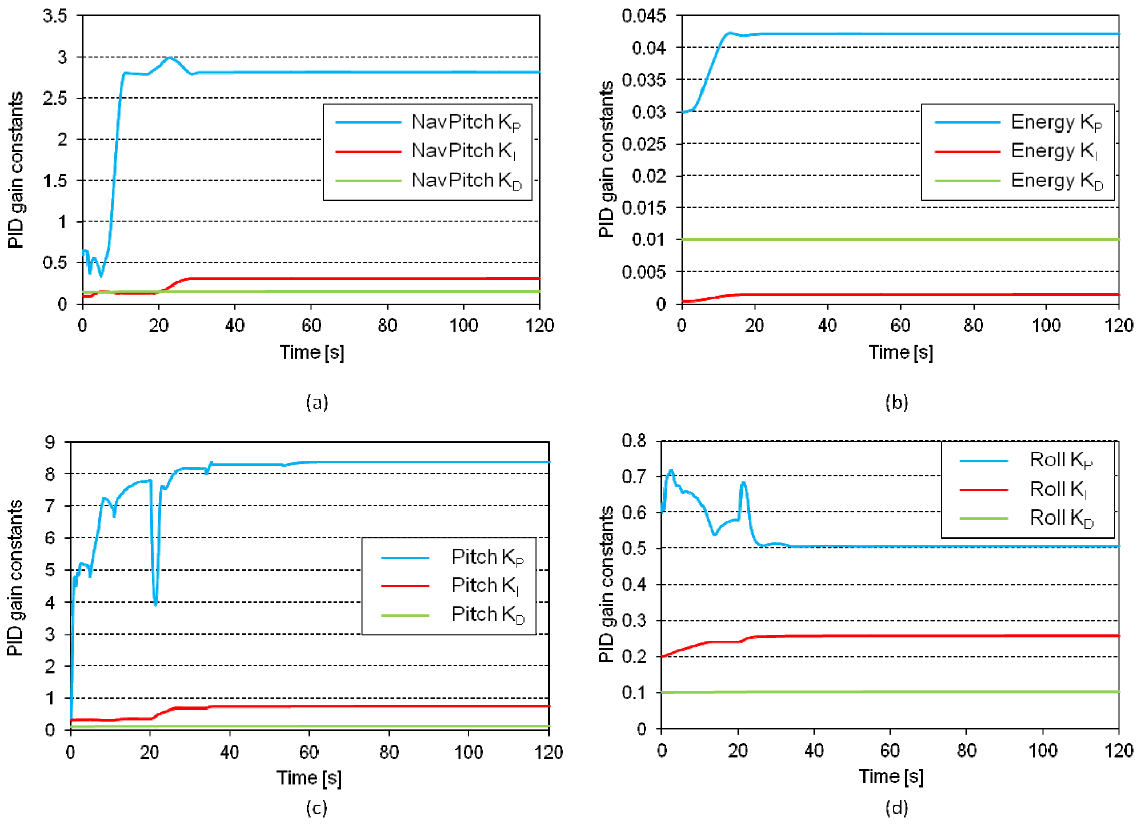


Figure 5.80: Simulated time history of the PID gain changes with the adaption mechanism enabled for level flight with a 50% hard over failure of the right elevator but with an increased pitch PID adaption rate

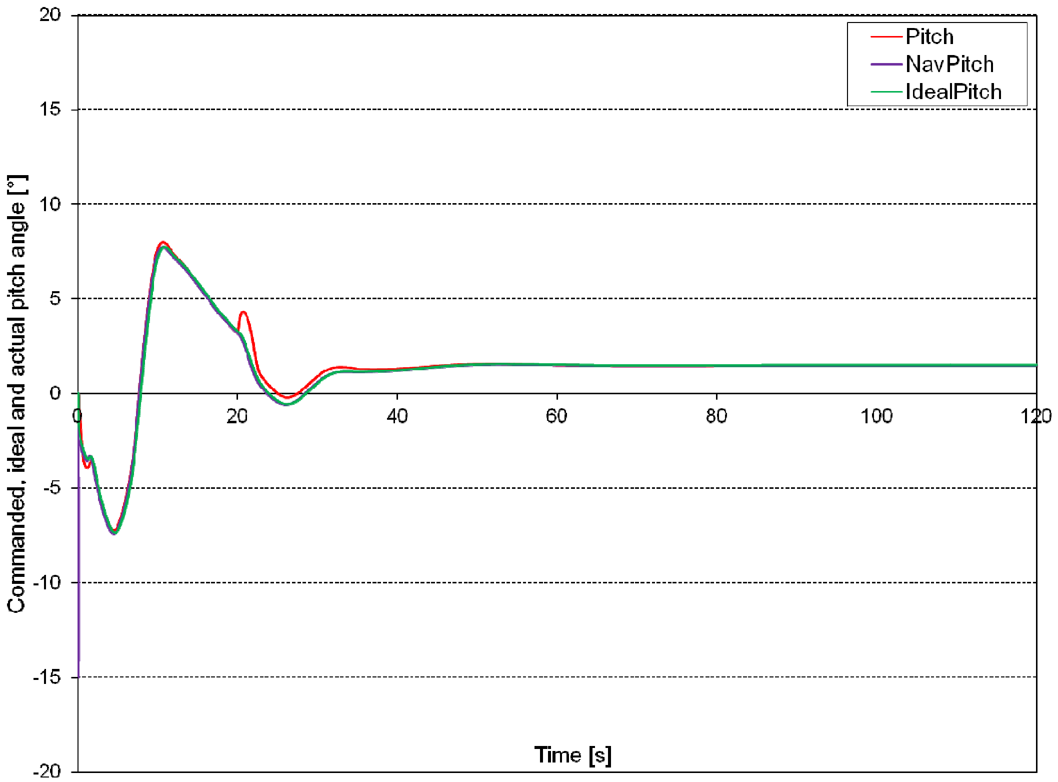


Figure 5.81: Simulated time history of the commanded pitch, ideal pitch and actual pitch for the aircraft in level flight with a 50% hard over failure of the right elevator but with reduced airspeed PID adaption rate

The results for the simulation of banked flight for the most successful configuration are presented in the figures that follow. The results of the banked flight case for the previous configurations have been excluded in this dissertation (But are available in Appendix C) as the results obtained generally followed a pattern similar to the level flight case. The results for the adaption algorithm in its most "tuned" form are presented as these offer the most insight.

Figure 5.82 (a) shows a slightly different result from the level flight case in that the control performance gain is marginal. While the set point value is attained more quickly than in the case without adaption, there is a far higher overshoot value. A similar result is obtained for the altitude plot which shows a large initial disturbance but a rapid re-establishment of the set point value. Again, of some concern is the high proportional gain value achieved in Figure 5.83 (a) and (c) respectively. This approached the maximum allowable value of 10 for the airspeed PID and reached the maximum value of 10 for the pitch PID. The concern is that, if the flight manoeuvre was more aggressive, the system could become unstable. Similarly to the level flight case, the throttle and roll PID gains, shown in Figure 5.83 (b) and (d) respectively, did increase but not to the extreme levels seen in Figure 5.83 (a) and (c).

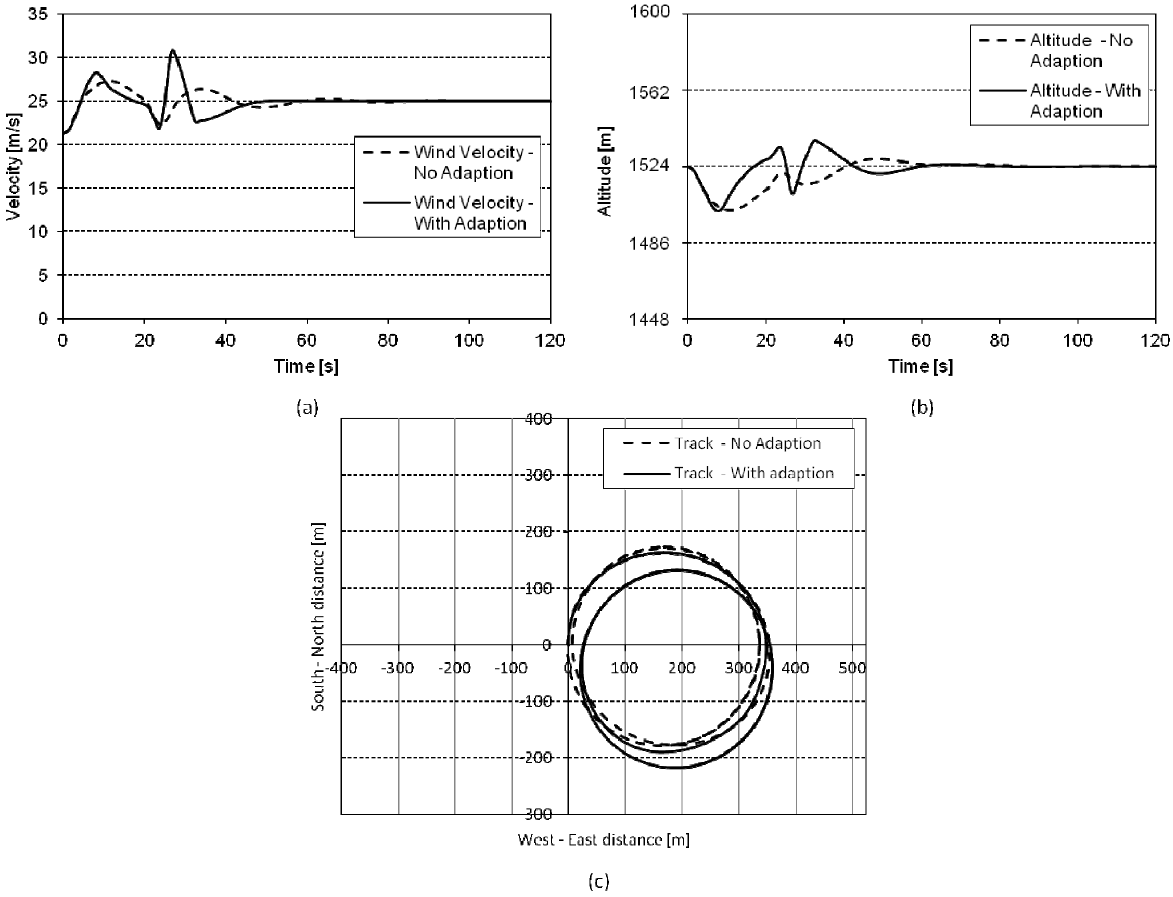


Figure 5.82: Simulated time history of airspeed (a), altitude (b) and the ground track (c) of the aircraft in a constant bank turn with the adaption algorithm enabled and disabled with a 50% hard over failure of the right elevator but with an increased pitch PID adaption rate

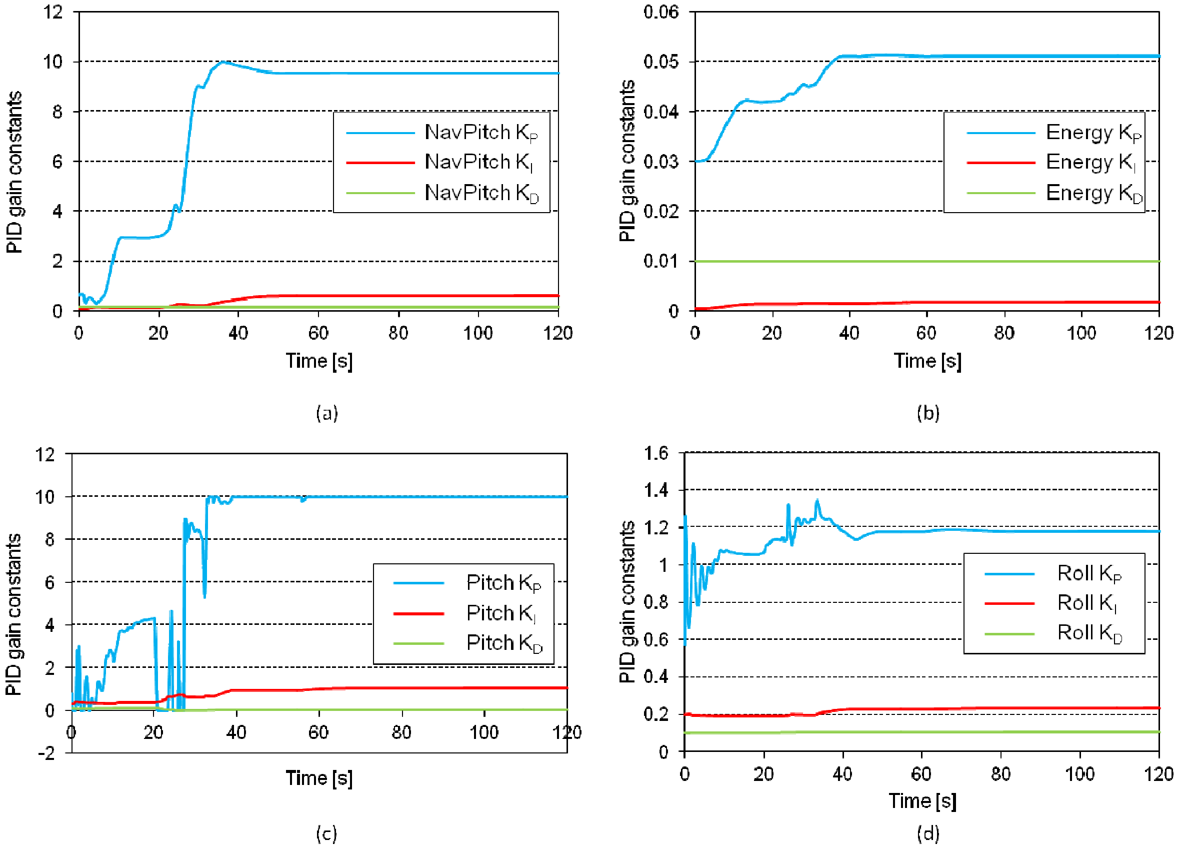


Figure 5.83: Simulated time history of the PID gain changes with the adaption mechanism enabled for the aircraft in a constant bank turn with a 50% hard over failure of the right elevator and an increased pitch PID adaption rate

Some of the reduced control performance noted in the results may be attributed to the difficulty encountered by the autopilot in tracking the ideal pitch value as seen in Figure 5.84 which shows that the pitch angle tracking was poorly achieved until approximately 15 seconds after the failure. After this time, a reasonable degree of tracking was achieved. Also of note was the introduction of flap deflection, shown in Figure 5.85 (d), to aid in pitch control, previously unseen in the level flight case. The flap deflections are momentary as the elevator control saturates and indicate that the autopilot may have failed to resolve this failure without the allocation algorithm in operation. It also highlights that the PID gains have risen too high as the autopilot should not require that amount of control effort to maintain level flight as the elevator failure was only a 50% deflection failure.

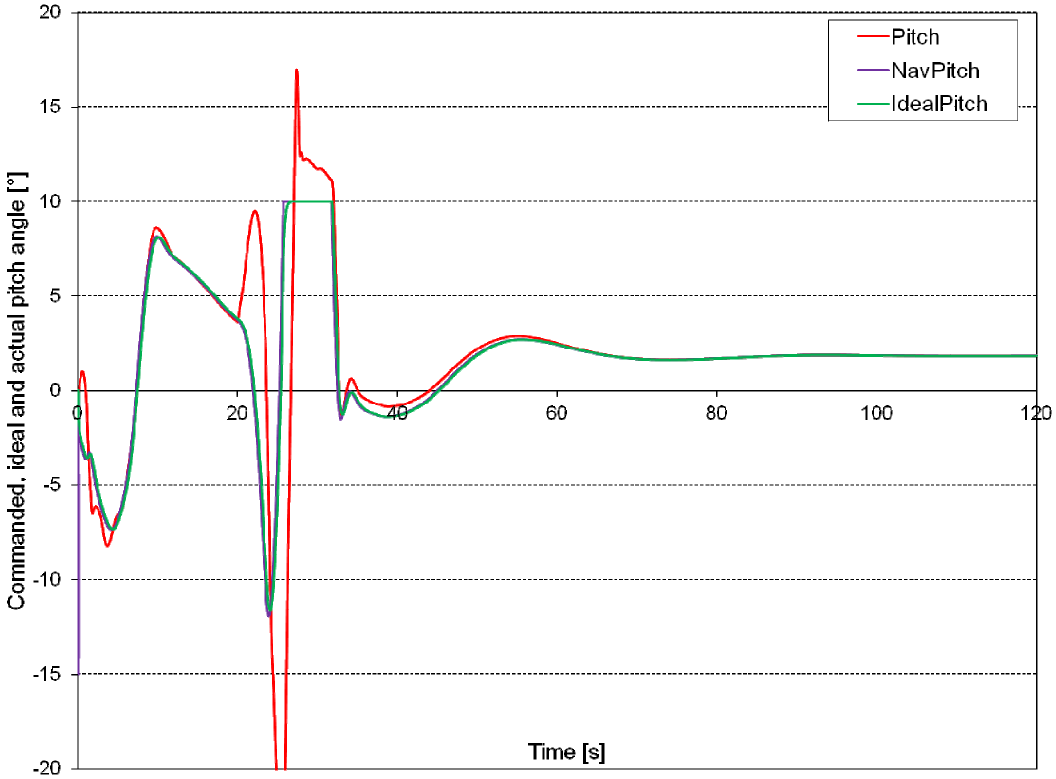


Figure 5.84: Simulated time history of the commanded pitch, ideal pitch and actual pitch for the aircraft in level flight with a 50% hard over failure of the right elevator but with reduced airspeed PID adaption rate

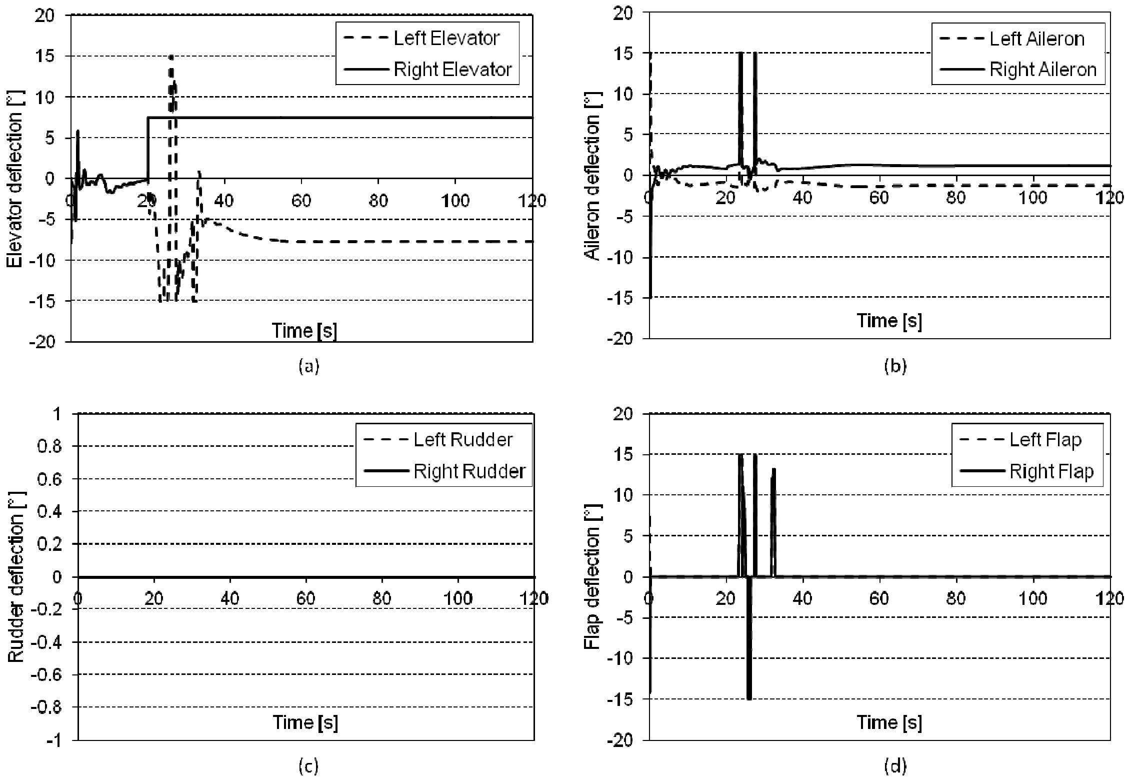


Figure 5.85: Simulated time history of elevator (a), aileron (b), rudder (c) and flap (d) deflections with the adaption mechanism enabled for the aircraft in a constant bank turn with a 50% hard over failure of the right elevator and an increased pitch PID adaption rate

Right Elevator Failure - 100% Hard Over (Test 3)

The simulation was run on this occasion inducing a 100% hard over elevator failure causing a nose-up pitching moment coupled with a roll to the right. The higher pitch PID adaption setting as determined in the 50% hard over failure was maintained for all of the simulations which follow, unless otherwise stated. Figure 5.86 (a) shows that the adaption algorithm performed poorly in re-establishing the desired airspeed value. The figure shows a large oscillation immediately after the failure is induced followed by an attempt to regain airspeed control, after which there is a large divergence in airspeed. This finally settles to the set point value 100 seconds after the failure has occurred. In comparison, the non-adapted case has re-established the set point value, after approximately 40 seconds. Thus there is a large degradation in control performance when the adaption algorithm is enabled. Figure 5.86 (b) shows slightly better behaviour for altitude than airspeed although erratic behaviour of altitude control is noted. While Figure 5.86 (a) and (b) do show erratic behaviour, the aircraft was not lost. The change in heading was lower with the adaption algorithm enabled.

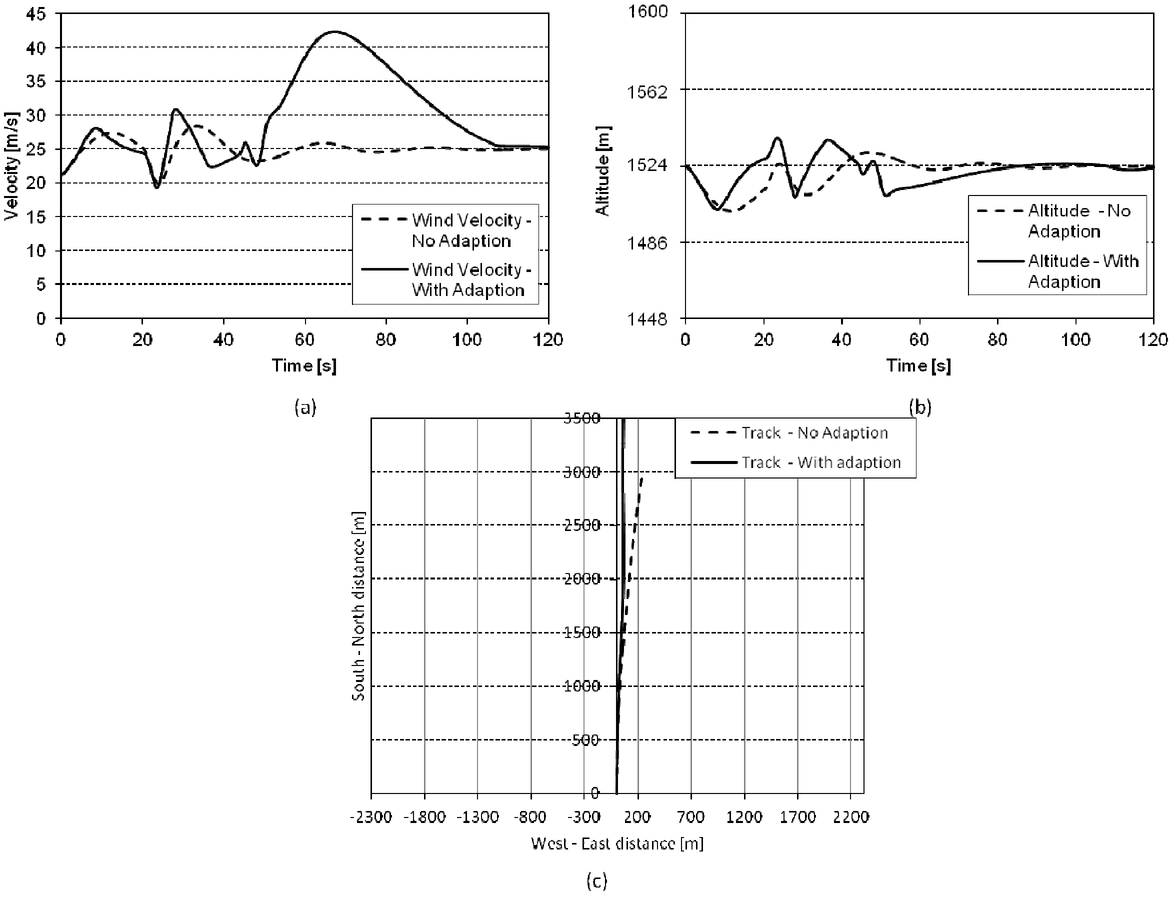
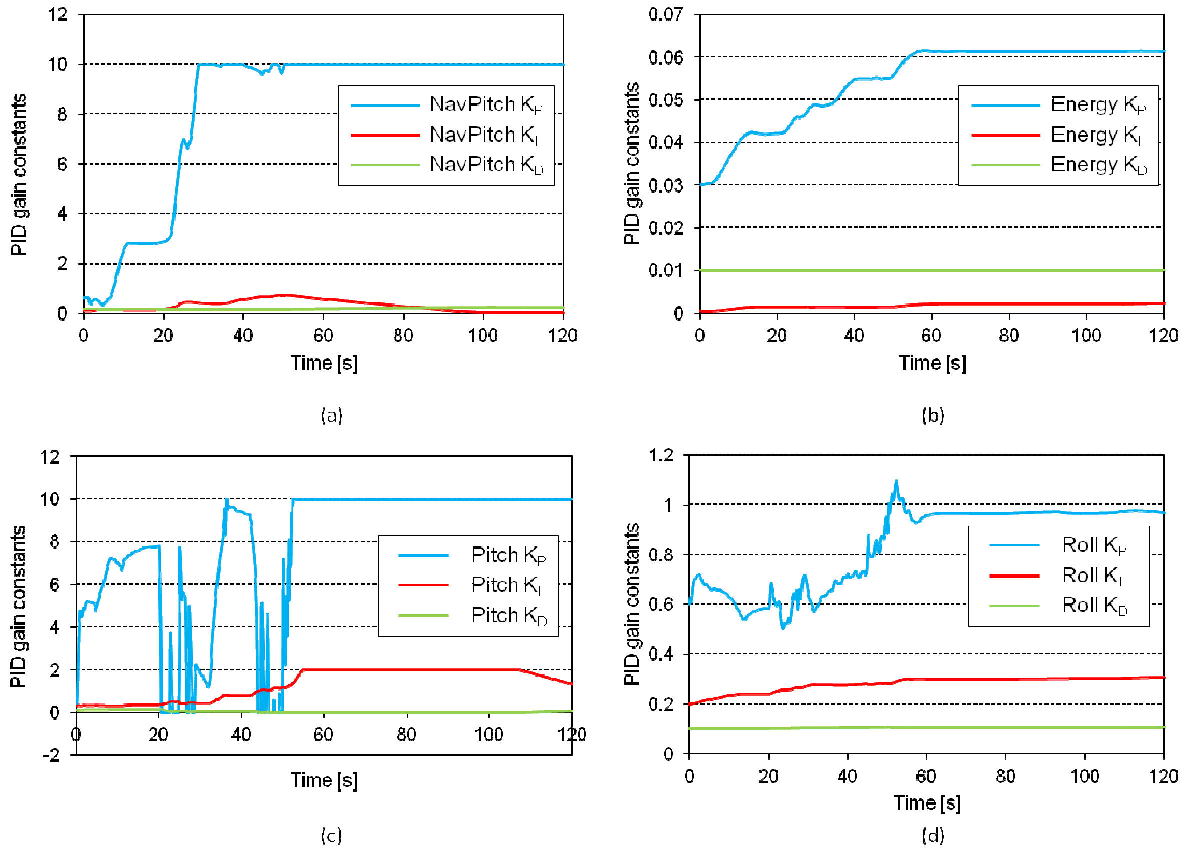


Figure 5.86: Simulated time history of airspeed (a), altitude (b) and the ground track (c) of the aircraft in a constant bank turn with the adaption algorithm enabled and disabled with a 100% hard over failure of the right elevator but with an increased pitch PID adaption rate

The cause of the erratic behaviour can be seen in Figure 5.87 which shows the saturation of both airspeed and pitch proportional gains. This was expected based on the concerns raised in the 50% hard over failure tests. The energy proportional gain is reasonably well behaved, although it more than doubles. The roll PID proportional gain was found also to be erratic although, once a more stable flight path was established, this erratic behaviour subsided. In all but Figure 5.87 (a), there is a general increase in integral gain with little to no change in derivative gain.



(a)

(b)

(c)

(d)

Figure 5.87: Simulated time history of the PID gain changes with the adaption mechanism enabled for the aircraft in a constant bank turn with a 100% hard over failure of the right elevator and an increased pitch PID adaption rate

Upon examination of Figure 5.88, it was noted that there was a large divergence in measured pitch to ideal pitch and, for a large portion of time, the maximum allowable pitch angle was commanded. It is thought that this is ultimately the mechanism that returned the aircraft to normal flight control. Had this limit not been enforced, it is thought that the aircraft may have become irrecoverable. Figure 5.89 shows that there was a large over control of pitch in that both flaps and ailerons were used to control pitch. The flaps are the secondary control surface for the pitch direction while the ailerons are the tertiary control surface. Up until this point, the use of the tertiary surface has not yet been seen.

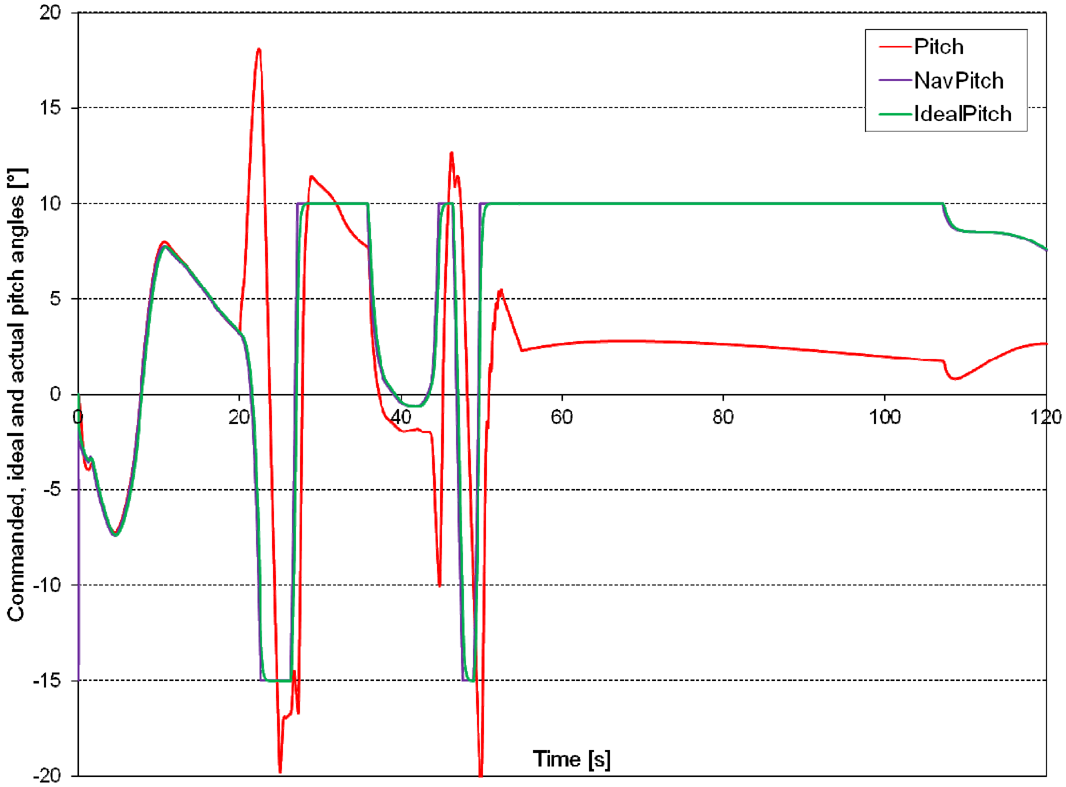


Figure 5.88: Simulated time history of the commanded pitch, ideal pitch and actual pitch for the aircraft in level flight, with a 100% hard over failure of the right elevator but with reduced airspeed PID adaption rate

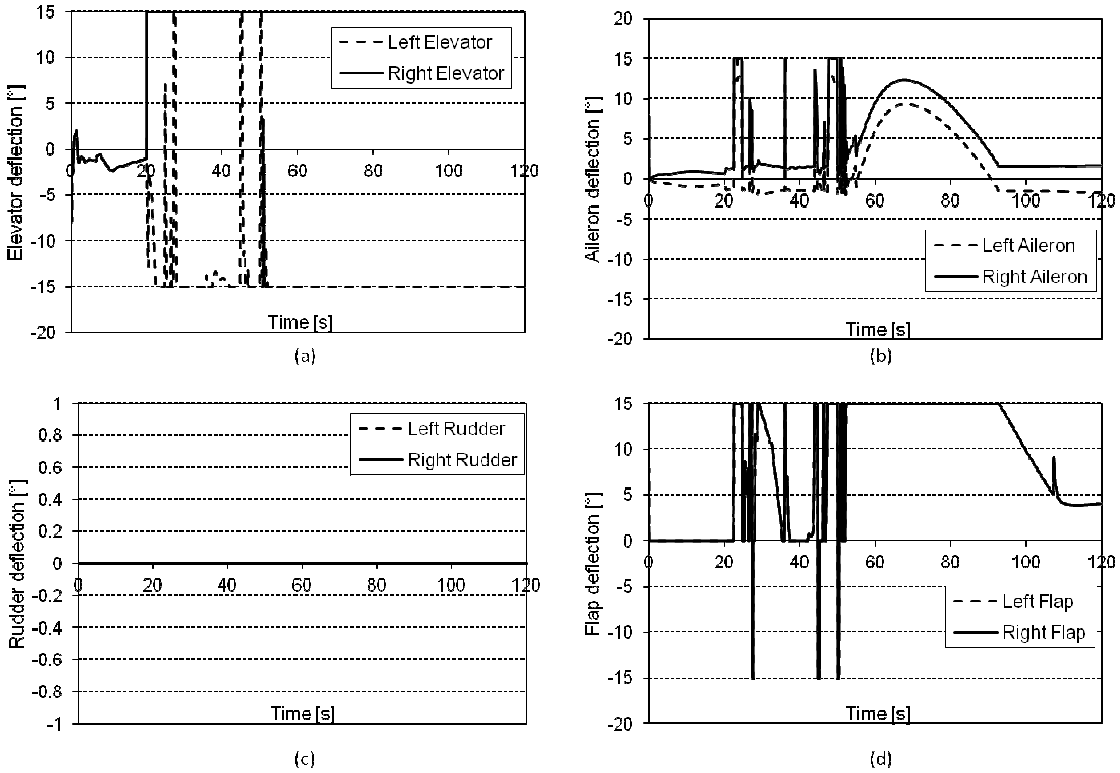


Figure 5.89: Simulated time history of elevator (a), aileron (b), rudder (c) and flap (d) deflections with the adaption mechanism enabled for the aircraft in a constant bank turn, with a 50% hard over failure of the right elevator, and an increased pitch PID adaption rate

Right Aileron Failure - 0° Deflection (Test 4)

The simulation was run inducing an aileron failure that moved the aileron to the neutral position 20 seconds into the simulation. As mentioned in the elevator failure cases, the higher pitch adaption settings were maintained. Only minor differences were observed, shown in Figure 5.90 (a) to (c) when compared to the control case. However, it was noted that there was a small disturbance in roll and pitch angles as seen in Figure 5.91. The roll angle disturbance was almost identical to those seen in the inner loop simulations. This is largely due to the fact that, essentially, roll control is still only controlled by the inner loops. Hence, any roll disturbance will only be marginally affected by the middle control loops. It is noted that the pitch angles have significant differences compared to the inner loop simulations but only minor differences to the middle loop control simulation.

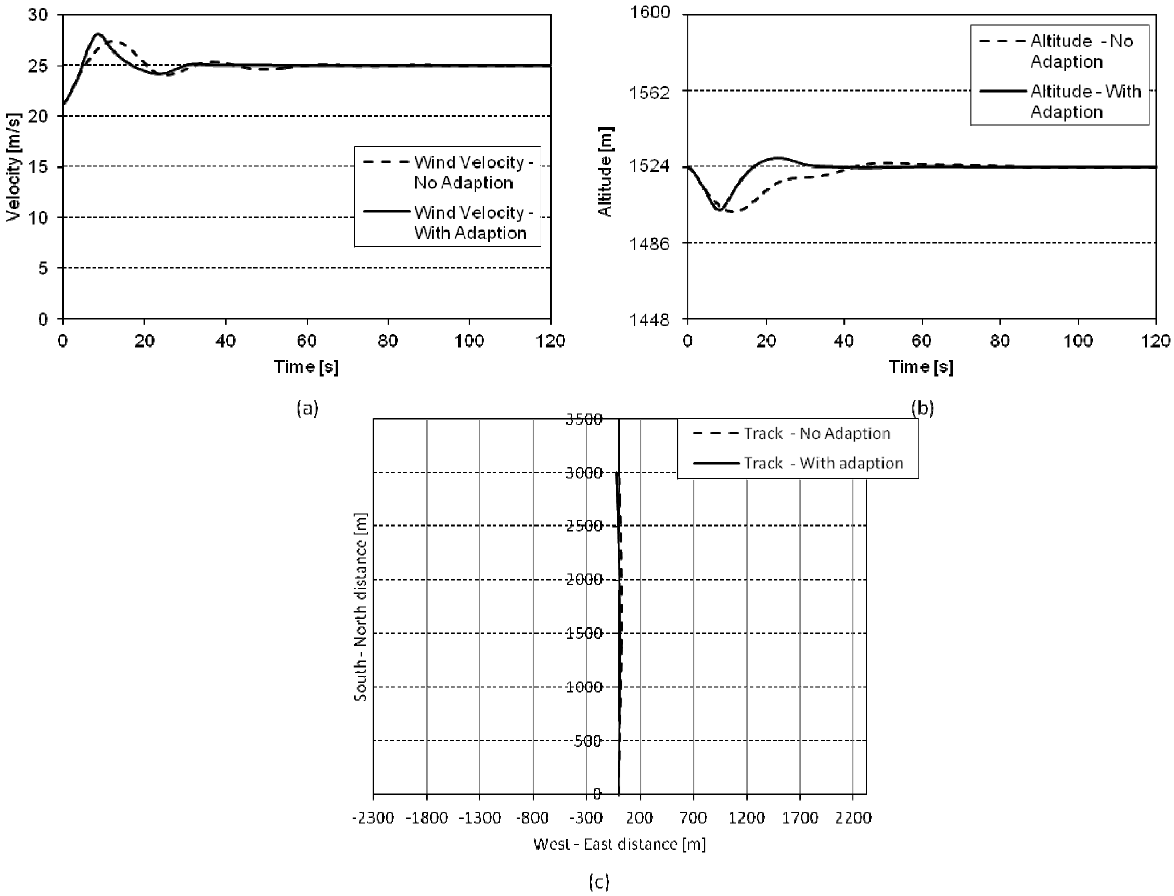


Figure 5.90: Simulated time history of airspeed (a), altitude (b) and the ground track (c) of the aircraft in level flight, with the adaption algorithm enabled and disabled with a 0° deflection failure of the right aileron but with an increased pitch PID adaption rate

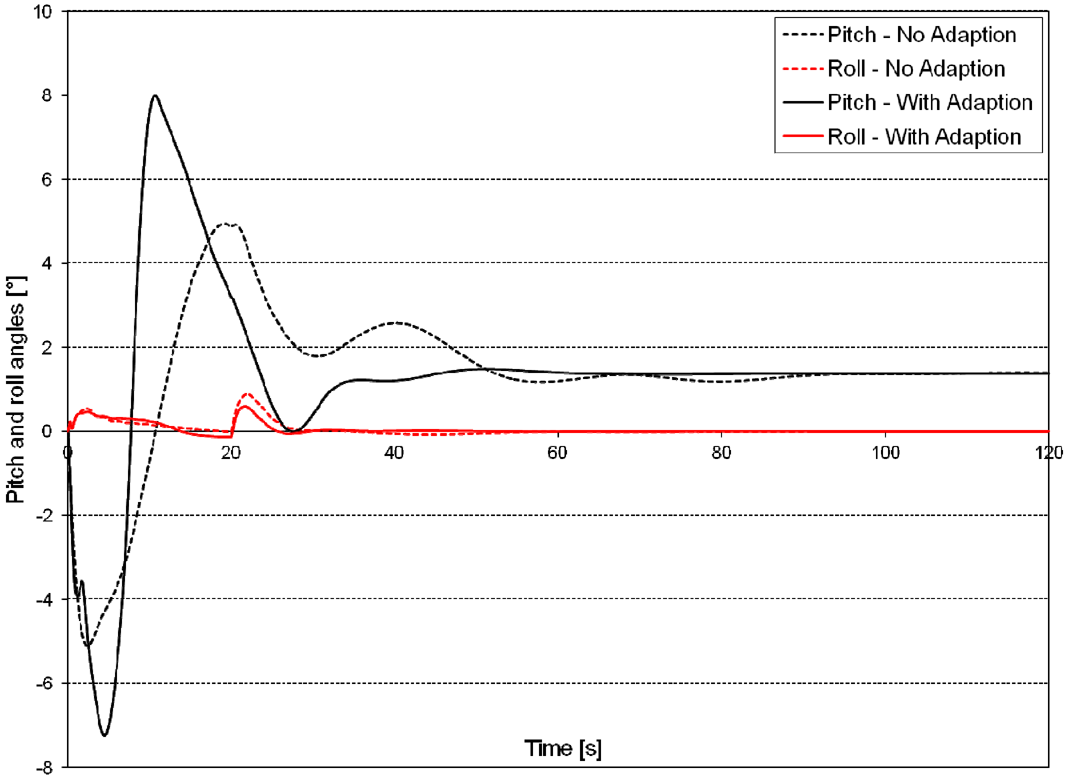
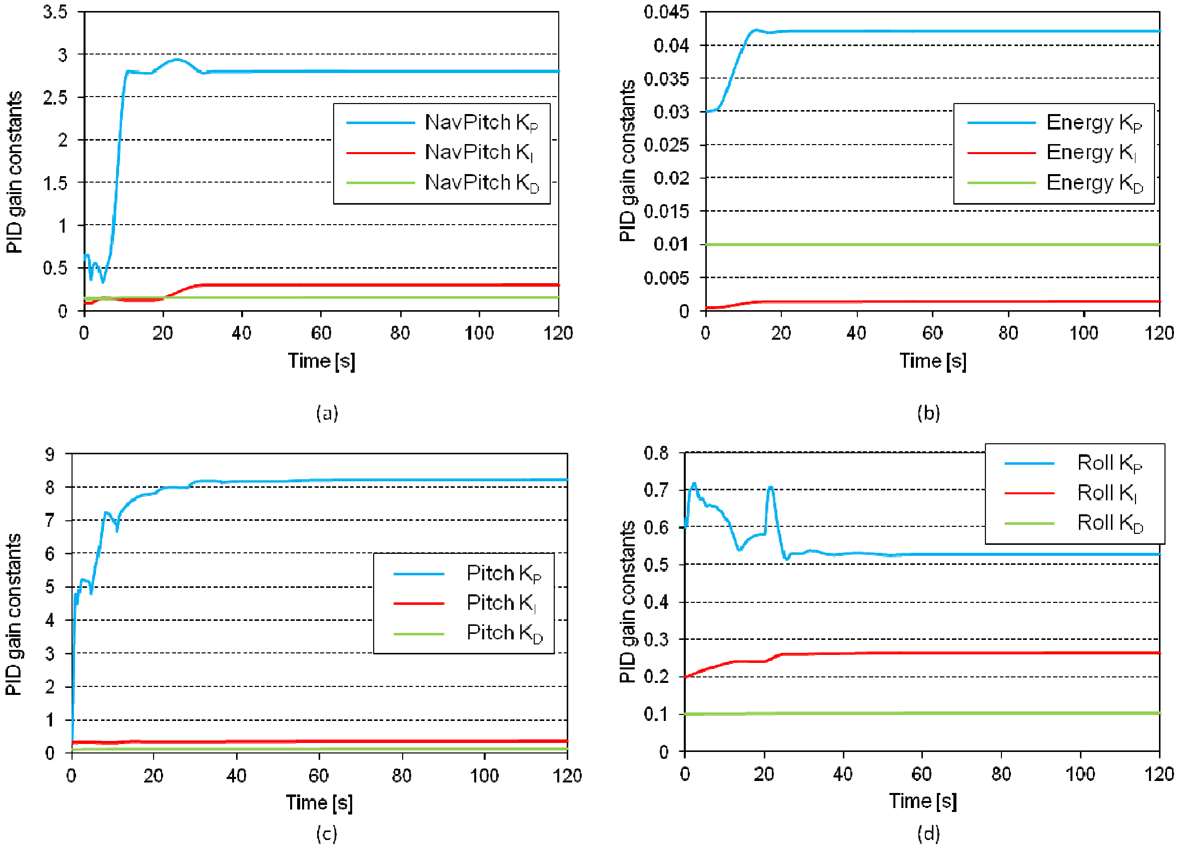


Figure 5.91: Simulated time history of pitch and bank angles (0° right aileron deflection failure) in level flight

The adaption algorithm generally improved the control performance, as was seen in the elevator failure cases. Of note, however, is the minor improvement in airspeed and altitude control with the higher pitch PID adaption rate without instabilities developing. It is noted however, that the pitch PID proportional gain rose to a fairly high value as seen in Figure 5.92 (c). The roll proportional gain showed a spike at the moment of failure followed by a reduction to levels below that set at the start of the simulation. There was also a slight increase in integral gain.

Little difference was observed in the fully banked case, and hence the results have not been included here; they can be found in Appendix C.



(a)

(b)

(c)

(d)

Figure 5.92: Simulated time history of the PID gain changes with the adaption mechanism enabled aircraft in level flight with the adaption algorithm enabled and disabled with a 0° deflection failure of the right aileron but with an increased pitch PID adaption rate

Right Aileron Failure - 50% Hard Over (Test 5)

The simulation was run inducing a 50% hard over failure of the right aileron causing a rolling moment to the right. Figure 5.93 (a) shows that the adaption algorithm was able to re-establish the set airspeed value and, although the overshoot was initially higher, the overshoot after the failure was lower and the time to return the airspeed to the set value was faster. Figure 5.93 (b) shows a much improved result in altitude tracking with less overshoot and a significantly faster re-establishment of the correct altitude. Finally, great improvement was seen in Figure 5.93 (c) where the change in heading was found to be considerably less with the adaption algorithm enabled. Again, the improved performance in these cases is primarily due to the inner loop roll control which has already been shown to have improved performance with adaption enabled. However, these results show that the adaption algorithm is functioning in the middle loops correctly.

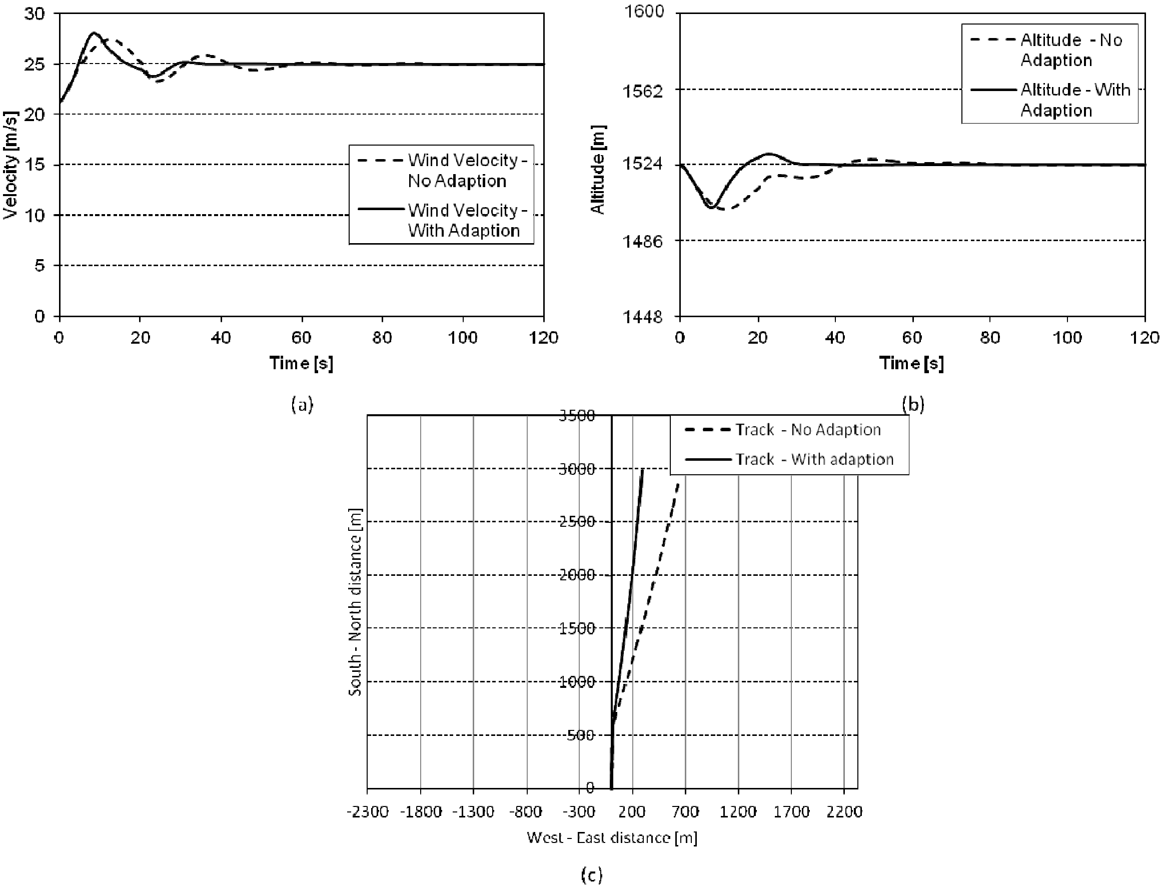
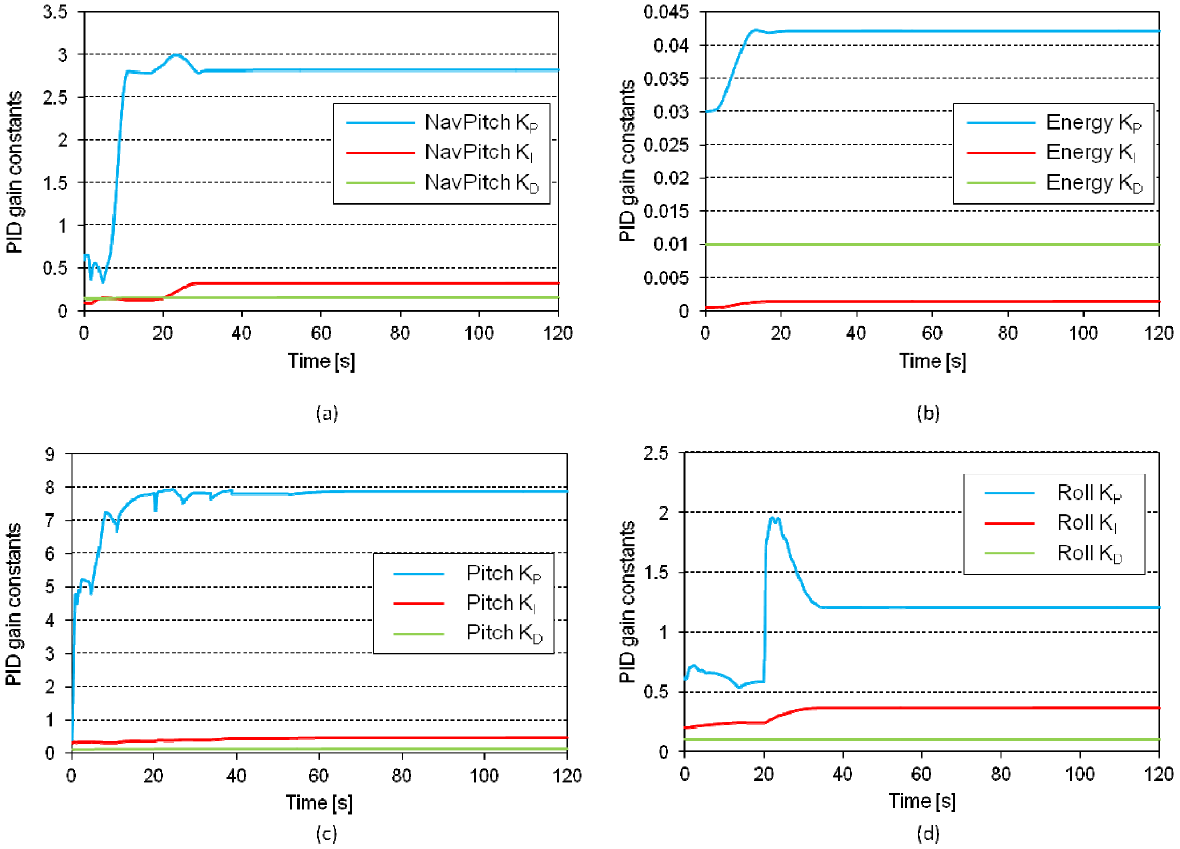


Figure 5.93: Simulated time history of airspeed (a), altitude (b) and the ground track (c) of the aircraft in level flight with the adaption algorithm enabled and disabled, with 50% hard over failure of the right aileron but with an increased pitch PID adaption

Figure 5.94 (c) again shows the rapid change in pitch proportional gain due to the high adaption rate set; however, the roll proportional gain, seen in Figure 5.94 (d), is reasonably well behaved with a peak forming at the moment of failure. The gain then reduces to a value still above the original base value. There is also an increase in integral gain which aids in returning the roll angle to the set value more quickly. The airspeed and energy gain constant show increases in both proportional gain and integral gain; however, only minor changes from the control simulation were observed.



(a)

(b)

(c)

(d)

Figure 5.94: Simulated time history of the PID gain changes with the adaption mechanism enabled aircraft in level flight, with the adaption algorithm enabled and disabled, with a 50% hard over failure of the right aileron, but with an increased pitch PID adaption

Similarly to the inner loop simulations, the maximum error seen in the bank angle after the failure was significantly reduced with the adaption algorithm (seen in Figure 5.95). The pitch angles again differ from the inner loop simulation cases but that is to be expected as a constant pitch angle is not being commanded. However, it is noted that the pitch angle reaches a steady-state value more quickly than the non-adapted case.

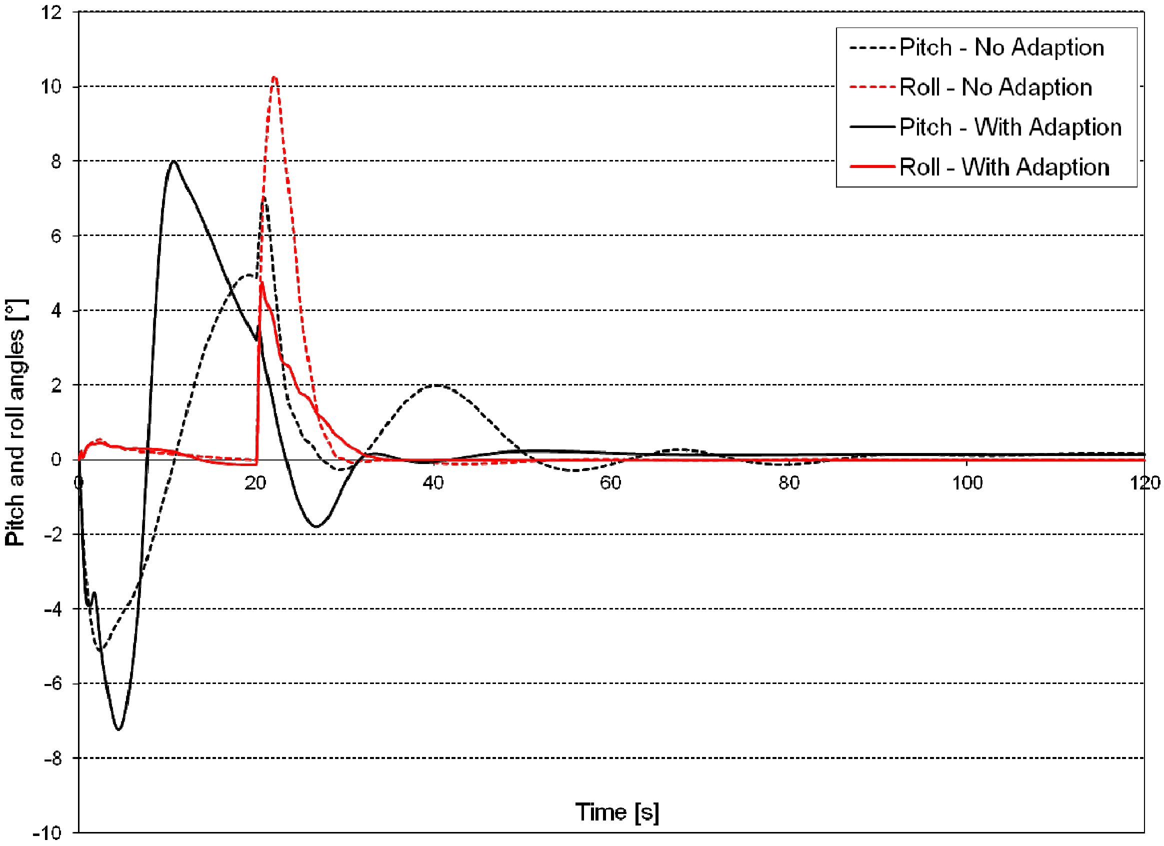


Figure 5.95: Simulated time history of pitch and bank angles (50% right aileron hard over failure) in level flight

The simulation was re-run but for the aircraft in a constant bank turn. Figure 5.96 (a) and (b) show similar results to the level flight case with the response of altitude, in particular, showing excellent performance. The ability of the adaption algorithm to correct the rolling moment more quickly also results in an improved ground track as shown in Figure 5.96 (c). At the moment of failure, it is observed that the flight path is diverted somewhat but little other effect is noted.

Similar results were obtained for the changes in PID gains seen in Figure 5.97 (a) to (c) while the roll proportional gain, shown in Figure 5.97 (d), showed that the proportional gain did not reduce in value after the failure as with the level flight case but instead remained at a higher level. Little other change was seen in the remaining results and thus they have not been included here. (See Appendix C)

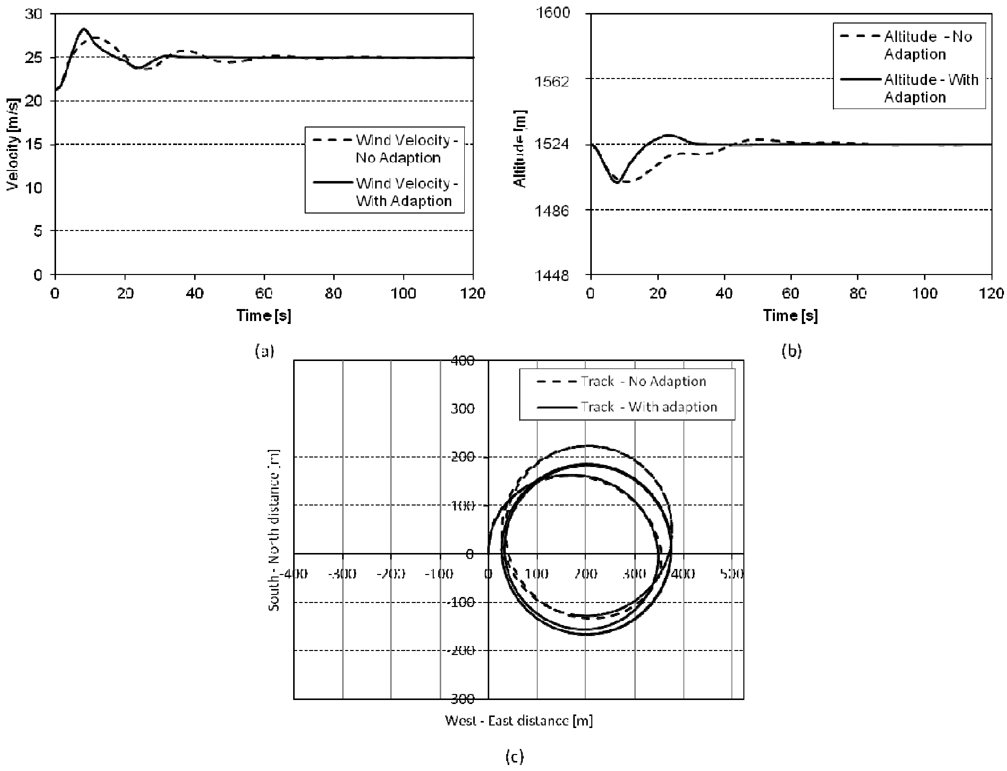


Figure 5.96: Simulated time history of airspeed (a), altitude (b) and the ground track (c) of the aircraft in a constant bank turn with the adaption algorithm enabled and disabled with a 50% hard over failure of the right aileron but with an increased pitch PID adaption

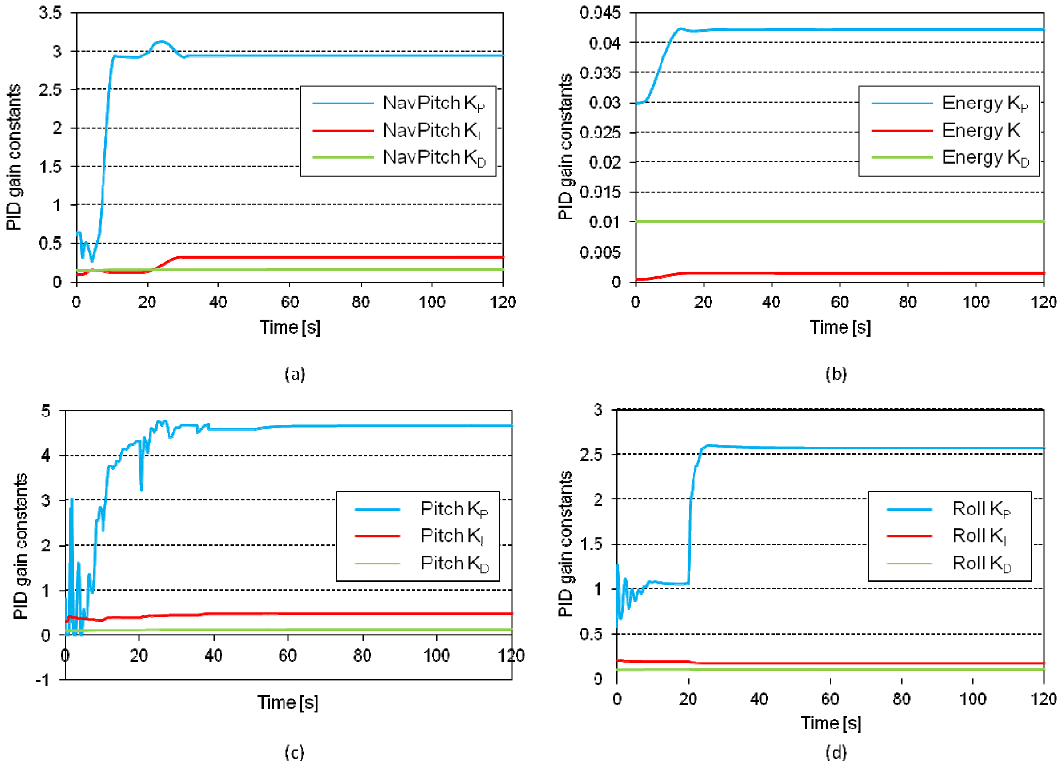


Figure 5.97: Simulated time history of the PID gain changes with the adaption mechanism enabled with the aircraft in a constant bank turn with the adaption algorithm enabled and disabled with a 50% hard over failure of the right aileron but with an increased pitch PID adaption

Right Aileron Failure - 100% Hard Over (Test 6)

The simulation was run inducing a 100% hard over failure of the right aileron. Similarly to the 50% hard over failure simulation, the results show that there was an improved control performance when the adaption algorithm was enabled. This, coupled with the use of the control allocation algorithm, allowed the aircraft to resume the desired flight path more quickly with a smaller change in heading, as seen in Figure 5.98 (c). Figure 5.99 (a) to (d) shows the importance of the control allocation algorithm in returning the aircraft to level flight as the flaps, Figure 5.99 (d), are used to roll the aircraft after the saturation of the ailerons, Figure 5.99(b). Without the allocation algorithm, an unsuccessful flight would have resulted. Figure 5.100, shows a similar trend to that observed in the 50% hard over case.

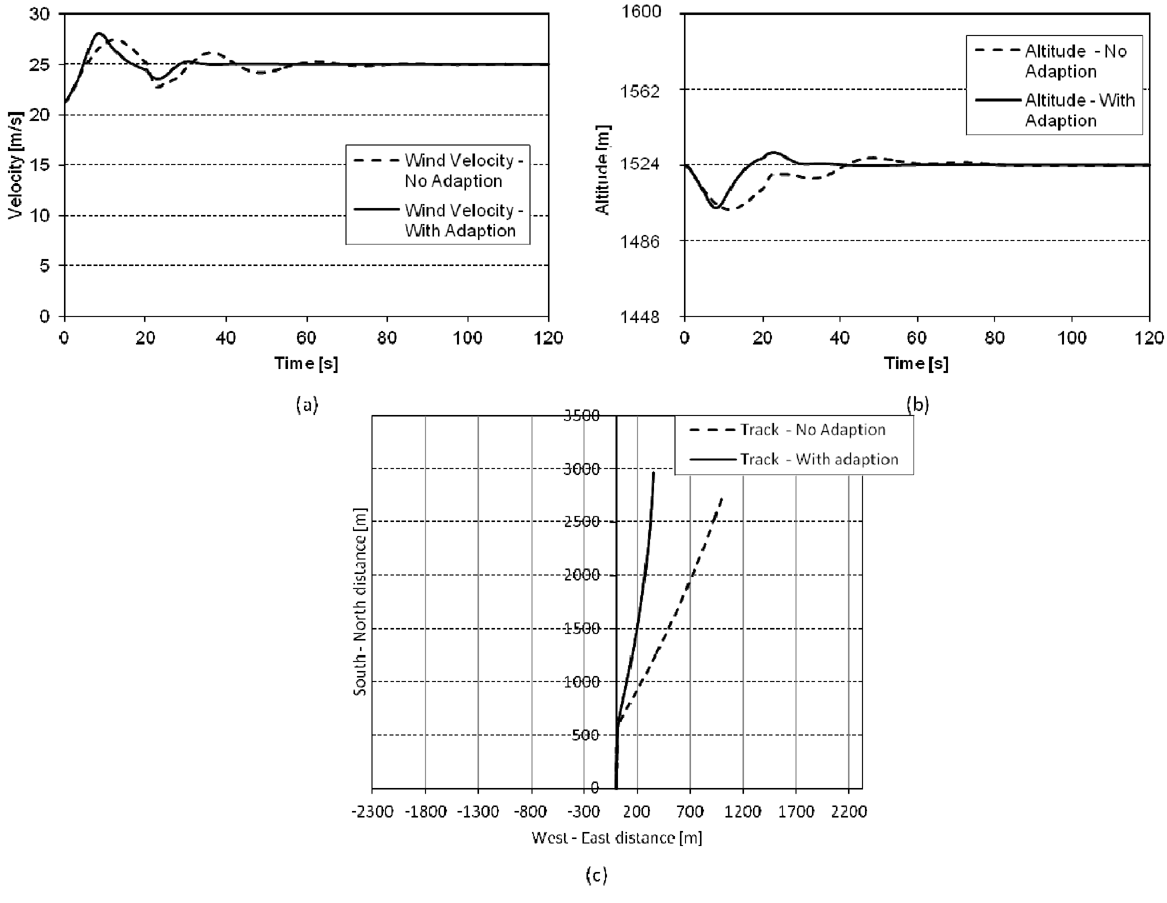
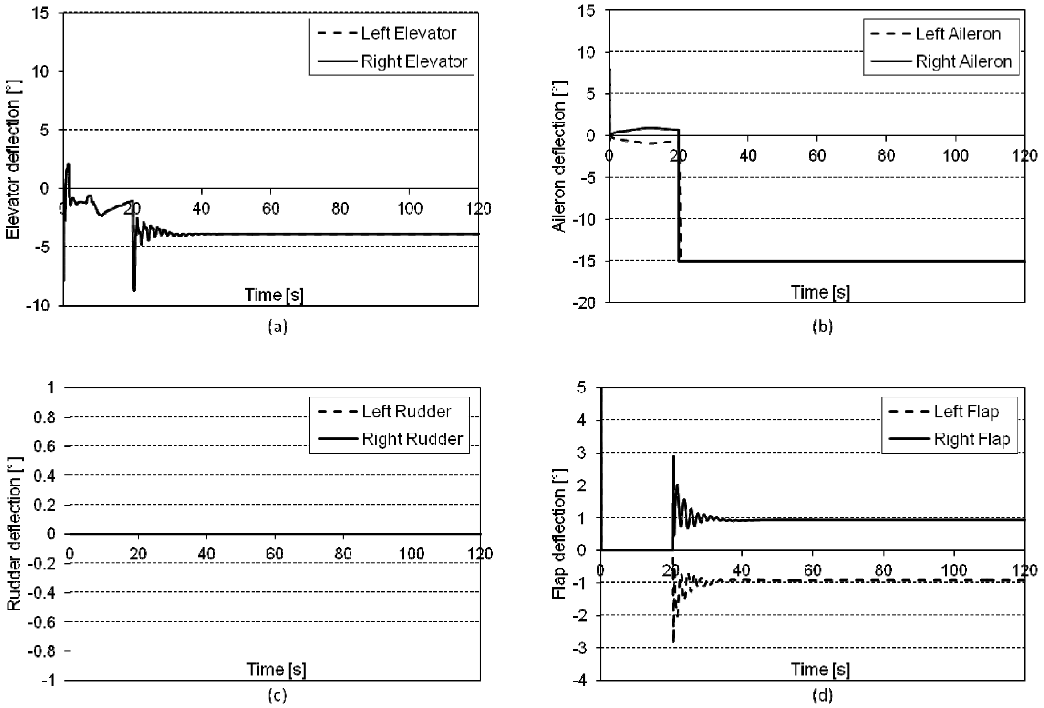


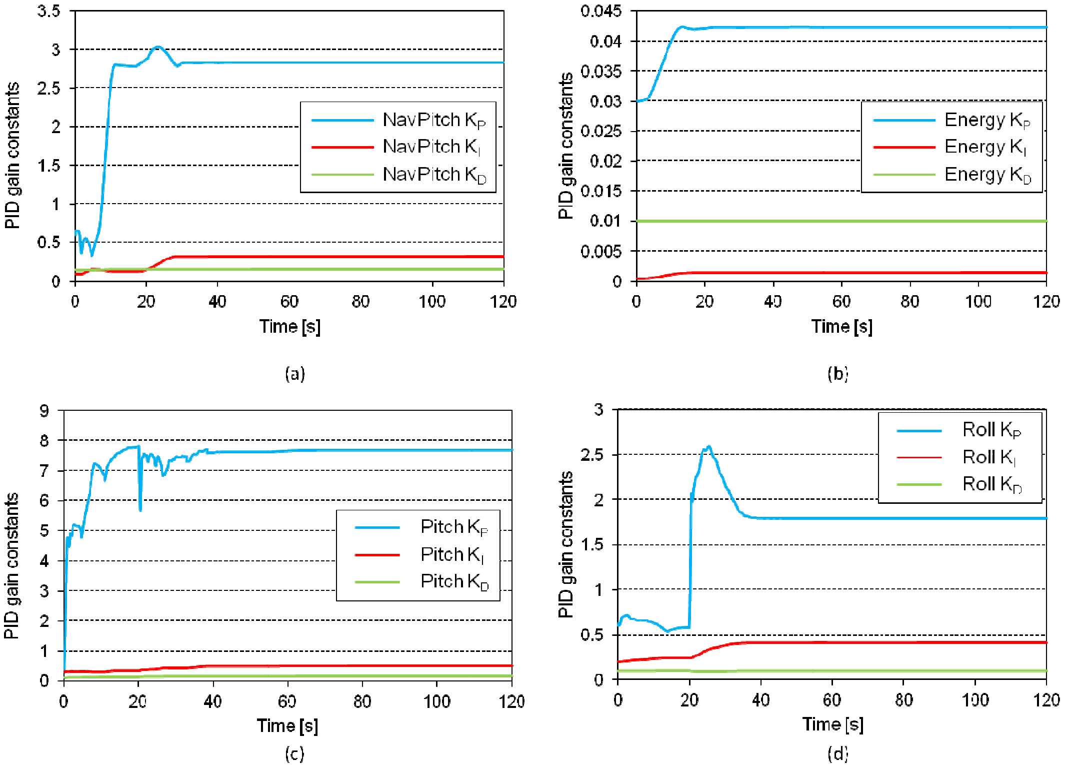
Figure 5.98: Simulated time history of airspeed (a), altitude (b) and the ground track (c) of the aircraft in level flight with the adaption algorithm enabled and disabled, with 100% hard over failure of the right aileron but with an increased pitch PID adaption



(a) (b)

(c) (d)

Figure 5.99: Simulated time history of elevator (a), aileron (b), rudder (c) and flap (d) deflections with the adaption mechanism enabled for the aircraft in a constant bank turn with a 100% hard over failure of the right elevator and an increased pitch PID adaption rate



(a) (b)

(c) (d)

Figure 5.100: Simulated time history of the PID gain changes with the adaption mechanism enabled with the aircraft in a constant bank turn with the adaption algorithm enabled and disabled with a 100% hard over failure of the right aileron but with an increased pitch adaption rate

The simulation was re-run for the banked flight case, which showed almost identical results to those obtained in the 50% deflection case, with minor changes in ground track seen in Figure 5.101(c). The remaining results can be found in Appendix C, as no additional insight is gained.

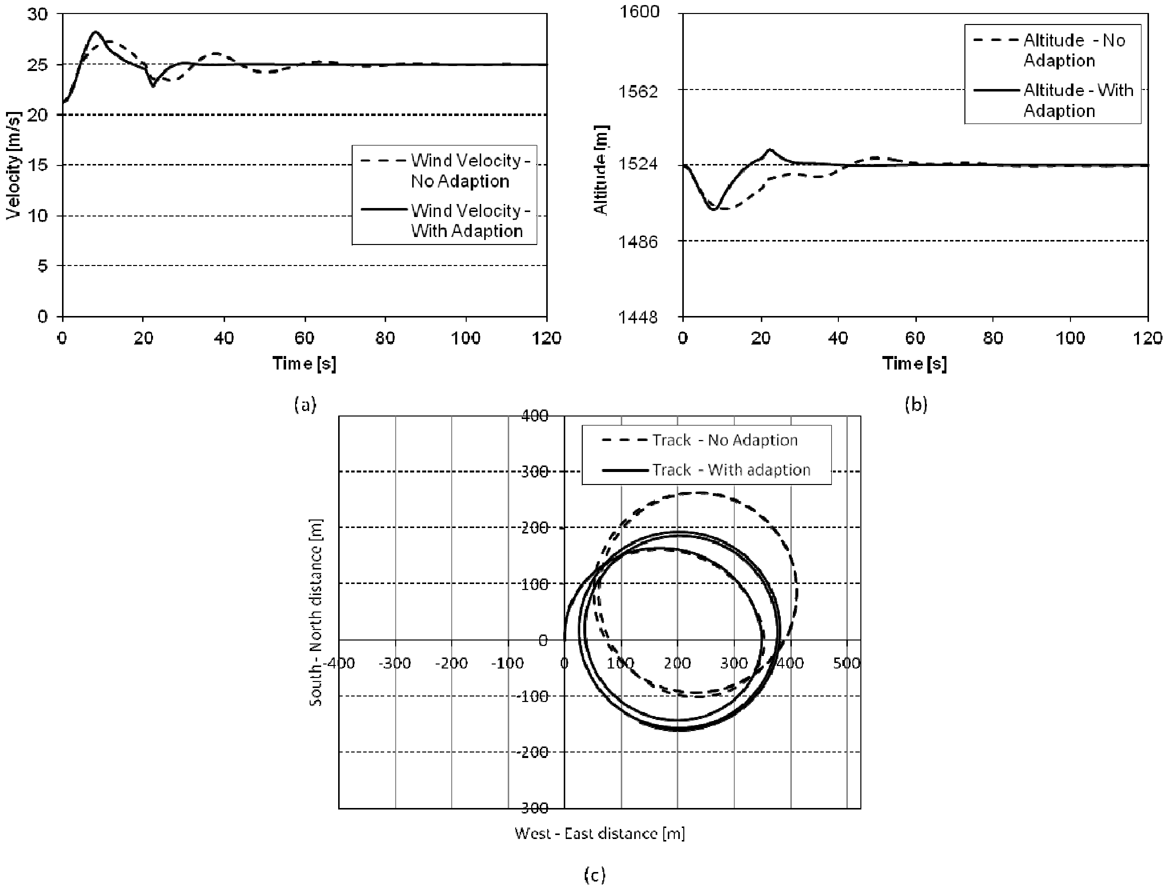


Figure 5.101: Simulated time history of airspeed (a), altitude (b) and the ground track (c) of the aircraft in a constant bank turn with the adaption algorithm enabled and disabled with a 100% hard over failure of the right aileron but with an increased pitch PID adaption

Comparison between Unmodified Autopilot and Autopilot with all Proposed Control Algorithms Enabled (Test 7)

A simulation was run inducing a combined failure of two control surfaces. A 50% hard over failure was applied to the right elevator and a 100% hard over failure was applied to the right aileron. In this test, a comparison between the unmodified autopilot (without the adaption and allocation algorithms) and the proposed control strategy are demonstrated. Figure 5.102 shows that the unmodified autopilot fails to maintain controlled flight with a rapid loss of altitude coupled with a rapid increase in airspeed and a spiral dive to the ground. In comparison, the proposed control strategy is able to tolerate this failure and, although there is a large upset in airspeed and altitude, the autopilot is able to re-establish the desired set points. The track shows a significant change in heading that is rapidly brought under control and straight and level flight is resumed.

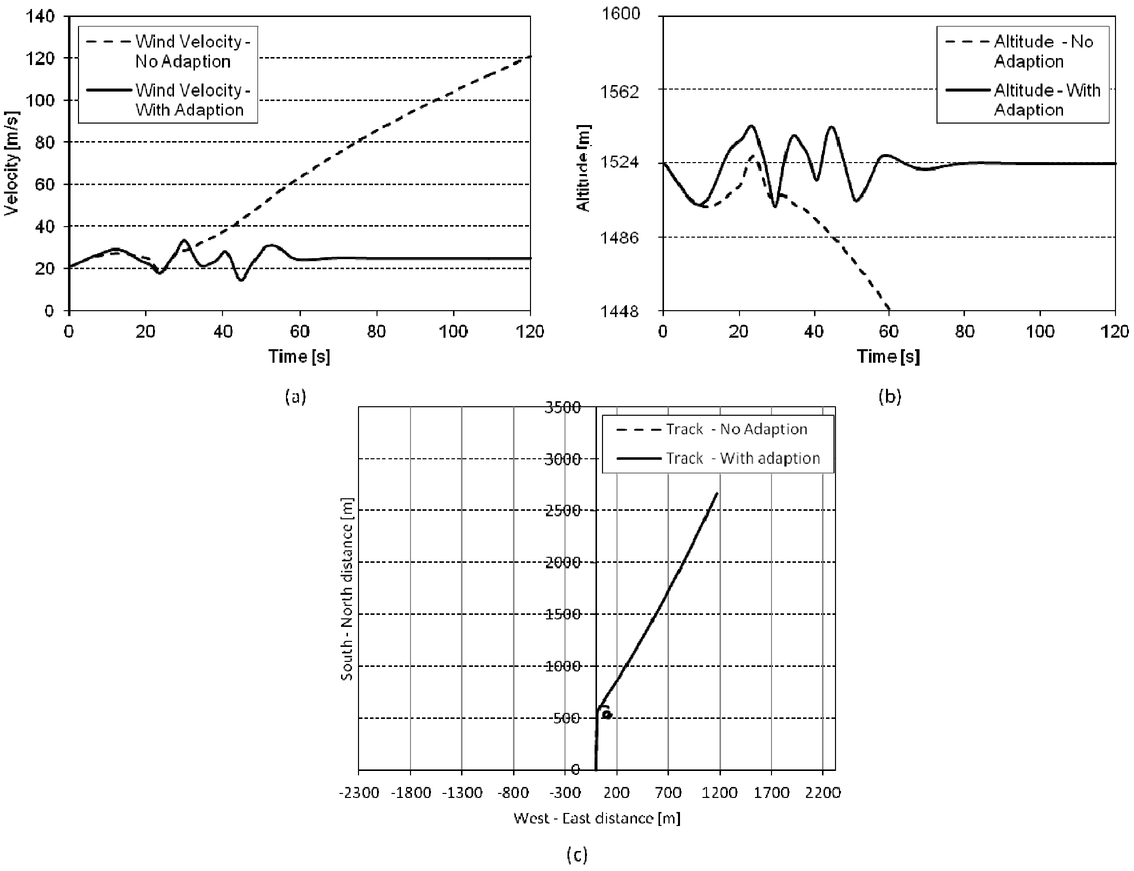


Figure 5.102: Simulated time history of airspeed (a), altitude (b) and the ground track (c) of the aircraft in level flight showing the difference between the un modified autopilot and the proposed control strategy with a combined failure of the right aileron and right elevator

The cause of this success can be seen in Figure 5.103 which shows the use of flap to control roll and to aid in pitch control when required. This is complemented by the adjustment of the PID gains for the inner loop and middle loop controllers as seen in Figure 5.104 (a) to (c).

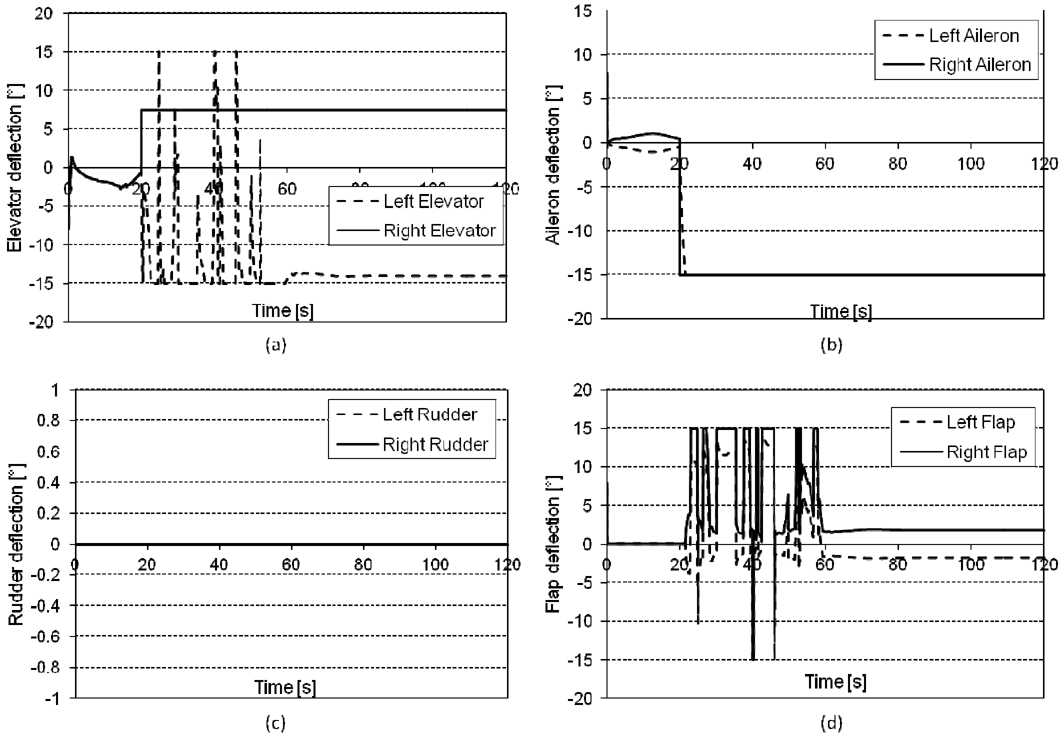


Figure 5.103: Simulated time history of elevator (a), aileron (b), rudder (c) and flap (d) deflections of the aircraft in level flight showing the difference between the un-modified autopilot and the proposed control strategy with a combined failure of the right aileron and right elevator

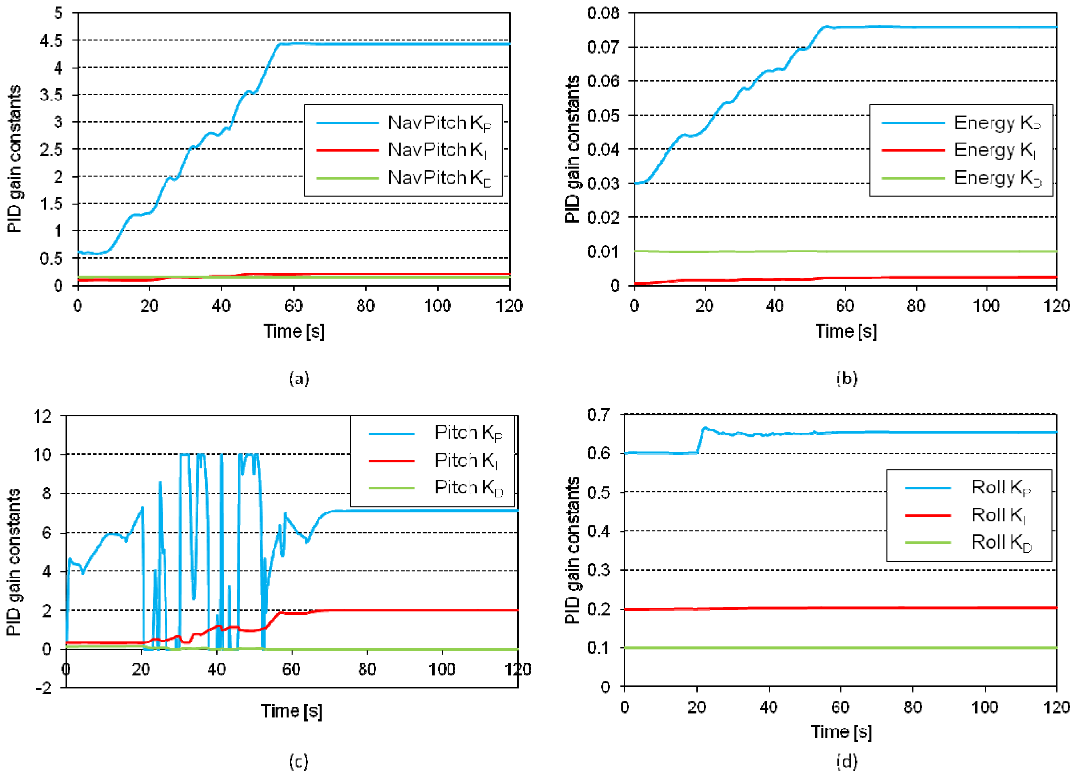
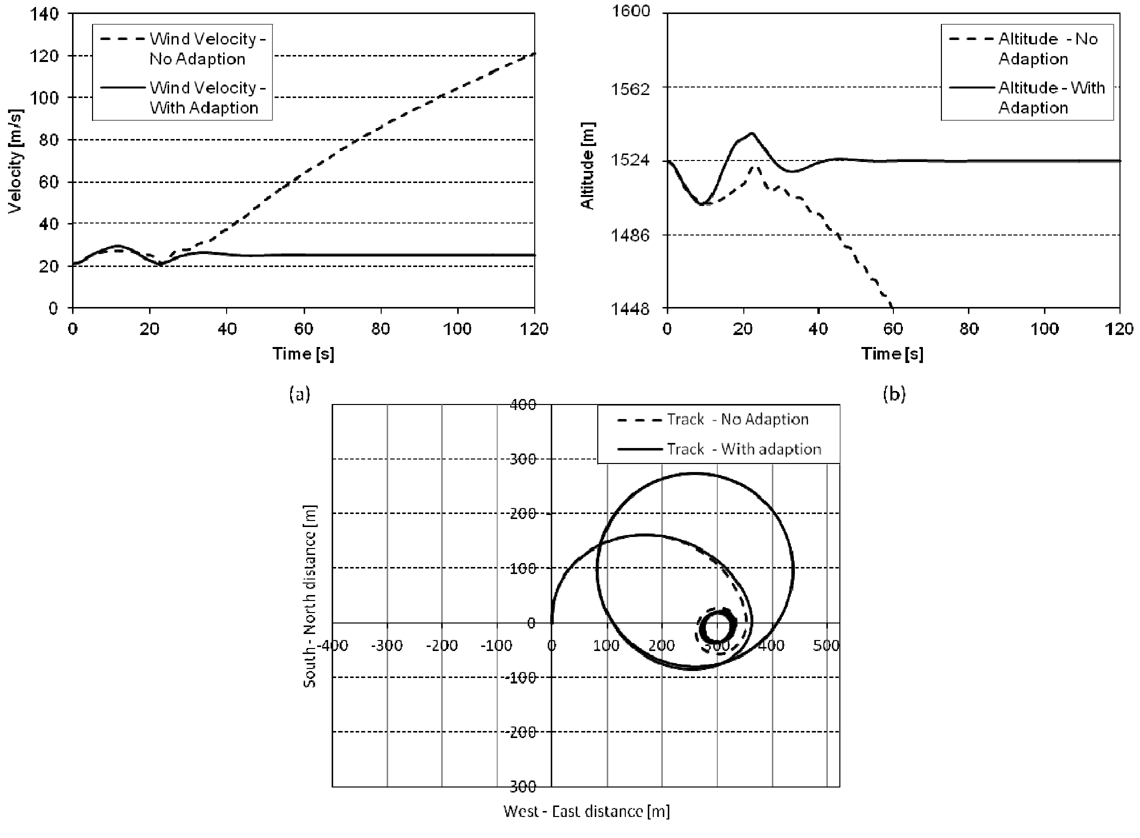


Figure 5.104: Simulated time history of the PID gain deflections of the aircraft in level flight showing the difference between the un-modified autopilot and the proposed control strategy with a combined failure of the right aileron and right elevator

A similar result was obtained for the constant bank turn flight case. The unmodified autopilot was unable to tolerate this failure, resulting in a spiral dive towards the ground and, ultimately the loss of the aircraft. With the proposed control strategy, the autopilot is able to tolerate this fault particularly well and excellent airspeed and altitude tracking are achieved as shown in Figure 5.105 (a) and (b). It is also noted that the flight path obtained is still circular and controlled, shown in Figure 5.105 (c).



(a)

(b)

(c)

Figure 5.105: Simulated time history of airspeed (a), altitude (b) and the ground track (c) for the aircraft in a constant bank turn showing the difference between the un-modified autopilot and the proposed control strategy with a combined failure of the right aileron and right elevator

5.5.3 Outer Control Loop Simulation

The adaption and allocation algorithms were tested in the fully autonomous flight mode and a set of three waypoints, the coordinates of which were arbitrarily chosen. The aircraft is initialised at a speed slightly below that set in the autopilot but at the correct altitude at the starting waypoint. The aircraft then flies to waypoint 1 which is approximately 400m from the initial way point and to the port side of the aircraft. The second waypoint is to the starboard side of the aircraft and is approximately 500m away. The last waypoint is the starting waypoint and thus a triangular flight path is intended. The waypoints are flown in turn until the simulation time has lapsed.

Control

A simulation, without inducing any control failures was run both with and without the adaption algorithms enabled. This allowed the establishment of a control that could be used to compare the performance of the autopilot system to the failures to be modelled. The control simulation also provided a test of the adaption system under normal operating conditions to determine whether the adaption system would cause instabilities to arise in the control of the aircraft.

During the simulations it was found that, for the current adaption settings of the inner loop roll PID controller, the simulation with the adaption algorithm enabled, failed. This failure was caused by too rapid a change in the roll PID controller and this induced large accelerations. This placed the aircraft out of the known flight envelope. The aerodynamic characteristics of the aircraft were described using a set of polynomials. Once the aircraft has exceeded the known flight envelope, the polynomials are no longer a reliable source of data and may produce unrealistic results. In this case, one of the aerodynamic parameters became large and a runaway condition, to infinite levels, developed. While this is not physically possible, it does show that the adaption algorithm would be likely to have exceeded the structural or aerodynamic capabilities of the aircraft, causing the loss of an airframe. The roll PID adaption rate was thus reduced by a factor of 10. Again this caused a divergence that resulted in the termination of the simulation. Again, the adaption rate was reduced by a factor of 10 and this simulation yielded a highly satisfactory response for the control of airspeed and altitude and a slight improvement in waypoint tracking.

In Figure 5.106, it can be seen that, with the adaption algorithm disabled, there is a large variation in airspeed each time a manoeuvre was induced, to the extent that a stable airspeed at the desired set point was never achieved. The simulation with the adaption algorithm, however, shows great improvements in control performance over the non-adapted case. Figure 5.106 shows that after an initial overshoot in airspeed, the airspeed quickly stabilises to the set point value and only very minor disturbances in airspeed during each manoeuvre. This variation is very slight and the set point value is quickly re-established.

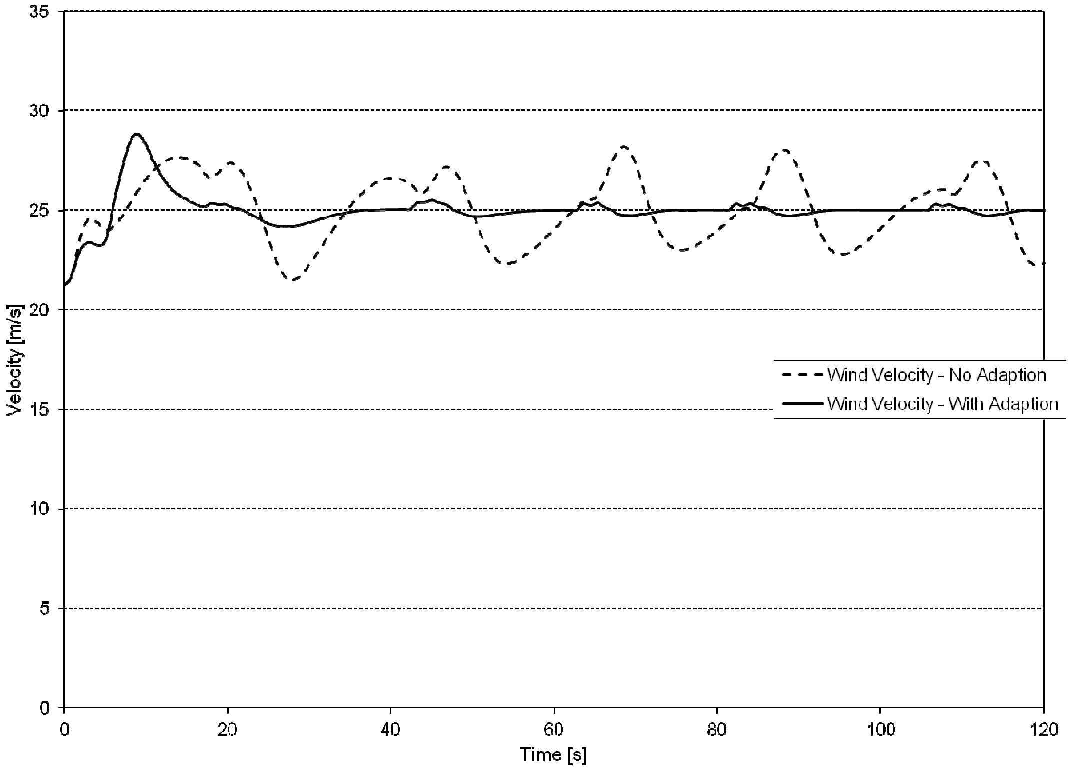


Figure 5.106: Simulated time history of airspeed of the aircraft in the fully autonomous flight mode with a high pitch adaption rate and a low roll adaption rate (no failures induced).

A similar set of results was obtained for the altitude control, (see Figure 5.107). During each manoeuvre there is a large variation in altitude for the non-adapted case. This error is highly reduced when the adaption algorithm is enabled. Similarly to airspeed, a minor change in altitude is noted, but it is of the order of only a few meters. The desired altitude is again very quickly re-established. Figure 5.108 shows the track followed by the aircraft during the simulations and shows a close correspondence to the tracks for the adapted and non-adapted cases initially. Improvement is seen after the first manoeuvre in that the turn radius with adaption enabled is reduced. This is seen during all changes in direction and shows the adaption algorithm is improving the waypoint following to some extent.

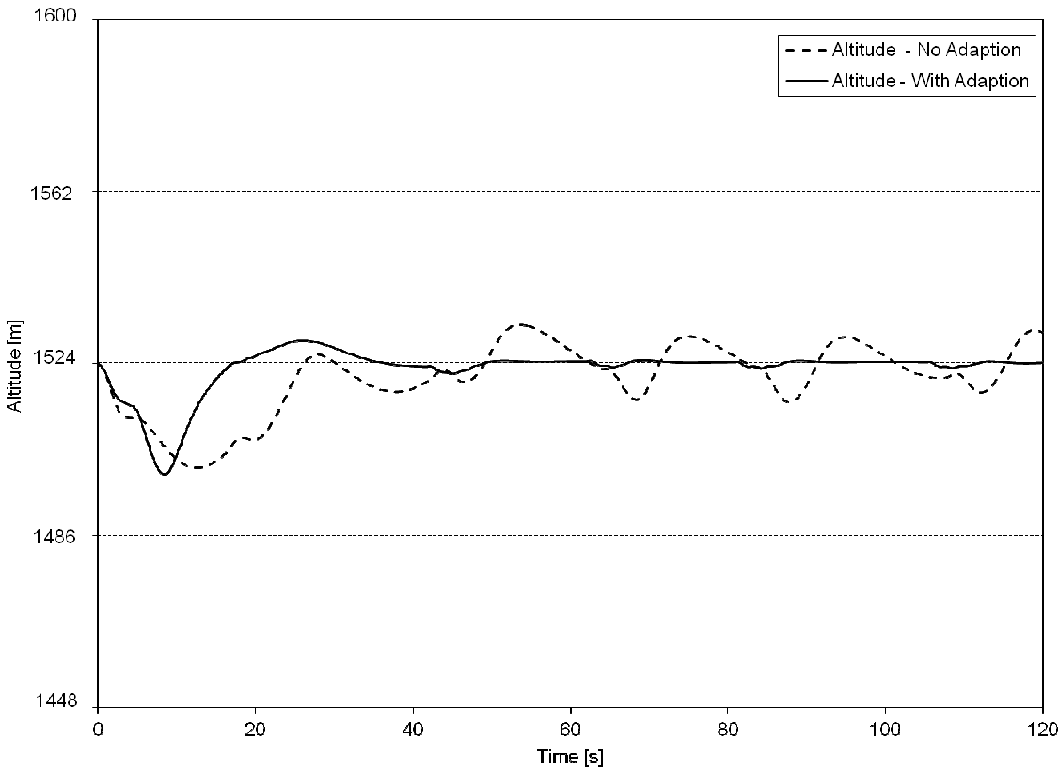


Figure 5.107: Simulated time history of altitude of the aircraft in the fully autonomous flight mode with a high pitch adaption rate and a low roll adaption rate (no failures induced).

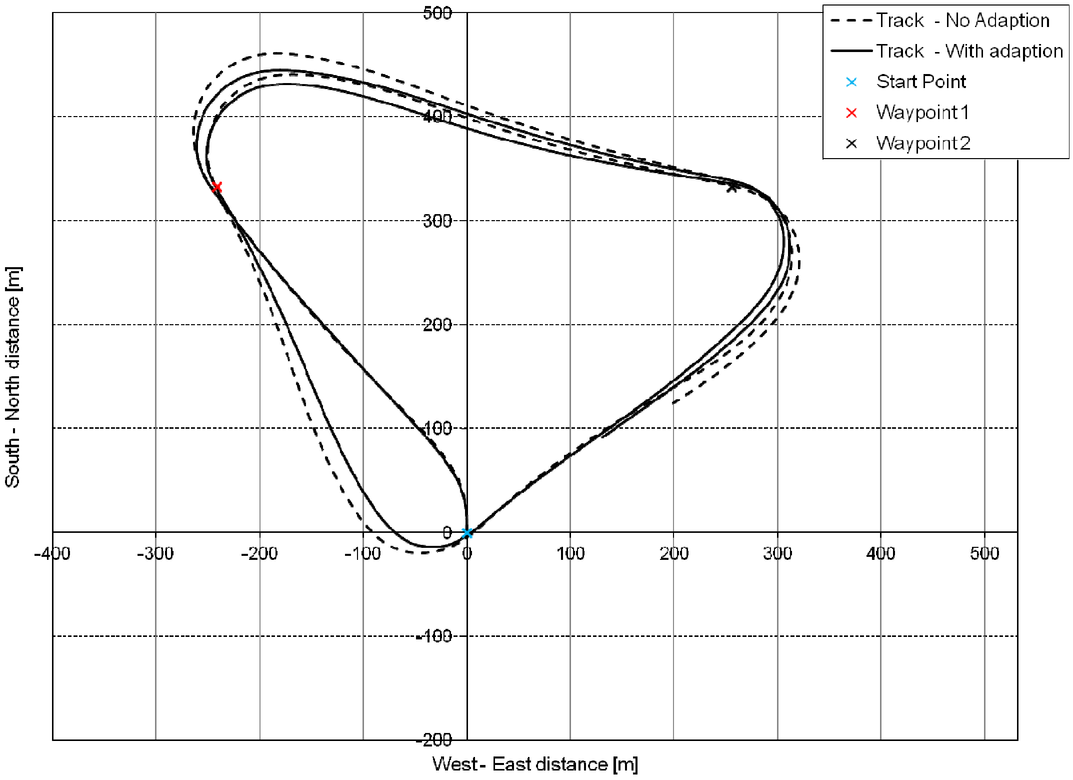
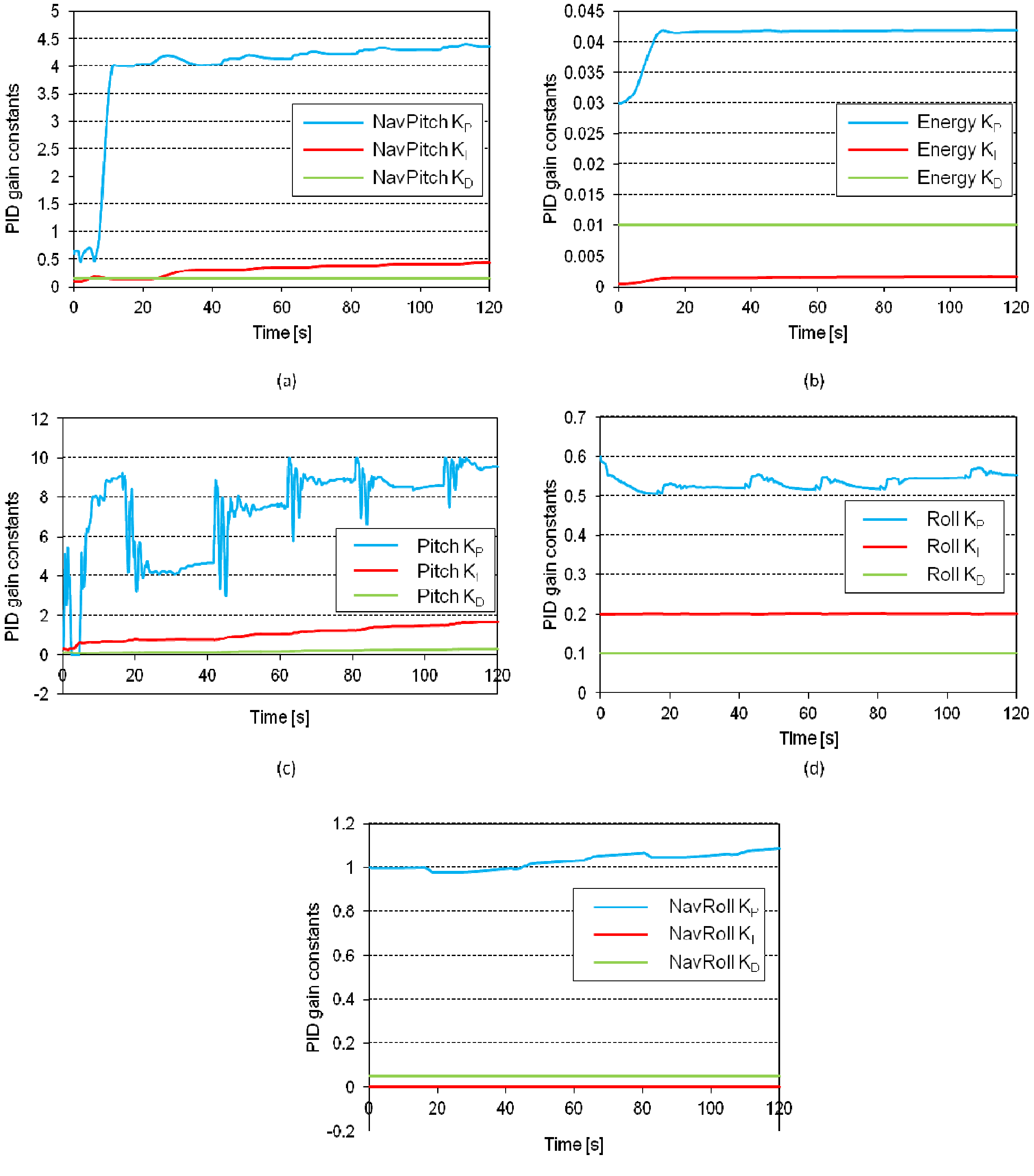


Figure 5.108: Simulated track followed by the aircraft in the fully autonomous flight mode with a high pitch adaption rate and a low roll adaption rate (no failures induced)

Figure 5.109 shows areas of concern. An examination of Figure 5.109 (a) shows that, initially, there is a large increase in the airspeed proportional gain followed by a stepped increase in this and the integral gain every time a manoeuvre is performed. This may result in an unstable condition developing should the flight continue. This concern is especially true of the pitch PID shown in Figure 5.109 (c), which has a very large increase in proportional gain followed by a rapid reduction but then an increase again, during each turn manoeuvre. The magnitude of the change does decrease during each successive manoeuvre, suggesting that the gain may reach an equilibrium point eventually. However, the change is still to increase the proportional gain. The energy, roll and navigational roll PID gains are reasonably well behaved and do increase slightly; however, it is noted that an equilibrium point, other than for the energy PID, is not attained.

Figure 5.110 shows that the measured pitch value oscillates around the ideal pitch value. This is thought to occur due to the high pitch PID proportional gain and the high frequency oscillations observed in Figure 5.109 (c). Towards the end of the manoeuvre, this oscillatory behaviour did reduce and the ideal pitch behaviour was more closely followed.



(a)

(b)

(c)

(d)

(e)

Figure 5.109: Simulated time history of the airspeed PID (a), energy PID (b), pitch PID (c), roll PID (d) and navigational roll PID (e) gain changes with the adaption mechanism enabled for fully autonomous flight

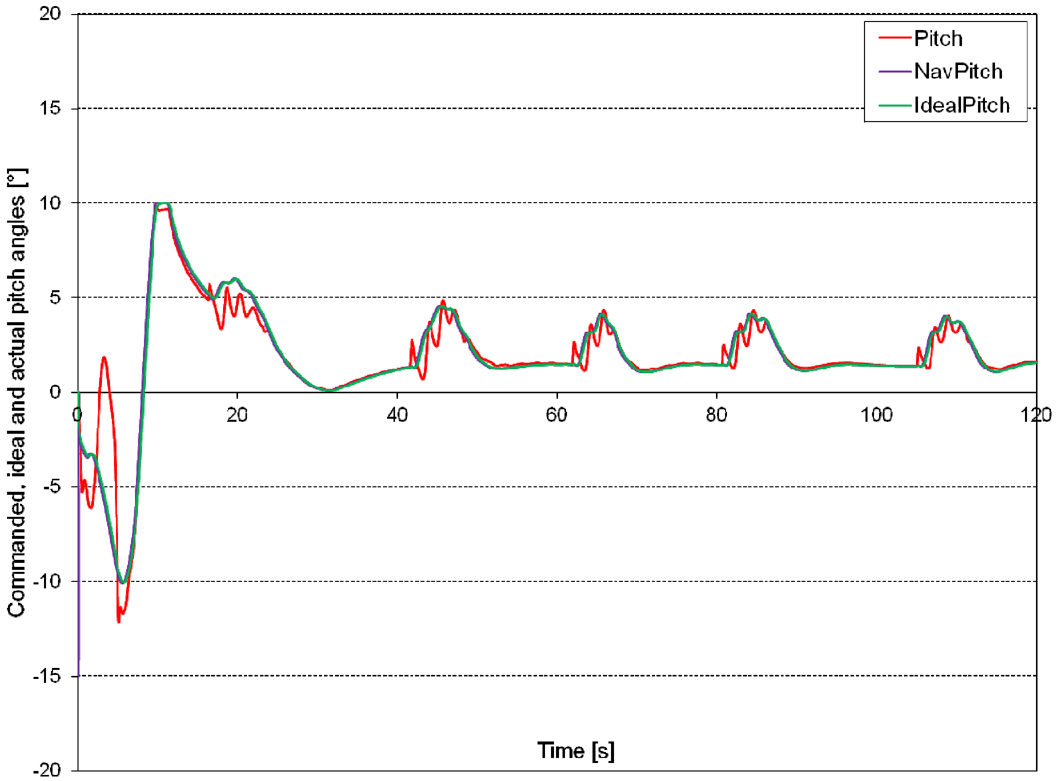


Figure 5.110: Simulated time history of the commanded pitch, ideal pitch and actual pitch attained for the fully autonomous simulation

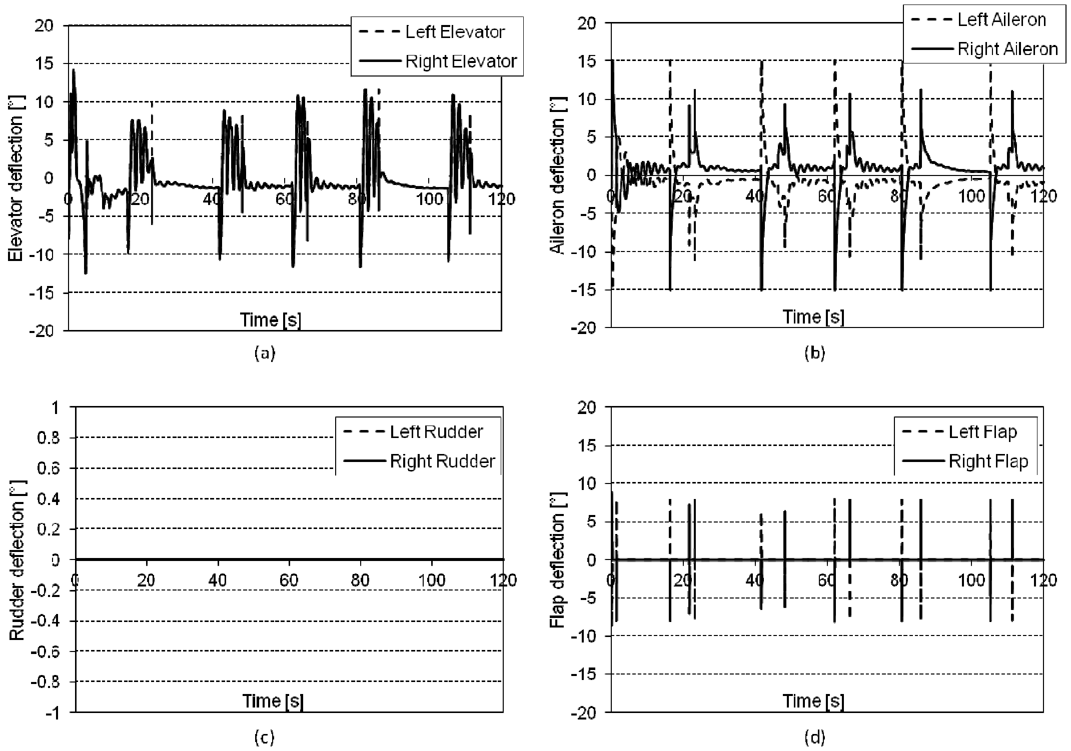


Figure 5.111: Simulated time history of the elevator (a), aileron (b), rudder (c) and flap (d) deflections of the aircraft in fully automated flight with the adaption algorithm enabled

Right Elevator Failure - 0° Deflection (Test 1)

The simulation was run with the aircraft in fully autonomous mode with a 0° deflection failure of the right elevator. Unfortunately, the results, shown in the figures that follow, indicate that the adaption algorithm was very sensitive to this failure and that the failure induced largely erratic behaviour. This can be seen in airspeed, Figure 5.112, and altitude, Figure 5.113, which both show this erratic behaviour. The non-adapted case did not yield much improved results; however, the results were more closely related to the results obtained in the control. The track was, however, largely unaffected by the airspeed and altitude disturbances and the improved waypoint following of the adaption algorithm was maintained as seen in Figure 5.114.

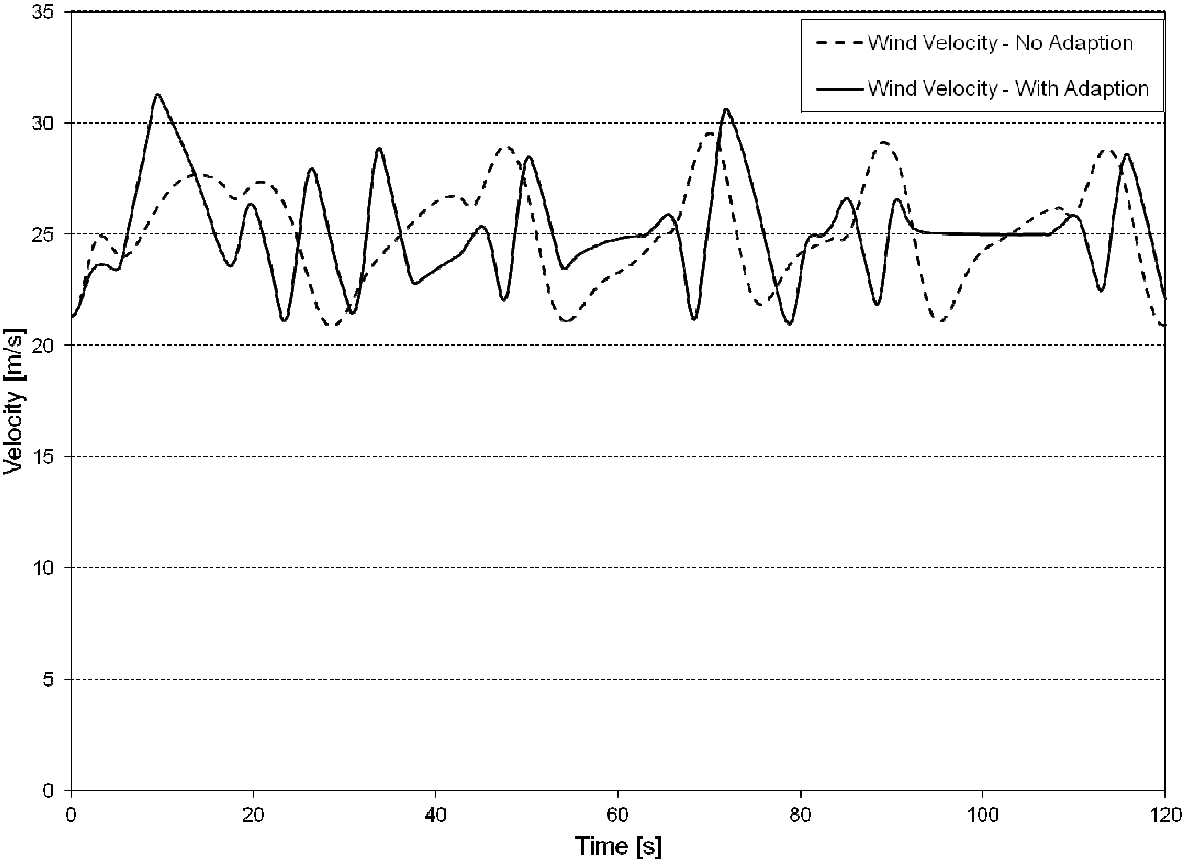


Figure 5.112: Simulated time history of airspeed of the aircraft in the fully autonomous flight mode with a high pitch adaption rate and a low roll adaption rate (0° deflection failure of the right elevator)

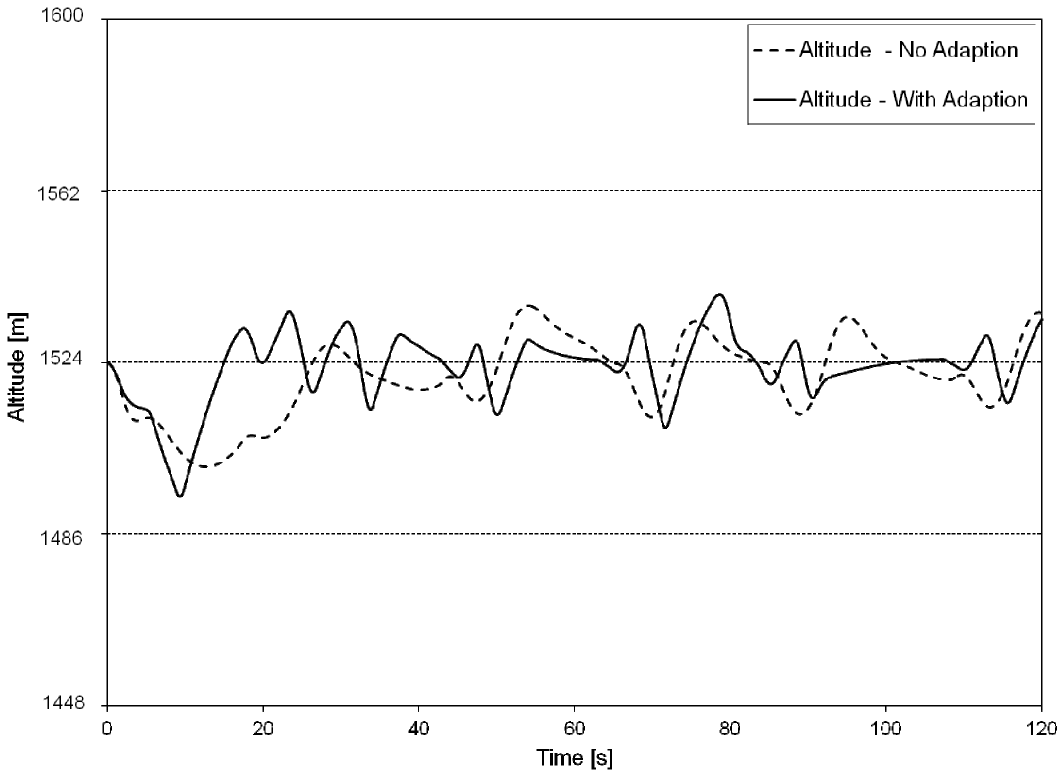


Figure 5.113: Simulated time history of airspeed of the aircraft in the fully autonomous flight mode with a high pitch adaption rate and a low roll adaption rate (0° deflection failure of the right elevator)

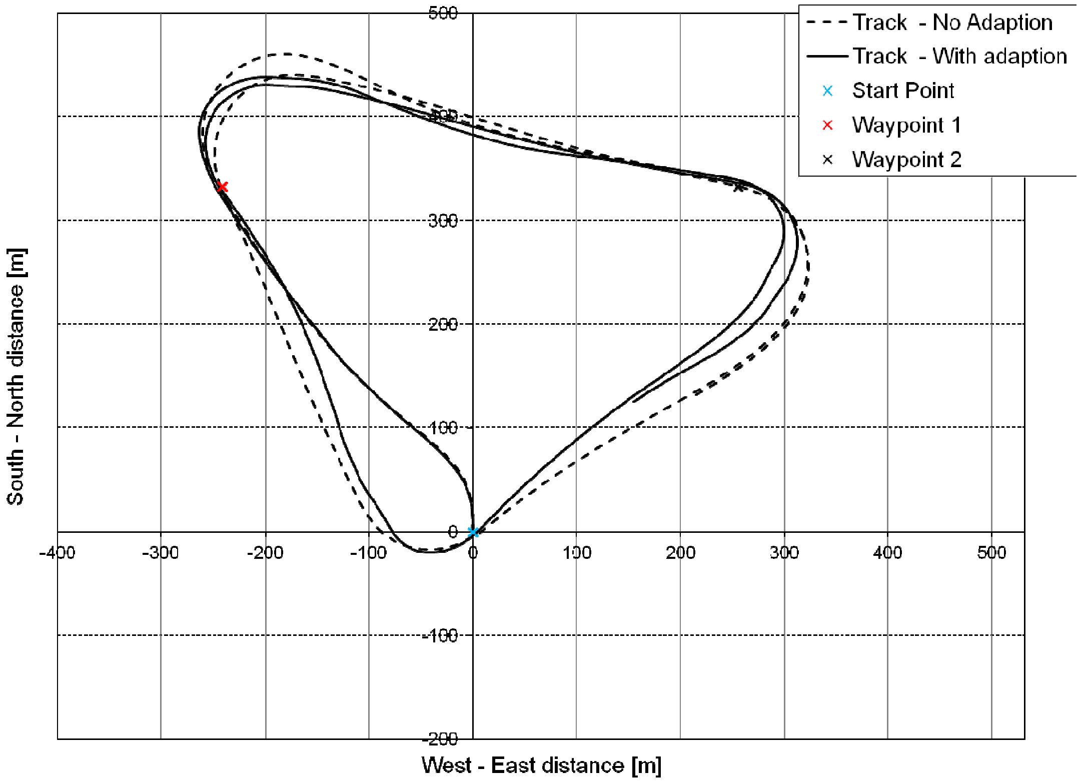
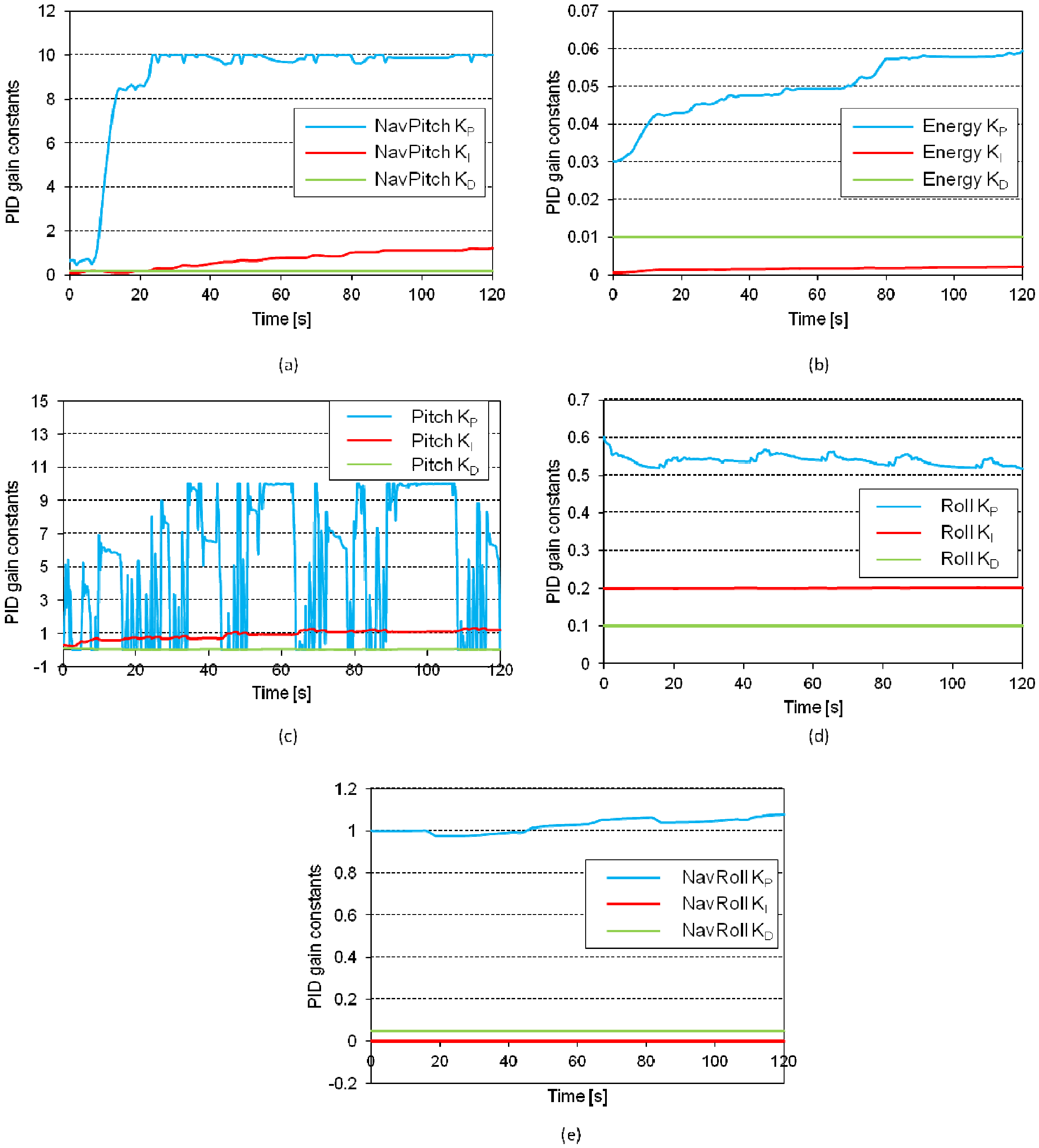


Figure 5.114: Simulated track followed by the aircraft in the fully autonomous flight mode with a high pitch adaption rate and a low roll adaption rate (0° deflection failure of the right elevator)

An examination of Figure 5.115 shows that, in Figure 5.115 (a), the airspeed proportional gain reached its maximum allowable value of 10 very rapidly and maintained approximately this high value throughout the simulation. There was also an increase in integral gain. There was, however, a large and highly unstable oscillatory variation in pitch proportional gain, as seen in Figure 5.115 (c). These results are highly undesirable as they show that the adaption algorithm is not tolerant to this failure in autonomous mode. Based on Figure 5.115 (a), it was decided to decrease the airspeed adaption rate by a factor of 10 to determine if this might improve the performance of the system, the results of which are shown in Figure 5.116 to Figure 5.118.



(a)

(b)

(c)

(d)

(e)

Figure 5.115: Simulated time history of the airspeed PID (a), energy PID (b), pitch PID (c), roll PID (d) and navigational roll PID (e) gain changes with the adaption mechanism enabled for fully autonomous flight

Figure 5.116 shows that the airspeed tracking has improved for the case with adaption and now demonstrates that the adaption algorithm improves the control performance of the system. The overshoot is generally smaller than the case without adaption and the time to reach the set point value is lower. This is also seen in Figure 5.117 which shows far better altitude tracking when compared to the non-adapted case. Thus, again, it is noted that the adaption rates have a large impact on the performance of the system and the ability of the adaption algorithms to cope with a failure.

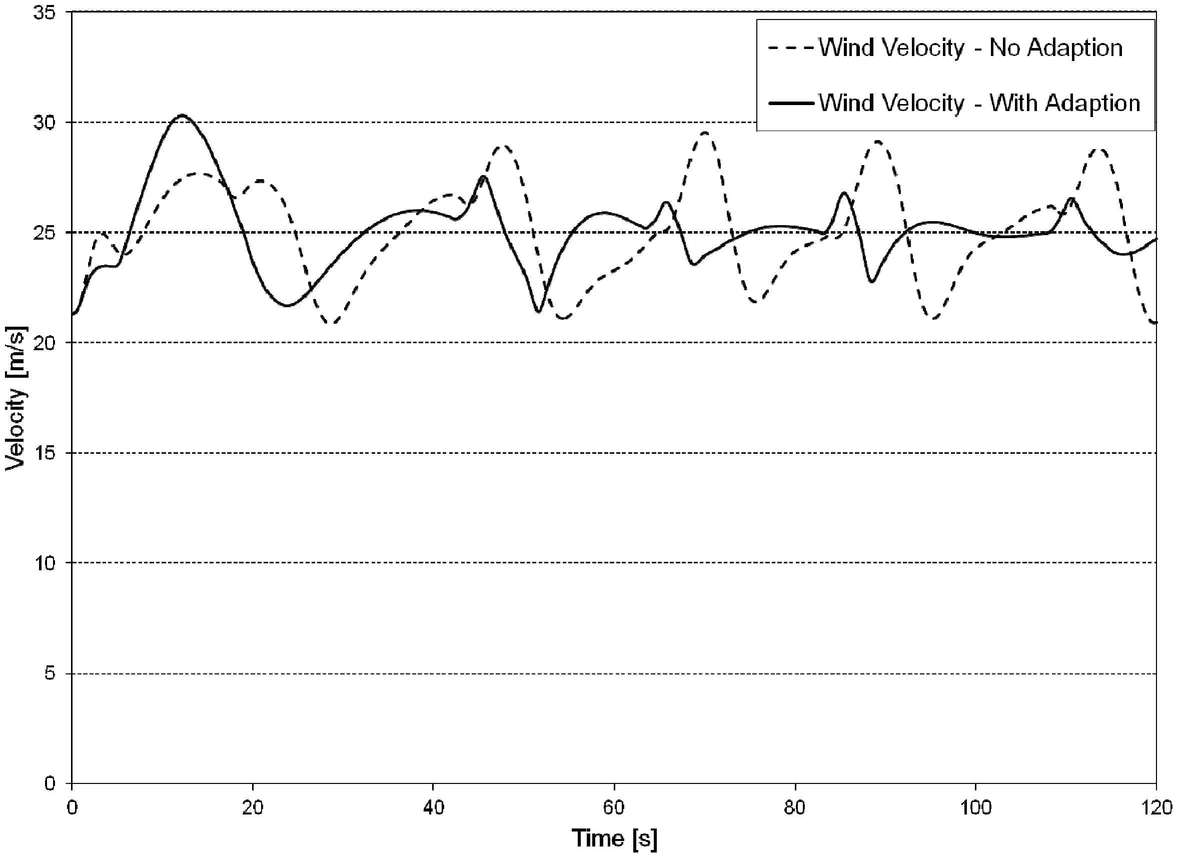


Figure 5.116: Simulated time history of airspeed of the aircraft in the fully autonomous flight mode with a low airspeed adaption rate and a low roll adaption rate (0° deflection failure of the right elevator)

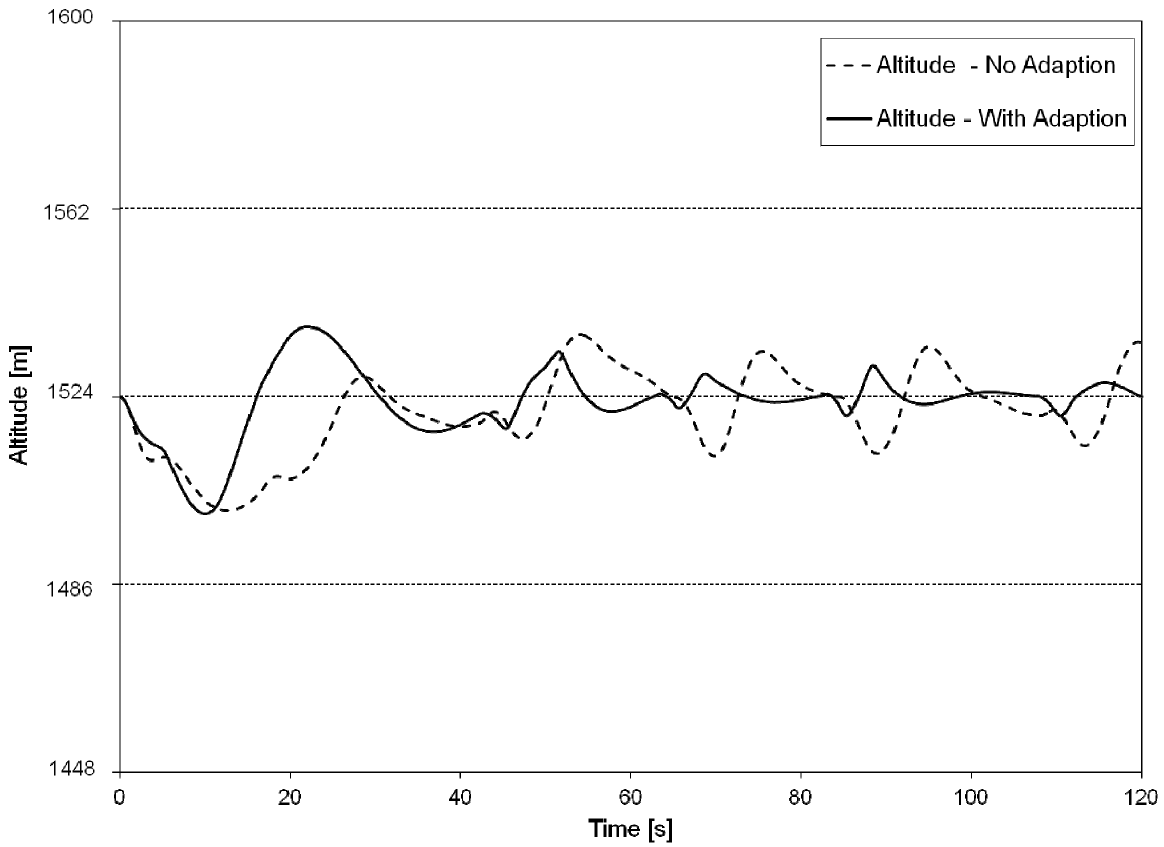
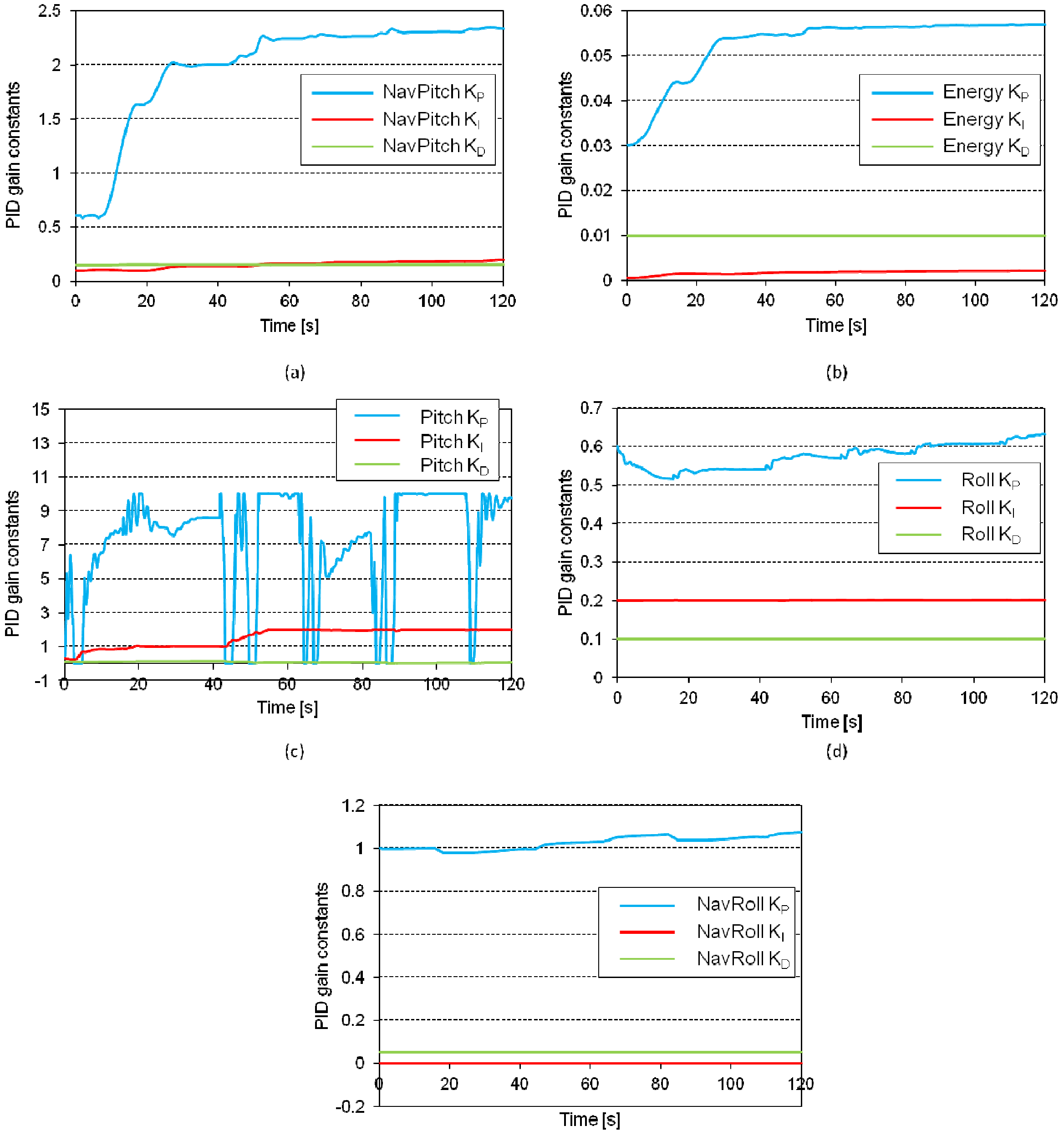


Figure 5.117: Simulated time history of altitude of the aircraft in the fully autonomous flight mode with a low airspeed adaption rate and a low roll adaption rate (0° deflection failure of the right elevator)

Figure 5.118 shows the response of the PID gains. It is observed that the large proportional gain seen in Figure 5.115 (a) is no longer present in Figure 5.118 (a). It is also noted that Figure 5.118 (c) has far fewer oscillations and is generally more stable; however, large oscillations and extreme values are still present. The PID gains for the remaining controllers are reasonably stable and well behaved.

By reducing the adaption rate of the airspeed PID, stability to the system was restored but this will compromise the control case. The controlled case was run and checked and it was found that the performance seen in Figure 5.106 and Figure 5.107 was degraded somewhat. The performance was still satisfactory and the results were far better than the non-adapted case. As a result the reduced adaption rate was maintained throughout the remaining simulations.



(a)

(b)

(c)

(d)

(e)

Figure 5.118: Simulated time history of the airspeed PID (a), energy PID (b), pitch PID (c), roll PID (d) and navigational roll PID (e) gain changes with the adaption mechanism enabled for fully autonomous flight mode with a low airspeed adaption rate and a low roll adaption rate (0° deflection failure of the right elevator)

Right Elevator Failure - 50% Hard Over (Test 2)

The simulation was run with the introduction of a 50% hard over failure of the right elevator causing a nose-up pitching moment and a rolling moment to the right. A similar result to the 0° deflection case was observed. It was found that, with the adaption enabled, there was a general improvement in control of airspeed and altitude with the overshoot following a steady decay from manoeuvre to manoeuvre, as shown in Figure 5.119 and Figure 5.120. The case without adaption shows a constant pattern of overshoot and oscillations that is in general larger than the case with adaption. Small changes were noted in the track followed by the aircraft in that the adaption algorithm seemed to induce a slight overshoot in heading although the turn radius in each turn was generally smaller as shown in Figure 5.121 .

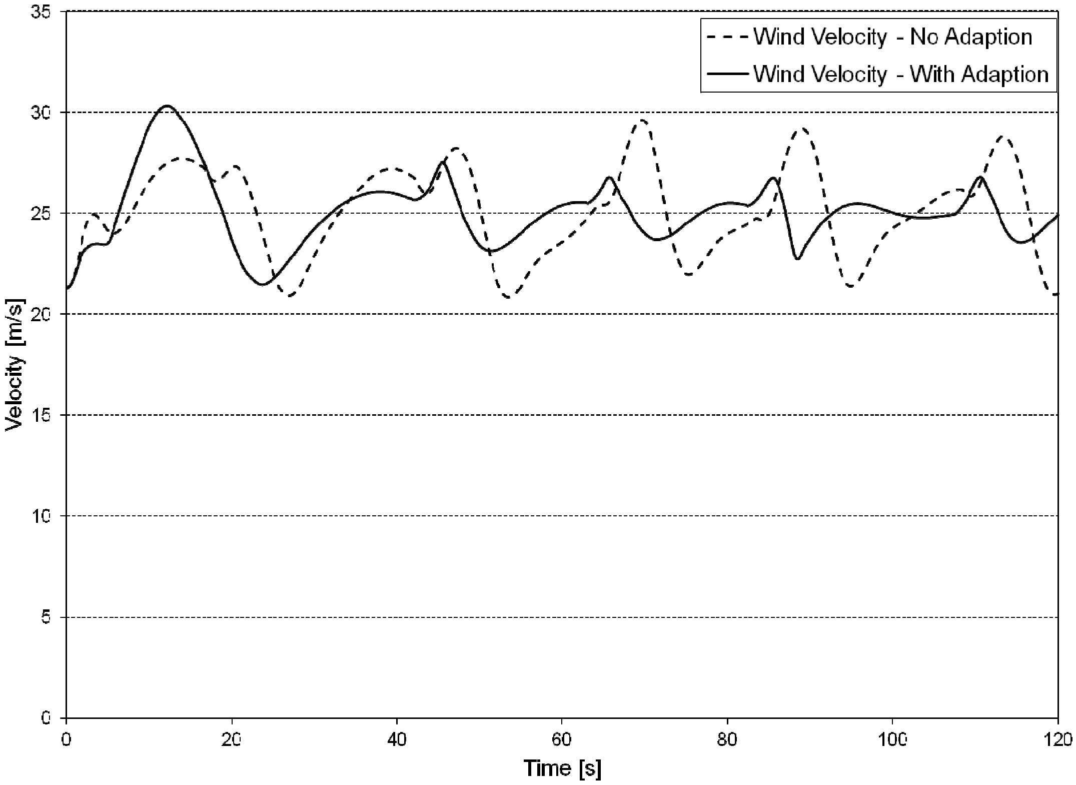


Figure 5.119: Simulated time history of altitude of the aircraft in the fully autonomous flight mode with a low airspeed adaption rate and a low roll adaption rate (50% deflection failure of the right elevator)

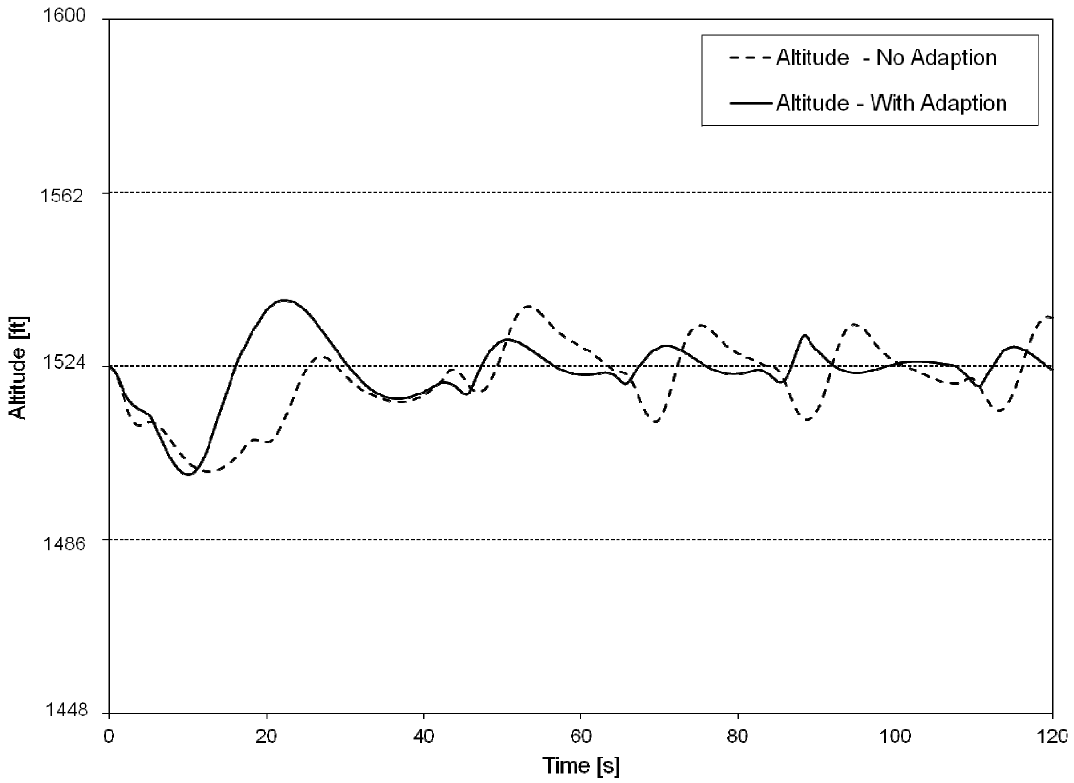


Figure 5.120: Simulated time history of altitude of the aircraft in the fully autonomous flight mode with a low airspeed adaption rate and a low roll adaption rate (0° deflection failure of the right elevator)

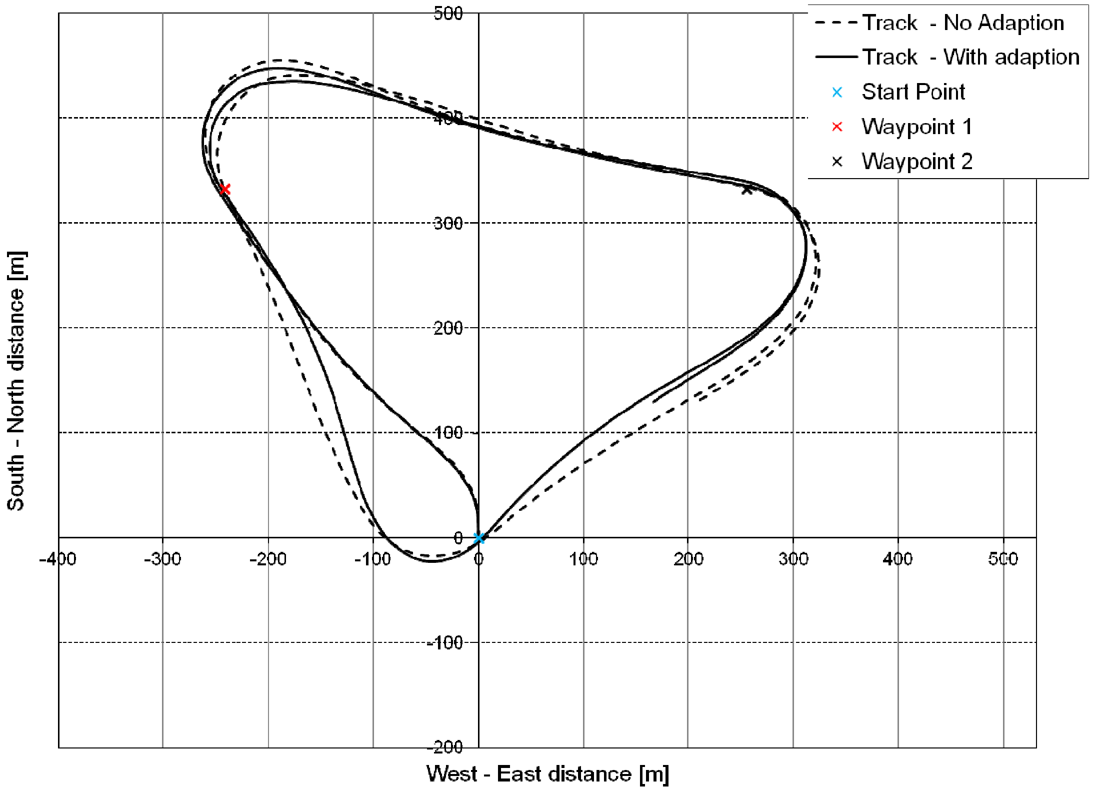


Figure 5.121: Simulated track followed by the aircraft in the fully autonomous flight mode with a low airspeed adaption rate and a low roll adaption rate (50% deflection failure of the right elevator)

Right Elevator Failure - 100% Hard Over (Test 3)

The simulation was run with and without the adaption algorithm enabled and with the introduction of a 100% hard over failure of the right elevator, this causing a nose-up pitching moment and a rolling moment to the right. Figure 5.122 (a) shows that, while the adaption algorithm was unable to return the airspeed to the set point value, the overshoot was typically lower than the case without adaption. Figure 5.122 (b) shows a similar trend with the altitude control. The altitude control was, however; generally better in achieving a more consistent altitude. The track results have changed only slightly, with slightly closer correlation between adapted and non-adapted cases.

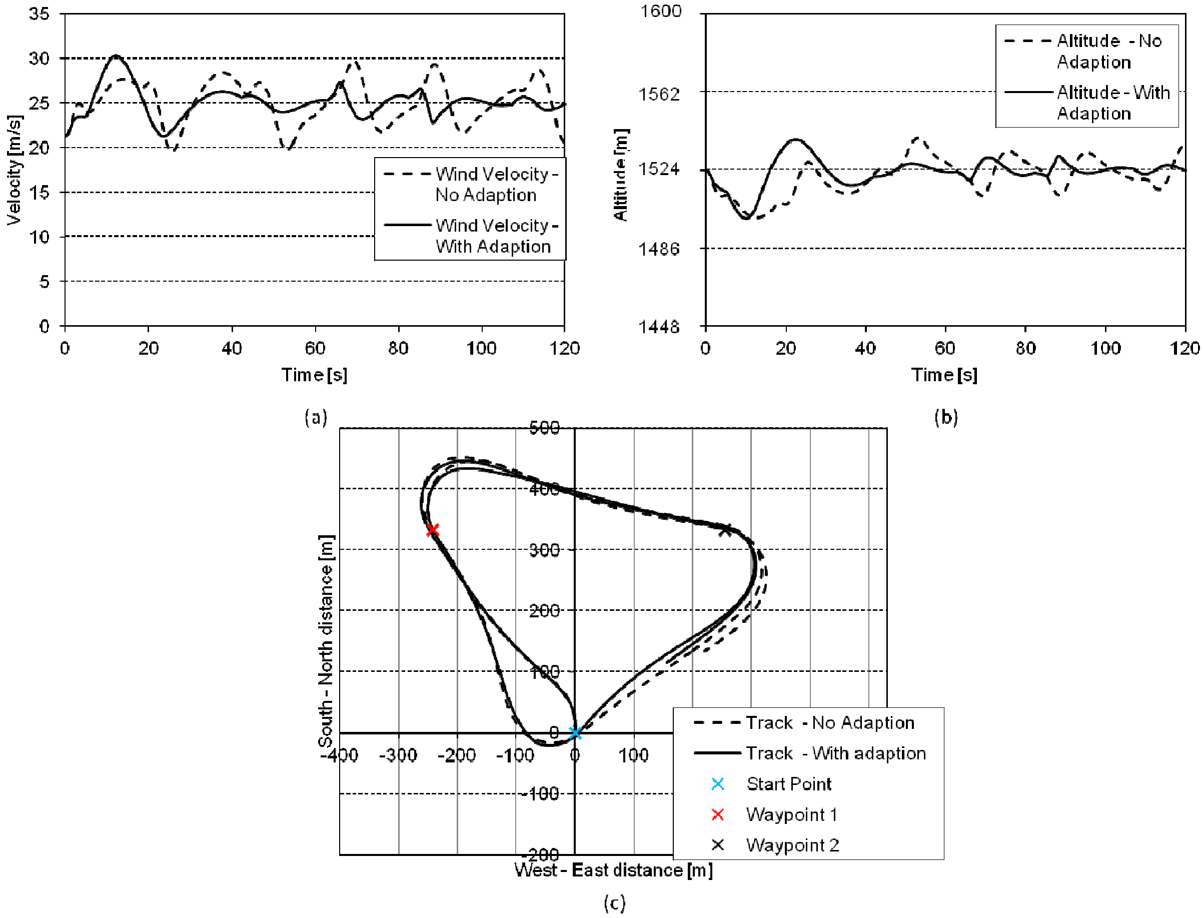


Figure 5.122: Simulated time history of airspeed (a), altitude (b) and the ground track (c) of the aircraft in the fully autonomous flight mode with a low airspeed adaption rate and a low roll adaption rate (100% deflection failure of the right elevator)

Right Aileron Failure - 0° Deflection (Test 4)

The simulation was run inducing a 0° deflection failure of the right aileron. This failure showed little effect on the performance of the aircraft to maintain the waypoint following and made no significant difference to the airspeed and altitude tracking, as shown in Figure 5.123, which compares well with the revised control results. This is unsurprising as any rolling moment induced by the neutral aileron position would be small and because the control allocation algorithm is enabled, any additional control effort is easily attained for this flight case (Figure 5.124).

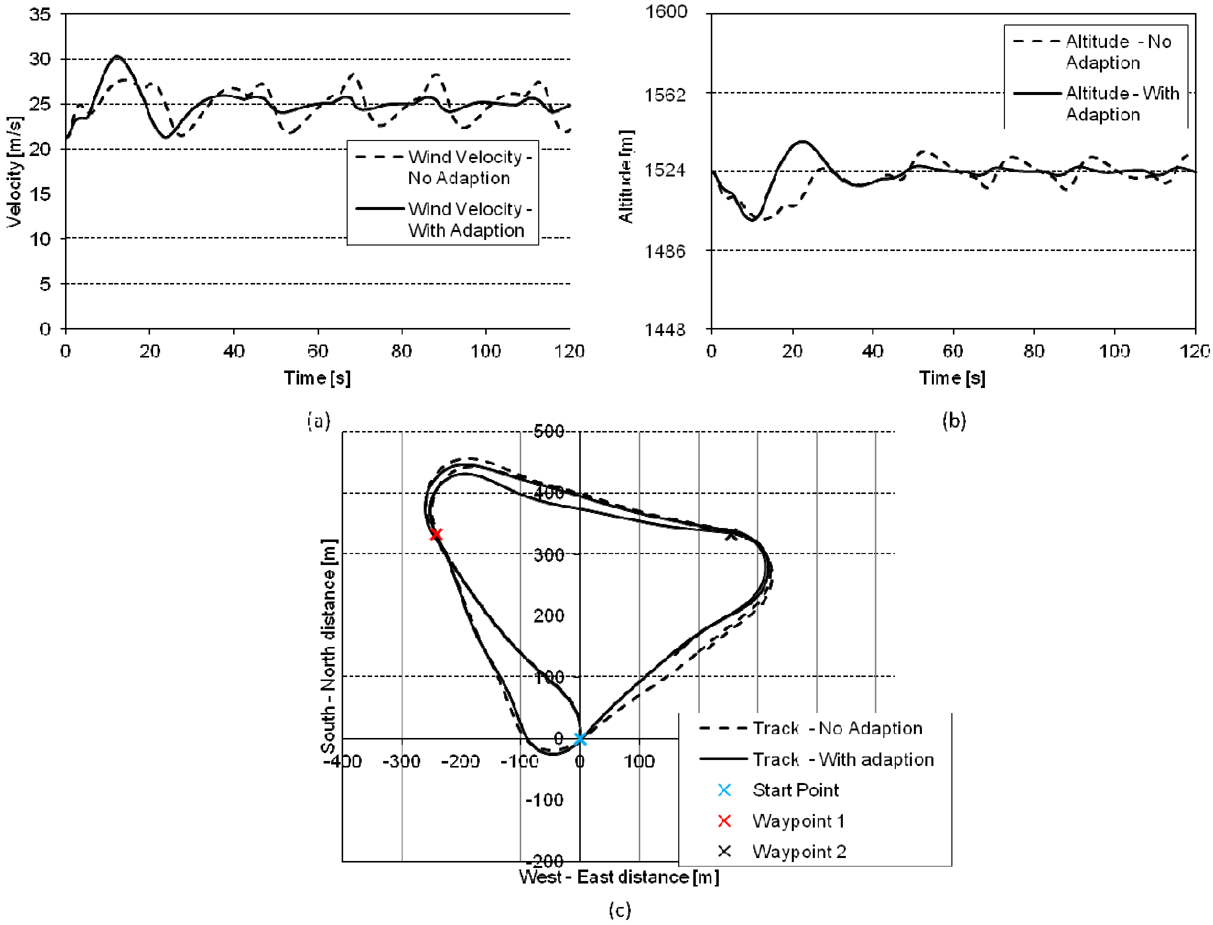


Figure 5.123: Simulated time history of airspeed (a), altitude (b) and the ground track (c) of the aircraft in the fully autonomous flight mode with a low airspeed adaption rate and a low roll adaption rate (0° deflection failure of the right aileron)

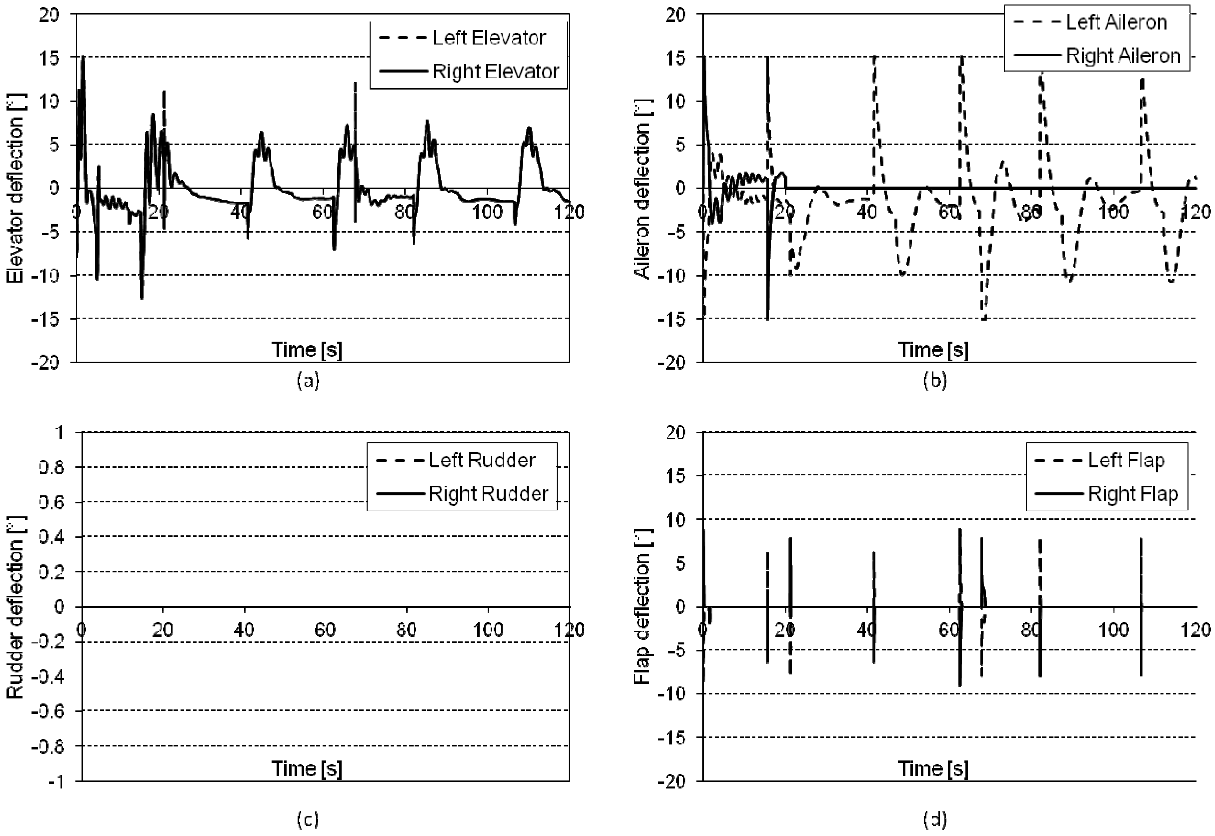


Figure 5.124: Simulated time history of elevator (a), aileron (b), rudder (c) and flap (d) deflections for the aircraft in the fully autonomous flight mode with a low airspeed adaption rate and a low roll adaption rate (0° deflection failure of the right aileron)

Right Aileron Failure - 50% Hard Over (Test 5)

The simulation was run with the introduction of a 50% hard over failure of the right aileron, inducing a rolling moment to the right. Again, similarly to the 0° deflection case, little variation was observed for this failure. This is promising in that the results exhibit no significant difference when compared to the revised control. There is, however, a control performance gain when the adaption algorithm is enabled with the bulk of the fault tolerance being provided by the control allocation algorithm. Again Figure 5.126 shows the importance of the allocation algorithm as the flaps are deflected to ensure that the required control authority is met.

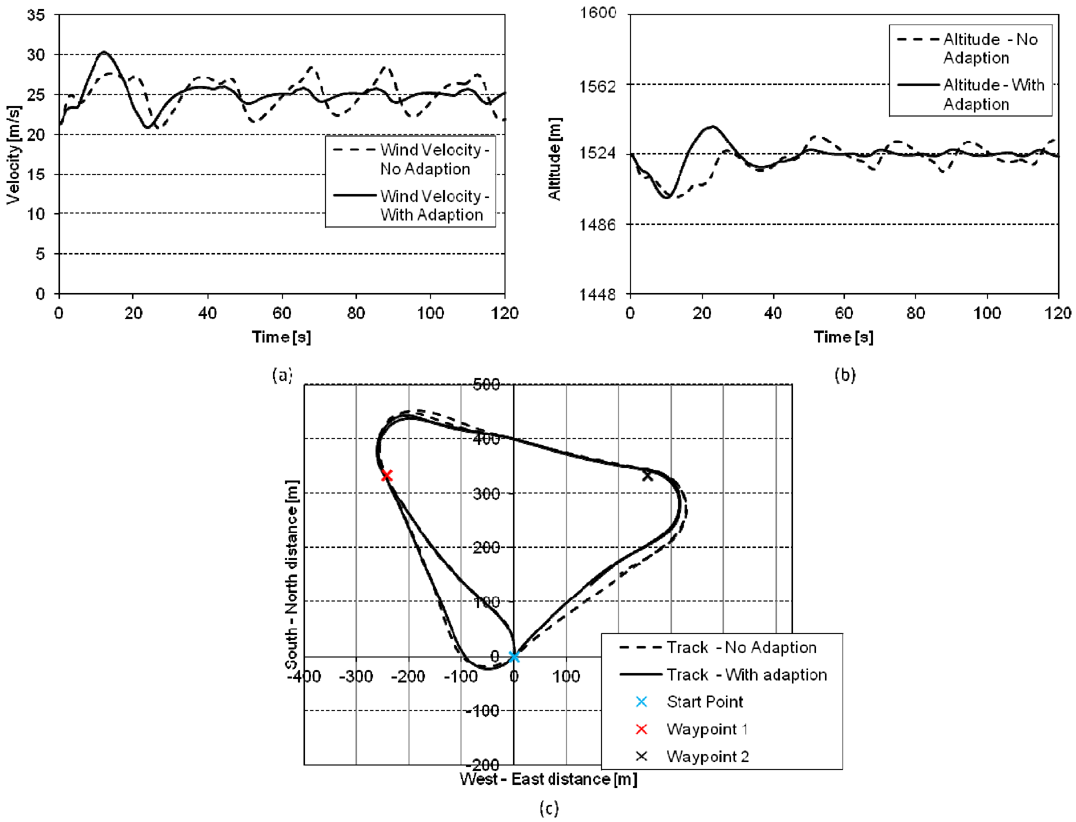


Figure 5.125: Simulated time history of airspeed (a), altitude (b) and the ground track (c) of the aircraft in the fully autonomous flight mode with a low airspeed adaption rate and a low roll adaption rate (50% deflection failure of the right aileron)

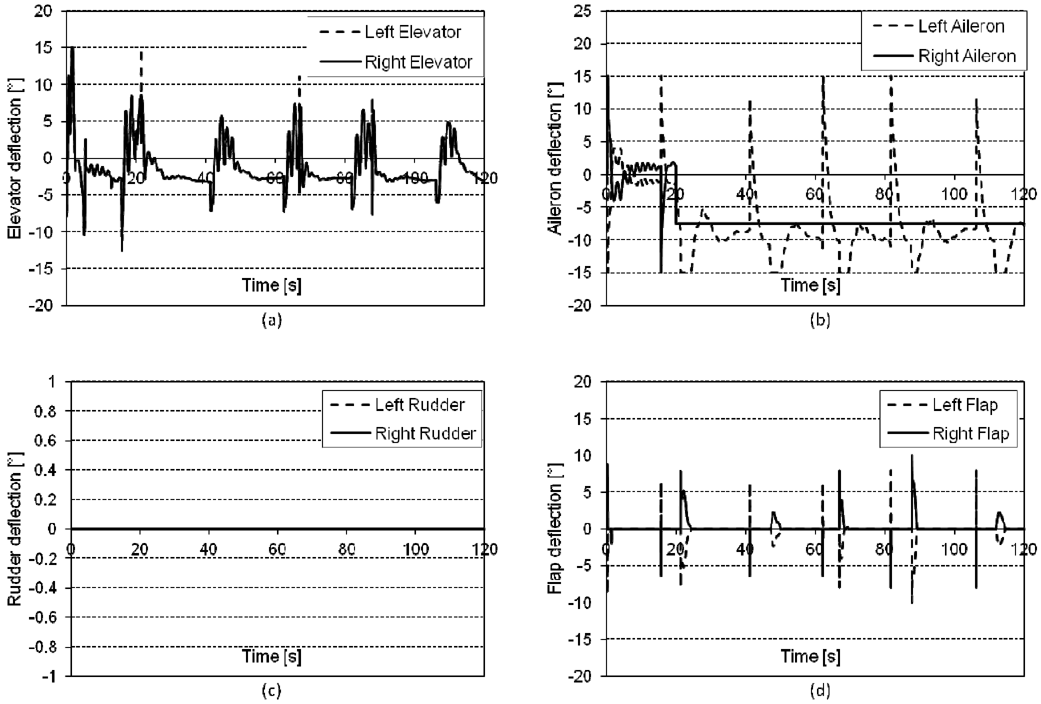


Figure 5.126: Simulated time history of elevator (a), aileron (b), rudder (c) and flap (d) deflections for the aircraft in the fully autonomous flight mode with a low airspeed adaption rate and a low roll adaption rate (50% deflection failure of the right aileron)

Right Aileron Failure - 100% Hard Over (Test 6)

The simulation was again run this time inducing a 100% hard over failure of the right aileron. Again, the proposed system was highly tolerant to this failure with little change in the airspeed, altitude traces and little variation in the track followed as shown in Figure 5.127. The remaining results show little difference from those observed in the previous cases. Should they be desired, the remaining results can be found in Appendix C.

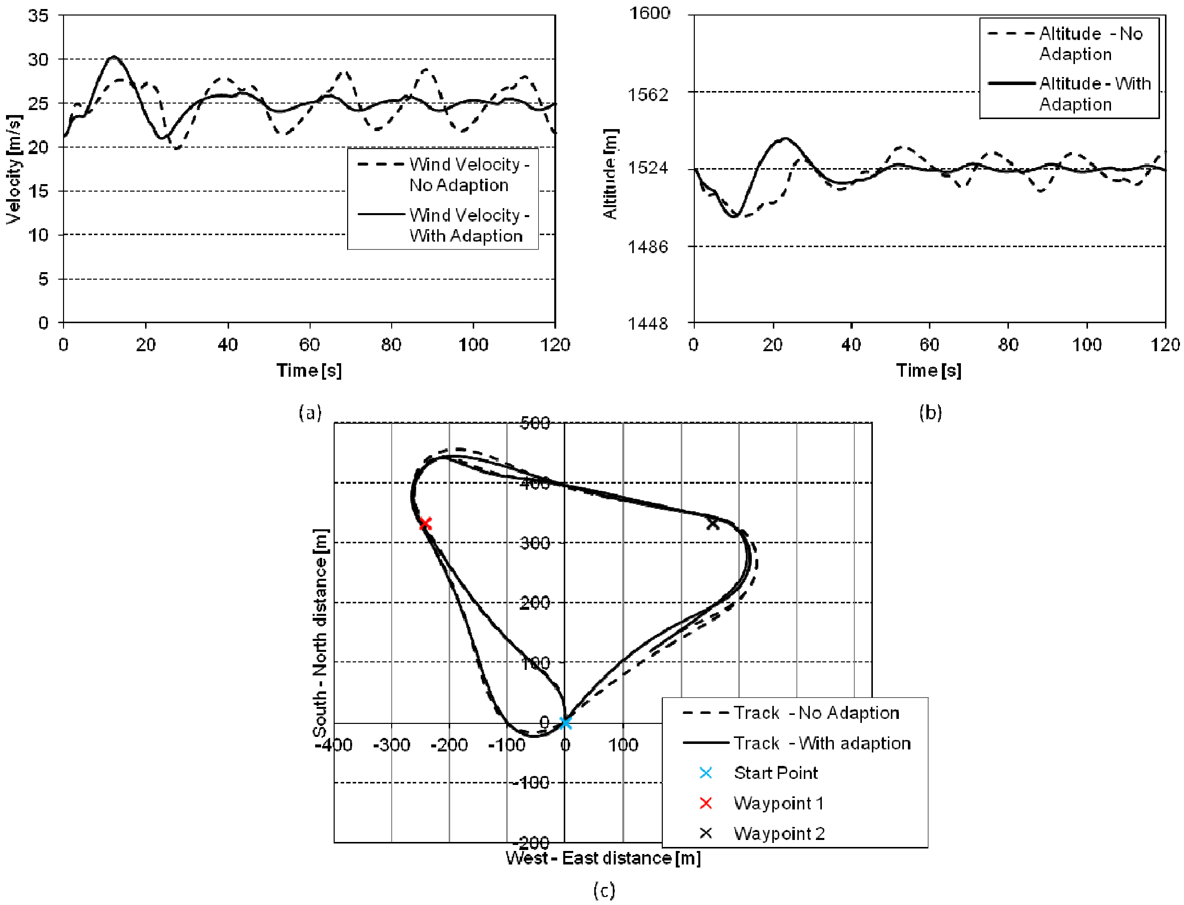


Figure 5.127: Simulated time history of airspeed (a), altitude (b) and the ground track (c) of the aircraft in the fully autonomous flight mode with a low airspeed adaption rate and a low roll adaption rate (100% deflection failure of the right aileron)

Comparison between Unmodified Autopilot and Autopilot with Proposed Control Algorithms Enabled (Test 7)

The simulation was run inducing a combined failure of two control surfaces. A 50% hard over failure was applied to the right elevator and a 100% hard over failure was applied to the right aileron. In this test, a comparison between the unmodified autopilot (without the adaption and allocation algorithms) and the proposed control strategy are demonstrated for the most complex flight mode. The results, presented in Figure 5.128 to Figure 5.133, show that, for this failure, the unmodified autopilot is unable to tolerate the failure. There is a rapid increase in airspeed, Figure 5.128, a rapid loss in altitude, Figure 5.129, and a spiral dive is established, Figure 5.130, which would result in the loss of the aircraft. In stark contrast, the proposed fault-tolerant control strategy shows the successful tolerance of this control failure.

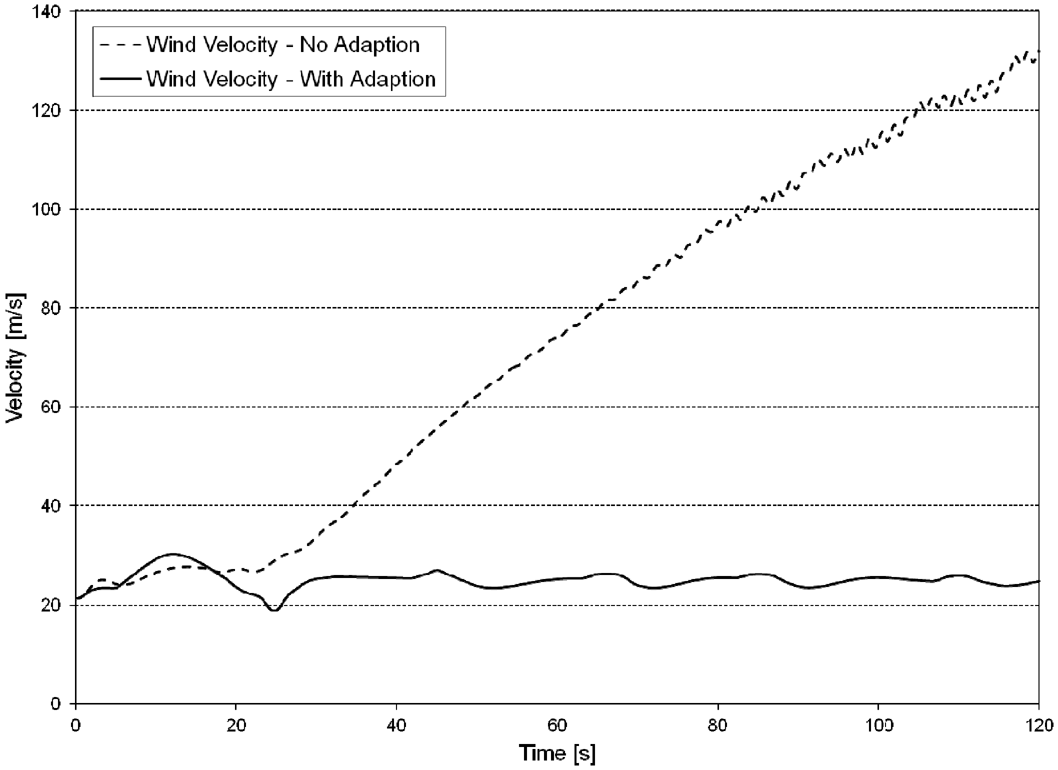


Figure 5.128: Simulated time history of airspeed of the aircraft in the fully autonomous flight showing the difference between the un-modified autopilot and the proposed control strategy with a combined failure of the right elevator and aileron

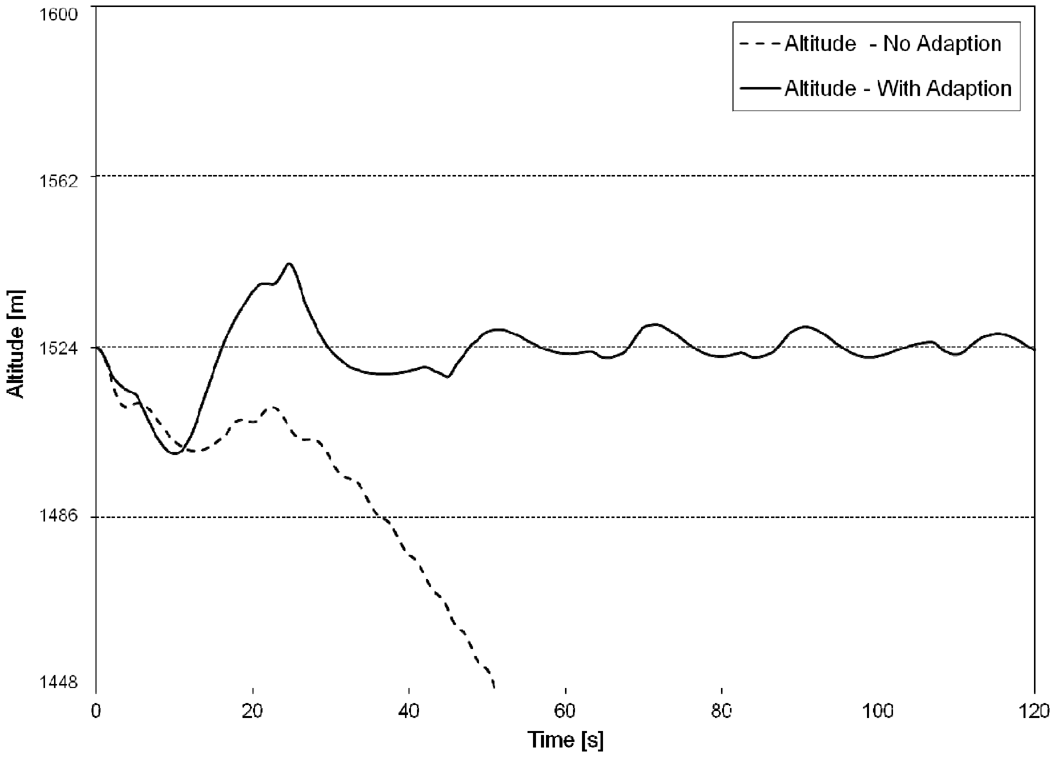


Figure 5.129: Simulated time history of the altitude of the aircraft in the fully autonomous flight showing the difference between the un-modified autopilot and the proposed control strategy with a combined failure of the right elevator and aileron

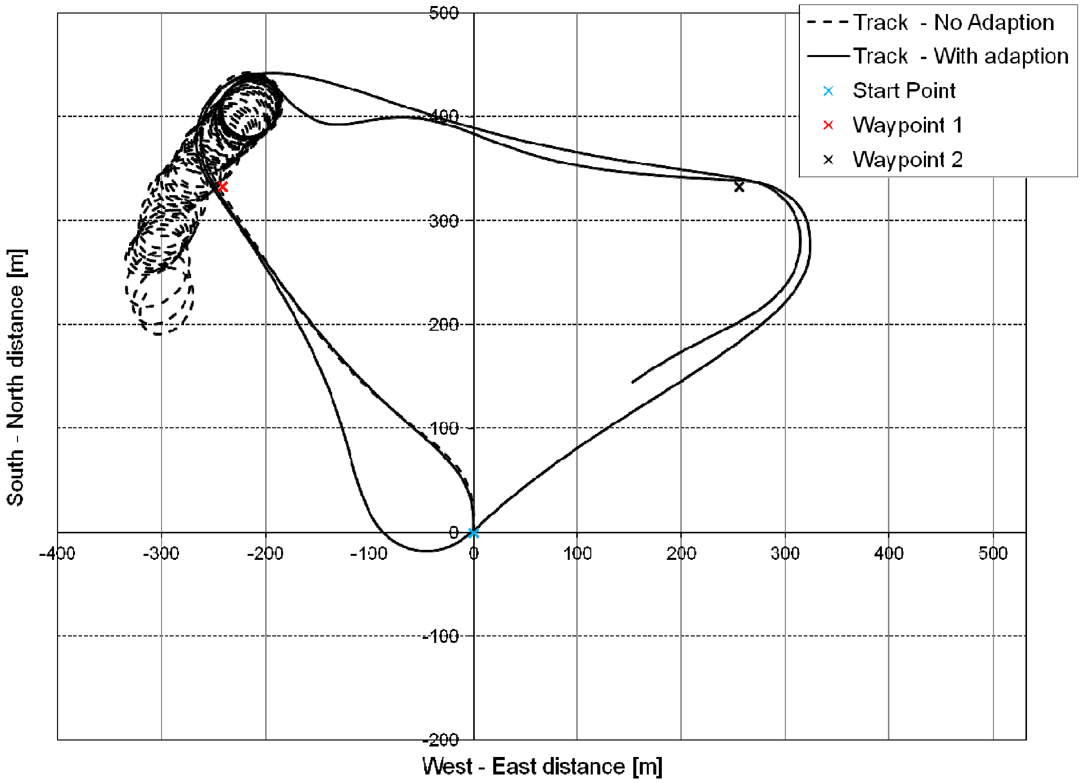


Figure 5.130: Simulated time history ground track of the aircraft in the fully autonomous flight showing the difference between the un-modified autopilot and the proposed control strategy with a combined failure of the right elevator and aileron

The pitch and roll angles, shown in Figure 5.131 , of the adapted case are also controlled and within acceptable limits unlike the non-adapted case. In the non adapted case, the roll angle diverges and undergoes a series of violent oscillations. The pitch angle also shows this oscillatory tendency. The allocation and adaption systems work together, seen in Figure 5.132 (a) to (d), which shows that the flaps ailerons and elevators are being actively used to control pitch and roll angles.

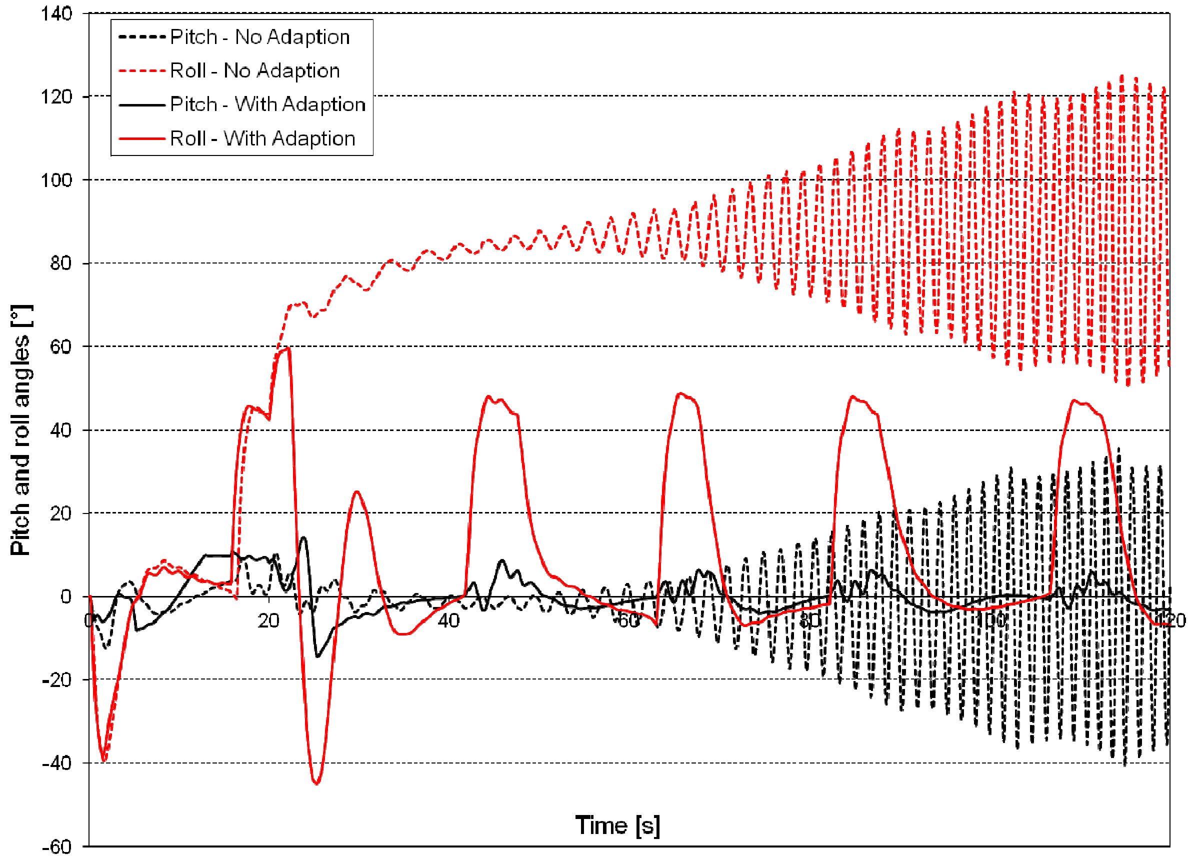


Figure 5.131: Simulated time history of pitch and bank angles for the aircraft in the fully autonomous flight showing the difference between the un-modified autopilot and the proposed control strategy with a combined failure of the right elevator and aileron

Of great importance is the stability of the PID gains shown in Figure 5.133 which indicates that this flight pattern could be repeated indefinitely as there is no significant build up in any controller gains. This test clearly shows the difference between the proposed fault-tolerant control system and the unmodified control system and shows that the proposed system is able to tolerate significant control actuation failures and maintain the safety of the aircraft.

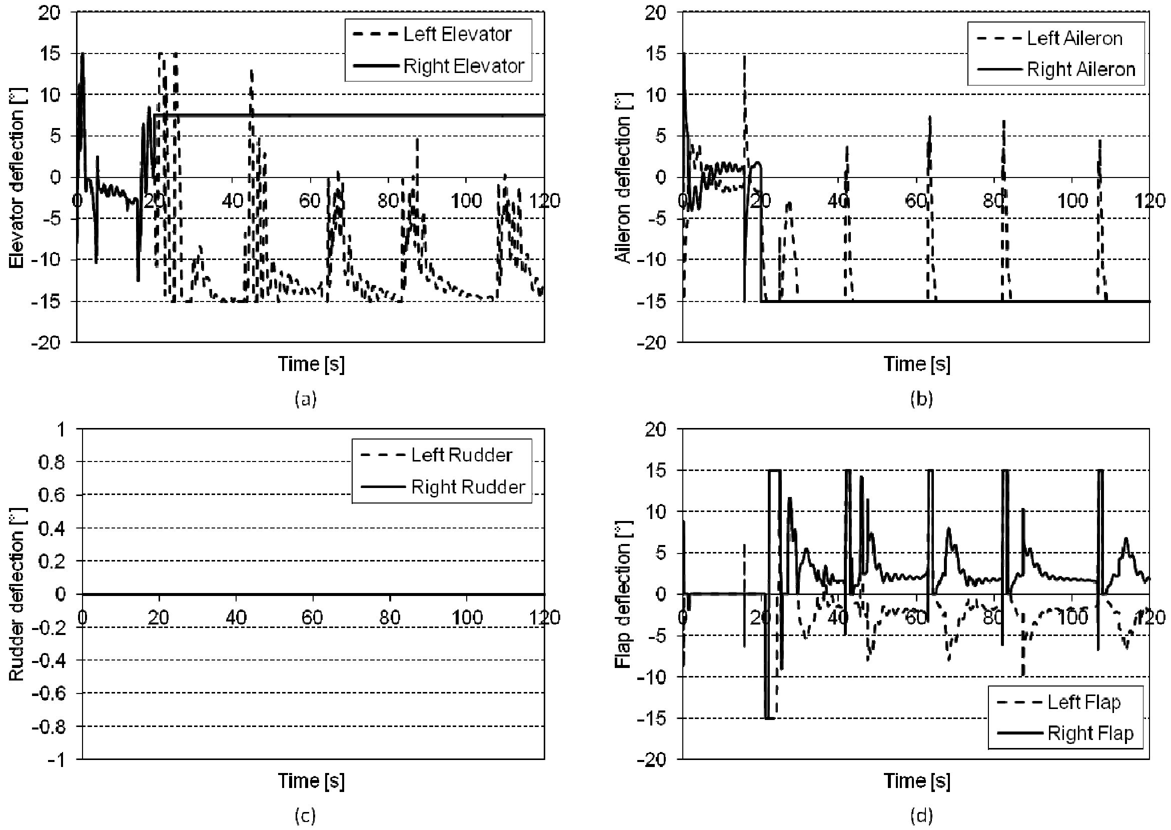
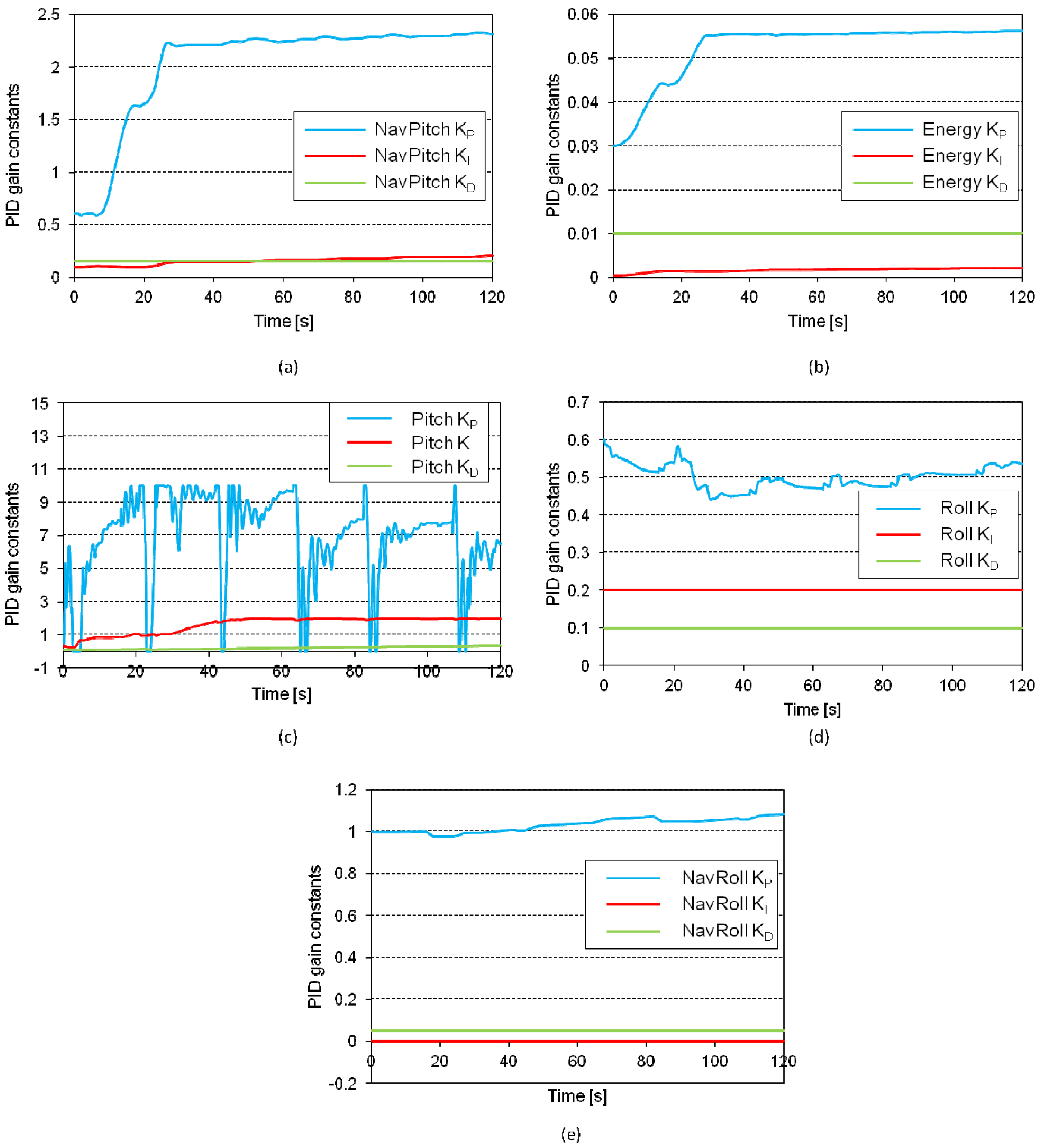


Figure 5.132: Simulated time history of elevator (a), aileron (b), rudder (c) and flap (d) deflections for the aircraft in the fully autonomous flight showing the difference between the un-modified autopilot and the proposed control strategy with a combined failure of the right elevator and aileron



(a)

(b)

(c)

(d)

(e)

Figure 5.133: Simulated time history of the airspeed PID (a), energy PID (b), pitch PID (c), roll PID (d) and navigational roll PID (e) gain changes with the adaption mechanism enabled for fully autonomous flight mode

6 Chapter 6 – Discussion of Results

6.1 Chapter Outline

This chapter presents a review of the objectives of the research as well as the results of Chapter 5. The success or failure of the proposed fault-tolerant control strategy is discussed and conclusions as to the suitability of the proposed fault-tolerant control system are drawn. Finally the recommendations for future work on this topic are presented.

6.2 Research Objectives

The primary objective of this research, as outlined in Chapter 1.7, is to research, design and simulate a fault-tolerant control system into a commercial autopilot system, while making use of artificial intelligence techniques. This objective was expanded into more detailed objectives which are as follows.

- Actuator failure
- The loss of a control surface
- Uncertainties within the aircraft model

The simulated tests concentrated on a number of different actuator failures. These included a 0° deflection failure, a 50% hardover failure, and a 100% hardover failure. The right elevator and right aileron were failed individually. Simulation where both the aileron and elevator were failed simultaneously were also conducted.

6.3 Inner Loop Simulations

The inner loop simulations provided valuable insight into the feasibility of the adaption algorithm in providing a measure of fault tolerance. The failures were tested for two different commanded flight manoeuvres, namely, straight and level flight and a constant bank turn. The adaption algorithm was tested against the non-adapted autopilot in each test and, in the cases where there was insufficient control authority available, the control allocation algorithm was introduced.

6.3.1 Elevator Failures

The first failure tested was a 0° deflection failure which generated a nose-up pitching moment coupled with a rolling moment to the right. This failure showed that the autopilot, in its current form and without adaption, is tolerant to this type of failure. It was found, however, that there was an improvement in the control performance of the system when the adaption algorithm was enabled.

The 50% deflection failure showed a marked increase in pitch and roll error although it was demonstrated that the unmodified autopilot was again able to tolerate this failure. It was noted that, similarly to the 0° deflection failure test, significant improvement in the control performance was attained with the adaption algorithm enabled. A significant improvement in control performance was seen in the pitch response while only minor improvements were seen in the roll response. Little difference was observed between the two flight manoeuvres tested.

The 100% elevator failure, which was expected to induce difficulties for the autopilot, was reasonably well controlled. However, the autopilot with and without adaption was unable to maintain the desired set point. This was expected as insufficient control authority was available. As soon as the elevator control saturated, the aircraft was subject to its stick-fixed dynamic behaviour which was shown by a damped phugoid motion. The roll error was again easily tolerated by the autopilot and improvement in control performance was demonstrated with the adaption algorithm. A subtle difference was observed between flight manoeuvres with marginally better tolerance to the failure being observed for the constant banked turn. This was due to the direction of the failure and the need for a higher pitch angle to maintain altitude in the turn. The difference between the required control authority and the available control authority was lower resulting in a smaller oscillation about the set point value.

Of note was the rapid increase in proportional gain for the 100% elevator failure case as the adaption algorithm attempted to regain control. This presents a potential problem with the adaption algorithm logic. Should a control surface reach its physical limits and saturate, the adaption algorithm has no mechanism to detect or resolve the resulting lack of control authority.   This was an unintended consequence that developed which has the potential to cause a catastrophic failure of the system. This is highly undesirable as the purpose of the adaption system is to improve fault tolerance and not to induce further difficulties. This result highlights the need for the development of a simple control allocation technique to be used in conjunction with the adaption algorithm.

The combination of control adaption and allocation was tested for the 100% elevator failure.   The control allocation algorithm was highly successful in tolerating this failure to the extent that the non-adapted autopilot was tolerant to the 100% hard over failure. It was found, however, that the adaption algorithm improves the control performance when compared to the non-adapted autopilot. This is largely due to the adaption of the PID gains to accommodate the change in control characteristics. This simulation showed the importance of the control allocation algorithm and suggests that this algorithm is the primary driver for the successful result obtained in this simulation. It is, however, of great advantage to include the adaption algorithm: the combination of the two algorithms yielded far improved control performance.

The simulations showed that, for the inner control loop, the proposed control methodology of combining the control adaption with the control allocation yielded a successful tolerance to all of the tested elevator failures.

6.3.2 Aileron Failures

The first aileron failure to be tested was the 0° deflection failure which yielded only a small roll disturbance. Because the disturbance was so small only a small opportunity existed for the adaption algorithm to make changes to the PID gains. As a result, only a small difference was noted in bank angle control performance for both level flight and the constant banked turn.

The more aggressive 50% hard over failure simulation yielded somewhat more significant results. There was a large roll disturbance that was adequately controlled by the non-adapted autopilot; however, the maximum roll error was reduced by half when the adaption algorithm was enabled. This test yielded a promising result although two areas of concern were noted, namely, the introduction of a more oscillatory behaviour and a delay in returning the bank angle to the set point value. It is noted that, although the adaption algorithm did increase the time to reach the set point value, the bank angle error was low and this delay would not have impacted significantly on the performance of the aircraft. The cause of the delay is speculated to lie in a number of potential areas. One may be a flaw within the fuzzy logic rule base; or another may be in the chosen behaviour of the ideal model. A review of these issues should be conducted.

The 100% hardover failure of the right aileron yielded an interesting set of results. It was thought that this failure, similar to elevator 100% hardover failure case, would result in the failure of the autopilot to maintain the set bank angle; however, this was not the case. The autopilot, without adaption, was able to level the wings of the aircraft with the aid of the rolling moment supplied by the rudder. This rudder input was generated by a feedforward loop commonly used on remotely piloted aircraft. The rudder-aileron mix was sufficient to cause the levelling of the wings, albeit slowly, despite the lack of available roll control from the ailerons. The same success could not be shown of the autopilot with adaption. The autopilot supplies a commanded aileron output based on the roll error and has no mechanism to detect when a control surface saturates. It then simply commands a higher control deflection after a control has saturated. Under normal circumstances, this does not cause any significant problems as the control reaches its mechanical limits and cannot be deflected further. In this case two problems arise. The first is due to the methodology that determines the rudder output based on the aileron output. With the increase in aileron output beyond control limits, there is a corresponding increase in rudder output that is not limited until full rudder deflection is achieved. While not a severe problem alone, when combined with the rapid rise in the proportional gain of the roll PID controller, a dynamically unstable rudder control response results. This causes a series of severe and increasing oscillations which results in excessive sideslip angles. During the simulation the aircraft achieved a sideslip angle of 30° at which point the simulation was stopped automatically. The mechanism which resulted in the successful control of the aircraft in the non- adapted case was the direct cause of the failure of the adapted case.

The problems encountered with the adaption algorithm were solved by removing the aileron to rudder mixing. However, while the system did not become unstable, as it did previously, it was not able to maintain the desired set point with the system achieving maximum bank angle error in excess of 70°. This occurred for the unmodified autopilot as well as the adapted autopilot and is highly undesirable. Similarly to the 100% hardover elevator failure, the introduction of the control allocation algorithm yielded a satisfactory result in that the set point was achieved both with and without adaption. As was found in the other simulations, in general, the adaption algorithm improved control performance. However, similarly to the 50% aileron failure case, the time to reach the roll set point value was longer than the non-adapted case, even though the maximum bank angle error was significantly reduced.

The exact cause of this has not yet been established as it was expected that a similar performance to that attained in the pitch direction would be achieved; however, it is thought that a modification to the fuzzy rule set or an adjustment to the ideal model behaviour may yield better results. Research into the reasons for this reduced performance is recommended in future work.

6.3.3 Combined Failures

A combined failure of the aileron and elevator was performed to test the ability of the proposed control laws to tolerate multiple surface failures. The first combination of failures was a 0° deflection failure of the right elevator and right aileron. This failure caused a larger pitch and roll error, as would be expected, than the single control surface cases. The test showed that the autopilot, even without adaption, was able to tolerate this failure; however, as in the previous cases, the system with adaption was found to have improved control performance in pitch and roll. The maximum errors in both pitch and roll directions were reduced by a small amount and the time to re-establish the set point value was reduced.

The 50% hardover failure of the elevator and aileron was also tested. This, again, resulted in a combined pitch and roll error that was the sum of each individual control failure. The autopilot without adaption was tolerant to this failure and restored the aircraft to the required set points in both pitch and roll. It was noted that the pitch control performance was greatly improved with the adaption algorithm with a reduced maximum error and a faster return to the set point value. In roll, the maximum error was greatly reduced but, similarly to the aileron failure cases, the time to reach the set point value was significantly longer.

Lastly, the simulation was run with a 100% hardover failure of both the right elevator and right aileron. This resulted in a complete failure in the autopilot, with or without adaption, to tolerate this failure. This is as expected as the control allocation algorithm was disabled. The bank angle, similarly to the previous test, increased to the point where the wings were nearly vertical. Slight improvements were seen in control performance with the adaption enabled but this improvement is meaningless in this context as the aircraft would have entered a spiral dive. The control allocation algorithm was then enabled and the simulation re-run. This resulted in the successful return of the aircraft to the desired set point for both banked flight and level flight. The allocation algorithm did, however, reach the limits of its ability to control this failure and a number of oscillations in pitch occurred. The flaps, being the only remaining control surface to provide both pitch and roll control, was unable to provide enough control authority and reached their limits of travel. It is highly unlikely that this flight would have been successful as the flaps were unable to provide adequate pitch control and the phugoid motion of the aircraft was evident both with and without the adaption algorithm. The rudder mix was removed to reduce the amount of sideslip angle seen in the simulation and, interestingly, this resulted in a slight improvement of the control performance of the system. This was found only to delay the inevitable as, towards the end of this simulation, the flaps were moments from reaching their limits and oscillations would have developed.

The results of the combined control failures show that, for the inner loop, the autopilot was tolerant of the majority of the failures tested. The adaption algorithm did show some control performance improvement in most cases although the time to re-establish the roll set point was longer. The system failed for the most extreme failure tested which prompted the use of the control allocation algorithm. This showed a vast improvement in control performance; however, it was observed that insufficient control authority was present and that the aircraft in all likelihood would be lost. Of note was the effect of the aileron-rudder mix that degraded the control performance demonstrating the sensitivity of the proposed system to feedforward control commands.

6.3.4 Final Test

A final test to demonstrate the difference between the unmodified autopilot and complete proposed control strategy was simulated. A 50% hardover elevator failure combined with a 100% aileron failure test was performed. When the unmodified autopilot was tested with this failure, a complete loss of control was demonstrated. The aircraft reached a bank angle of 70° where it remained. Large oscillations in pitch angle were also observed. In short, this would have resulted in the loss of the airframe. In contrast, the proposed control strategy was able to maintain level flight with only a small roll disturbance. The desired bank angle and desired pitch set point were re-established satisfactorily. The use of flaps was sufficient to maintain this flight condition and shows that for the inner loop control loops, the proposed control strategy has largely met the objectives of the research.

6.4 Middle Loop Simulations

The middle loop simulations provide great insight into the suitability of the proposed algorithms in the higher flight modes of the autopilot. These tests are important for two reasons, the first being the testing of the adaption algorithm to track a constantly changing set point value and the second being the testing of the coupling effect of the adaption algorithms.

The inner loop simulations showed that the adaption algorithm was able to track a constant set point with little interference from other adaption algorithms. In the middle loop tests, the pitch PID controller receives commands from the airspeed controller, both of which have their own adaption algorithms in place. This is complicated by the coupling of the energy and airspeed PID controllers. These simulations provide a means to determine if the coupling of the three PID controllers will initiate a series of instabilities or if the system will be enhanced by the adaption algorithms. Of primary importance, however, is the objective of this research in providing a system that is tolerant to various control actuation failures: hence, the middle loop simulations provide a small step forward in determining the suitability of this proposed control system. This small step allows for a detailed examination of the airspeed and altitude controllers and allows any final tuning of the adaption algorithms.

6.4.1 Control

The simulation was run in "FBW B" mode with no failures being induced in the system. This allowed for the confirmation of stable operation with the coupling of the various PID controllers. It also provided a baseline result with which to compare the failure test results.

It was found that there was a greatly enhanced tracking of airspeed and altitude with the adaption algorithm enabled which is promising considering the coupling of the PID controllers. The results also showed that the adaption algorithm was able to track a moving set point value. While there was some overshoot of the ideal pitch angle at times, this was quickly resolved by the adaption algorithm and the correct tracking of the ideal pitch angle was achieved.

6.4.2 Elevator Failures

The first elevator failure modelled was the 0° deflection failure. This failure had little effect on the aircraft in this flight mode, the only difference being seen in a subtle diversion of the pitch angle. Almost no effect was seen on airspeed or altitude and the adaption algorithm was still able to maintain excellent pitch tracking. Thus, the improvements seen in the control were maintained and the adaption algorithm shows improved tracking of both airspeed and altitude.

The 50% hardover failure of the right elevator provided a surprising result given the success of the previous test. The autopilot, without adaption, showed minor changes to airspeed and altitude results when compared to the control case and the performance, while not exceptional, was adequate. The results with the adaption algorithm enabled showed a very poor control performance that had overshoot values larger than those seen without adaption. The response was also erratic with no indication that the set point for airspeed or altitude would ever be achieved. From this test, it was observed that there was a large proportional gain increase of the airspeed PID controller. It was theorised that this was the cause of the instabilities of the system and thus, the airspeed PID adaption rate was reduced by a factor of 10. The simulation results showed a marked improvement in system performance of the adaption algorithm. However, the performance of system with adaption was still not as good as the performance without adaption. The reduction in the airspeed adaption rate did provide some insight into the potential cause of the problem. The tracking of the pitch angle was seriously inadequate with large variations in ideal pitch and measured pitch. This warranted further investigation into the adaption rates of the pitch controller. The adaption rate was increased by a factor of 10 and the airspeed adaption rate set back to its original value. With these changes, there was a marked improvement in airspeed and altitude tracking and the simulation with adaption showed great improvements when compared to the non-adapted case. The proposed system was found to be tolerant to this failure, albeit only with extensive simulations and changes or "tuning" of the adaption algorithm. The simulation results for the banked flight case highlighted, the fact that the sensitivity of the system to tolerate failures is highly dependent on the flight manoeuvre being performed as the performance of the banked flight case had degraded when compared to the level flight case. However, the system was found to be tolerant of this fault and the adaption algorithm typically improved the performance of the system.

Following this test, the 100% hardover failure of the right elevator test was simulated. The settings determined in the 50% deflection simulations were maintained and it was observed that the adaption algorithm induced a large degradation in control performance when compared to the non-adapted case. This was attributed to large changes in pitch and airspeed proportional gain values. It was noted that there were large and abrupt control deflections for the case with adaption which is likely to have resulted in the poor airspeed and altitude tracking performance. This result was a disappointment in that the adaption algorithm created this reduced performance. It is theorised that, with more adjustment of the adaption rates, this degradation may be resolved but this has not yet been tested.

It was found that although the adaption algorithm generally improved the control performance of the autopilot, in the case of a 100% hardover failure, the adaption algorithm reduced the control performance. It can be concluded that the proposed control system is tolerant to the less severe elevator failure cases.

6.4.3 Aileron Failure

The first aileron failure to be tested for the middle control loop was the 0° deflection failure test. This, like the elevator case, showed little to no effect on the system. The airspeed and altitude were not visibly affected by the aileron failure. This is not surprising as only a small change in pitch angle occurs at the moment of failure which is easily controlled by the autopilot both with and without adaption. The only real evidence of failure lay in the change in bank angle. This change was identical to the changes seen in the inner loop simulation which is unsurprising as the bank angle is still, essentially, controlled by the inner loop with no influence by the middle control loop. Thus, for all of the aileron failures, a result similar to that attained in the inner loop simulations was demonstrated. Of note, however, was the ability of the autopilot to be tolerant to all aileron failures tested. The ability of the system to tolerate these failures is largely due to the control allocation algorithm that made use of the flaps to restore any lack of roll authority. Great improvement was seen with the adaption algorithm in the ground track results. For the 50% hardover and 100% hardover failures, it was found that the change in heading was significantly reduced when the adaption algorithm was enabled. While it was seen in the inner loop simulations that the adaption algorithm increased the time to reach the roll set point value, the reduction in maximum roll error is considerably more important in limiting the change to the aircraft flight path.

It was concluded that the proposed system is tolerant and well suited to aileron failures and that there is a large performance advantage to be gained through using the adaption and control allocation algorithms.

6.4.4 Combined Test

A test was conducted to show the difference in fault tolerance of the unmodified autopilot and the proposed control strategy. This resulted in a successful display of the proposed control strategy to tolerate control actuator faults. For this simulation, a 50% hardover failure of the elevator was combined with a 100% hardover failure of the right aileron. For the unmodified autopilot this failure resulted in a complete loss of control that resulted in a rapid speed build up, dramatic loss in altitude and a very tight downward spiral ultimately resulting in the loss of the aircraft. The proposed control strategy was able to tolerate this fault and successfully maintained airspeed and altitude with only minor disruptions from the set point. The track followed by the aircraft showed a slight change in heading as the failure occurred but this heading was maintained and straight and level flight was achieved. A constant bank angle turn was also performed with this failure and, again, the proposed control algorithm was able to tolerate this failure. Airspeed and altitude were maintained and a circular flight path demonstrated.

This test showed that the proposed autopilot is tolerant of the tested control actuation failures for the middle control loop and that the control performance gain is significant.

6.5 Outer Loop Simulations

The outer loop simulations tested the most advanced form of the autopilot control strategy and provide insight in the suitability of the proposed fault-tolerant design. The outer loop simulation was run in the "Auto" flight mode of the autopilot with three waypoints being used to create a triangular flight path. A constant altitude and airspeed were set for all waypoints. The same three control failures were applied to the elevators and ailerons and a combined failure of the elevator and aileron was tested. In all tests other than the final test, the control allocation algorithm was enabled.

A control was run in which no control failures were induced. This showed that the adaption algorithm created an unattainable solution as, due to the methodology of aerodynamic force characterisation, a runaway condition resulted. It was decided to modify the adaption rate of the roll PID adaption algorithm by a factor of 100 which yielded a highly successful result. The results of the control showed a considerable improvement in airspeed and altitude tracking and showed that the adaption algorithm was able to curb the change in airspeed and altitude resulting from the change in heading to almost perfect levels.

These levels were not observed for the autopilot without adaption despite attempting to improve performance with feedforward pitch and throttle compensation. The ability of the adaption algorithm to achieve this level of performance was very encouraging.

6.5.1 Elevator Failures

The first failure to be tested was a 0° deflection failure of the right elevator. This simulation generated an interesting set of results that showed erratic airspeed and altitude tracking. The waypoint track was, however, somewhat improved. This result was surprising and somewhat disappointing as the adaption algorithm reduced the control performance of the autopilot. Upon examination of the PID gain change for the airspeed PID, it was noted that the adaption algorithm increased the proportional gain of the airspeed PID to very high levels. It was decided to reduce the adaption rate of the airspeed PID adaption controller by a factor of 10. This resulted in greatly improved performance. The performance was not at the level of that seen in the control; however, the change in airspeed and altitude was less than in the case without adaption and the time to reach the set airspeed and altitude was greatly reduced. The results showed a great benefit in using the adaption algorithm.

Of some concern is the sensitivity of the adaption algorithm to the adaption rate. An incorrect value of the adaption rate can quickly lead to the complete breakdown of the simulation and needs to be chosen very carefully. It is also a concern that this adaption rate may well be sensitive to changes in other parameters such as the set airspeed, altitude or aircraft characteristics and may cause unstable behaviour to develop. The sensitivity of the adaption algorithm at this stage is unknown and a recommendation into the research of this aspect of the algorithm will be made.

The second failure to be tested was a 50% deflection failure of the right elevator. This resulted in slightly more difficulty for the non-adapted autopilot in maintaining the set airspeed and altitude but little change to the adapted case. It was found that the trends found in the 0° deflection case were maintained and that the airspeed and altitude tracking with the adaption rate was significantly improved over the non-adapted autopilot. This trend continued for the 100% hardover failure and it was found that, because the allocation algorithm was enabled, the autopilot was relatively unaffected by the failures in this flight mode. Thus, the adaption algorithm combined with the allocation algorithm was found to have superior performance in airspeed tracking, altitude tracking and even in waypoint following.

It was concluded that the proposed fault-tolerant design is indeed tolerant to the control failures tested.

6.5.2 Aileron Failures

The first aileron failure was the 0° deflection failure of the right aileron. This test showed that the control allocation alone was able to tolerate this failure, with little change observed from the control case. It was, however, observed that the adaptation algorithm improved the airspeed and altitude tracking significantly. The control allocation algorithm was important for this failure as the flaps were used to supplement roll control for brief periods during the start of each turn.

The second failure tested was the 50% hardover failure of the right aileron. Surprisingly, very little variation was noted in the flight path of the aircraft. This is likely caused by the fact that the failure occurred during a right hand turn. However, it is noted that the autopilot was able to tolerate this failure without adaption, only making use of the allocation algorithm, although there was a significant improvement on control performance when the adaption algorithm was enabled.

Finally, when a 100% aileron deflection failure was induced, again, little change in airspeed or altitude tracking was observed. There was a slight degradation in waypoint tracking although this was minor. The adaption algorithm provided the same performance gains seen in the previous cases.

It is concluded that the proposed control strategy is tolerant of the aileron failures tested and that there was a great control performance gain when the adaption algorithm was used in conjunction with the control allocation algorithm.

6.5.3 Final Combined Test

This test was deemed to be the ultimate comparison of the autopilot in its unmodified form and the autopilot with the proposed fault-tolerant control strategies. This test involved testing the unmodified autopilot with a 50% hardover failure of the right elevator and a 100% hardover failure of the right aileron and then comparing the results to the case where the control adaption and allocation algorithms were enabled.

The unmodified autopilot completely failed to tolerate this failure. The airspeed increased rapidly to a level beyond that considered acceptable for this airframe, with a corresponding loss in altitude. A very tight spiral dive is seen that would be likely to cause catastrophic structural damage. The proposed control strategy was, however, tolerant to this failure and the results showed that the airspeed and altitude were satisfactorily maintained. The autopilot was also able to maintain the correct waypoint tracking with minor performance degradation when compared to the control case. The aircraft was able to track the waypoints for another complete cycle before the simulation ended. Based on the results of the PID gain changes which were reasonably stable, it is expected that the aircraft would be able to fly in this manner indefinitely without the introduction of instabilities; however, this has yet to be confirmed.

In conclusion, this test demonstrates the success of the proposed strategy to tolerate control actuator failures of a severe nature and shows that the use of control adaption combined with control allocation is necessary to tolerate a control failure of this type with satisfactory control performance.

7 Conclusions and Recommendations

7.1 Conclusions

The current research has shown considerable potential in the proposed design of an intelligent fault-tolerant control system for an unmanned aerial vehicle. It has shown that the proposed system, consisting of a fuzzy logic model reference adaptive controller and a simple daisy chain control allocation algorithm, can be used successfully to tolerate a range of actuator failures, loss of a control surface as shown by the 0° deflection failures, and uncertainties within the aircraft model as shown by the use of the secondary and tertiary control surfaces. The system was tested under a number of different conditions in a number of different flight modes and, in the majority of cases, the system performed well, yielding safe and fault-tolerant flight. This was made particularly evident when comparisons were made between a completely unmodified autopilot and an autopilot with the proposed control architecture. In this respect, this research and the proposed intelligent fault-tolerant system can be considered successful. However, there are several concerns and issues that were identified.

A number of problems were encountered in the research that may limit the ability of this system to function under some circumstances. These problems stem primarily from the intelligent adaption system. The first issue of concern is the sensitivity of system to the correct selection of the adaption rates. Too low an adaption rate results in no real control performance gain and is of little value while too high an adaption rate results in the potential for numerous instabilities to develop. This sensitivity to adaption rate was seen in a number of cases and necessitates the need for significant additional simulation of the system. There is also uncertainty as to the performance of the proposed system for different airframes and autopilot control structures and there may well be issues surrounding changes in aircraft characteristics with changes in airspeed and altitude.

Furthermore, the complexity of a practical autopilot makes the determination of the optimum settings difficult. There are numerous limits, correcting factors and adjustments that can be applied to a greater or lesser extent that all contribute to the successful simulation of fault tolerance. To understand fully the nuances of the system and to determine with confidence that this system can be implemented into a UAS airframe is particularly difficult and time consuming due to the many different combinations of parameters that can be set. This complexity is further increased by the introduction of the proposed system which increases the number of parameters to be adjusted. Thus, while a successful result was demonstrated, it is difficult to say with confidence that the proposed system is fully fault-tolerant in all conditions.

7.2 Recommendations for Future Research

This research has provided great insight into the use of an artificial intelligent adaptive controller combined with a simple control allocation algorithm to create a control strategy that is tolerant to several control actuation failures. However, in this research a number of concerns were raised, and a number of recommendations are required.

The first concern developed from the need to "tune" the fuzzy adaption controller carefully. This tuning was accomplished by adjusting the adaption rates for each PID controller. While in most cases this proved to solve many control issues, it also highlighted several difficulties of the fuzzy adaptive controllers and the implementation thereof. It is recommended that a review and study of the effect of the adaptive rates on the performance of the system be considered. Potentially, the use of global optimisation techniques could be used to improve the tuning of the PID gains.

Another area of concern lies in the difficulties noted with the use of feedforward loops. These feedforward loops are commonly used in real autopilot systems to aid control. These are often in the form of an aileron to rudder coupling, a roll to elevator coupling or a pitch to throttle coupling as is found on the Ardupilot Mega. The adaption algorithm perceives these feedforward couplings as a form of offset error that tends to introduce instabilities within the system as the autopilot attempts to reduce the offset seen. Thus, it is recommended that a more robust fuzzy adaption system be developed. This should have a form of feedforward signal, indicating to the adaption system not to attempt to remove the offset.

Other issues surround the ability of the aircraft to manoeuvre after the failure of a control surface. This was not seen often in these tests especially in the higher flight modes but, with the reduced control authority that develops from the control failure, an introduction of a flight envelope restriction should be applied. This may be in the form of reduced pitch and bank angles, an increase in stall speed, or a reduction in maximum altitude. It is also recommended that a system be developed, that detects that the control authority is reduced and prevents the fuzzy adaption algorithm from increasing the PID gains uncontrollably. This would prevent a situation where the controller introduces instability, as would occur with this system

It is recommended that an additional form of intelligence be introduced where information on the health condition of each control surface is gathered. This information could then be used to aid the adaption and allocation algorithms in controlling the aircraft. This could be accomplished in several ways. The first and perhaps simplest method of control surface monitoring would be the direct measurement of the control actuator deflections. While this technique would be simple, it adds to the hardware complexity and in fact may decrease the fault tolerance of the system as the additional sensors could fail and provide incorrect or spurious data.

The other potential solution lies in the use of a fuzzy logic monitoring algorithm that compares the behaviour of the system to a set of predefined control inputs. As an example, the ailerons and flaps could be moved in unison and in an opposite sense by a predetermined amount that results in no rolling moment. If a roll is then measured, a potential fault then exists and other methods can then be used to determine the point of failure.

It is recommended that extensive testing of the proposed system be conducted through the use of simulations before the system is implemented on an operational UAS. In its current form, the proposed control strategy cannot be guaranteed to perform satisfactorily under all circumstances and for long durations. It is thus recommended that the length of simulation be extended and that a variety of different flight conditions be simulated. These should include turbulence, wind drift, altitude variations and airspeed variations. Other control surface failures that include power plant failures and rudder failures should also be included. The addition of structural failures would also be highly valuable in future research.

Finally, it is recommended that the flight dynamic model be validated through the use of flight tests and that the proposed control strategy in its current form is flown. This flight should be conducted with a "man in the loop" that would allow a pilot to regain control should the proposed control strategy develop instabilities.

References

1. **Christopher Edwards, Thomas Lombaerts, Hafid Smaili.** *Fault Tolerant Flight Control, A Benchmark Challenge.* Berlin : Springer, 2010. e-ISBN 978-3-642-11690-2.

2. All about Military - January 2012. *All about Military.* [Online] January 2012. [Cited: 27 July 2012.] http://1.bp.blogspot.com/_fI1By0fFB5s/S1bhcVmDB-I/AAAAAAAAAVc/oRAF2H8aGyk/s400/Eurofighter+Typhoon+High+G.jpg.

3. Aviation Milestones. *Popular Mechanics.* [Online] 28 December 2009. [Cited: 27 July 2012.] http://www.popularmechanics.com/technology/aviation/news/4340775.

4. The Fight for Iraq. *Wall Street Journal.* [Online] 18 December 2009. [Cited: 27 July 2012.] http://online.wsj.com/article/SB126102247889095011.html.

5. **Magdi S.Mahmoud, Yuanqing Xia.** *Analysis and Synthesis of Fault-Tolerance Control Systems.* Chichester : John Wiley & Sons Ltd, 2013. ISBN 978-1-118-54133-3.

6. *Intelligent fault tolerant control using adaptive and learning methods.* **Yixin Diao, Kevein M. Passino.** 801-817, s.l. : Elsevier Science Ltd - Control Engineering Practice, 2002, Vol. 10. 0967-0661/02.

7. **David Henry, Silvio Simani, Ron J Patton.** *Fault Detection and Diagnosis for Aeronautic and Aerospace Missions.* Berlin : Springer Berlin Heidelberg, 2010. Online ISBN 978-3-642-11690-2.

8. Fuzzy Application Library/What is Fuzzy Logic? *FuzzyTech.* [Online] [Cited: 04 06 2012.] http://www.fuzzytech.com/e/what.html.

9. **Gordon Hayward, Walter Banks.** Fuzzy Logic in Embeded Microcontrollers and Control Systems. *Byte Craft Limited Code Development Systems.* [Online] October 2002. [Cited: 21 May 2012.] http://www.bytecraft.com/Fuzzy_Logic.

10. **Jantzen, Jan.** *Design of Fuzzy Controllers - Technical Report No.98-E 864.* Lyngby : Technical University of Denmark, Department of Automation, 1998.

11. **Venayagomoorthy, Ganesh K.** *Teaching Neural Networks Concepts and their Learning Techniques.* : Proceedings of the American Socciety for Engineering Education Midwest Section Conference,Pittsburg, Kansas, Sept 20 -Oct 1, 2004. p. 5.

12. **Hassan Noura, Didier Theilliol, Jean-Christophe Ponsart, Abbas Chamseddine.** *Fault-tolerant Control Systems, Design and Practical Applications.* London : Springer, 2009. eISBN - 978-1-84882-653-3.

13. *Aircraft Accident Report, United Airlines Flight 232, McDonnell Douglas DC-10,Sioux Gateway Airport, Sioux City, Iowa, July 19, 1986.* Washinton : National Transportation Safety Board, 1990.

14. DHL Plane Struck by Missile in Baghdad. *DW.* [Online] 22 11 2003. [Cited: 16 01 2014.] http://www.dw.de/dhl-plane-struck-by-missile-in-baghdad/a-1039411.

15. **L.F.Mendonca, J.M.G. Sa da Costa, J.M. Sousa.** *Fault Detection and Diagnosis using Fuzzy Models.* : European Control Conference, ECC2003,Cambridge, United Kingdom, 2003. pp. 1-6.

16. **Nigam, Priyank Jain and Dr M.J.** *Design of a Model Reference Adaptive Controrller Using Modified MIT Rule for a Second Order System.* s.l. : Research India Publications - Advance in Electronic and Electric Engineering., 2013. Vol. 3. ISSN 2231-1297.

17. **Dydek, Zachary Thompson.** *Adaptive Control of Unmanned Aerial Systems.* s.l. : Massachusetts Insitute of Technology, 2010.

18. **SU, Wei.** *A Model Reference-Based Adaptive PID Controller for Robot Motion Control of not Explicitly Known System.* s.l. : International Journal of Intelligent Control and Systems, 2007. Vols. 12 No. 3, pp237-244.

19. **Barkana, Itzhak.** *Gain conditions and convergence of simple adaptive control.* : International Journal of Adaptive Control and Signal Processing, 2005. Vols. 19, No.1,pp.13-40.

20. **Farah, Hassan.** *The Fuzzy Logic Control of Aircraft, M.Sc Thesis.* Ottawa, Canada : Department of Mechanical and Aerospace Engineering, Institute for Mechanical and Aerospace Engineering, Ottowa, Canada, 1999.

21. **Sefer Kurnaz, Omer Cetin, Okyay Kaynak.** *Fuzzy Logic Based Approach to Design of Flight Control and Navigation Tasks for Autonomous Unmanned Aerial Vehicles.* s.l. : Journal of Intelligent and Robotic Systems, 2009. Vols. 54, No., pp 229-244.

22. **Omer Cetin, Sefer Kurnaz, Okyay Kaynak.** *Fuzzy Logic Based Approach to Design of Autonomous Landing System or Unmanned Aerial Vehicles.* s.l. : Journal of intelligent and Robotic Systems, 2011, 2010. Vols. 61, No. ,pp239-250.

23. **L.Doitsidis, K.P. Valavanis,N.C. Tsourveloudis, M. Kontitsis.** *A Famework for Fuzzy Logic based UAV Navigation and Control.* s.l. : Proceedings of the IEEE International Conference on Robotics and Automation, New Orleans,LA, April 2004, pp 4041-4046.

24. **Duerksen, Noel.** *Fuzzy Logic Decoupled Longitudinal Control of General Aviation Airplanes.* Hampton, Virgina : National Aeronautics and Space Administration, Langley Research Centre , 1996.

25. —. *Fuzzy Logic Decoupled Lateral Control for Genral Airplanes.* Hampton, Virginia : National Aeronautics and Space Administration, Langley Research Centre, 1997.

26. **Mohammad Hadi Amoozgar, Abbas Chamseddine, Youmin Zhang.** *Fault-Tolerant Fuzzy Gain-Scheduled PID for a Quadrotor testbed in the Presence of Actuator Faults.* : Proceedings of the IFAC Conference on Advances in PID Control, Brescia, Italy, March 28-30, 2012, pp.

27. **Kimberly Bickraj, Thierry Pamphile, Aylin Yenilmez, Ming Li, Ibramhim N. Tansel.** *Fuzzy Logic Based Integrated Controller of Unmanned Aerial Vehicles.* : Proceedings of the Florida Conference on Recent Advances in Robotics FCRAR 2006, Miami, Florida, May 25-26, 2006, pp 1-6, 2006.

28. **Waihon A. Kwong, Kevin M. Passino, Eric G. Laukonen,Stephen Yurkovich.** *Expert Supervision of Fuzzy Learning Systems for fault Tolerant Aircraft Control.* : Proceedings of the IEEE, Vol 83, No.3, March 1995, pp466-483.

29. **Jeffrey R. Layne, Kevin M. Passino.** *Fuzzy Model Reference Learning Control.* : Proceedings of the IEEE conference on Control Applications 1992,Dayton Ohio, 13-16 Sept 1992, pp686-691.

30. **Travis Dierks, S. Jagannathan.** *Neural Network control of Quadrotor UAV Formations.* s.l. : Proceeding of American Control Conference, St Louis, Mo, UAS, June 10-12. 2009, pp2990-2996.

31. **Song, Yong D.** *Fault Tolerant and Reconfigurable Control of Unmanned Aerial Vehicles (UAV).* s.l. : Department of Electrical and Computer Engineering, North Carolina A&T State University, 2008.

32. **Beck, Roger E.** *Application of Control Allocation Methods to Linear Systems with Four or More Objectives, PhD Thesis.* s.l. : Virginia Polytechnic Institude and State University, Blacksburg, Virginia,2002.

33. **Scalera, Kevin R.** *A Comparision of Control Allocation Methods for the F-15 ACTIVE Research Aircraft Utilizing Real-Time Piloted Simulations, Masters Thesis.* : Virginia Polytechnic Institude and State University,Blacksburg, Virginia, 1999.

34. **Susan A. Frost, Brian R. Taylor, Christine V. Jutte, John J. Burken, Khanh V. Trinh, Marc Bodson.** *A Framework for Optimal Control Allocation with Structural Load Constraints.* s.l. : Proceedings of AIAA Atmospheric Flight Mechanics Conference; 2-5 Aug. 2010; Toronto, Ontario; Canada.

35. Case Study: Rockwell Collins demonstrate damage tolerant flight control and autonomous landing capabilities. *Rockwell Collins.* [Online] Rockwell Collins. [Cited: 27 July 2012.] http://rockwellcollins.com/sitecore/content/Data/Success_Stories/DARPA_Damage_Tolerance.aspx.

36. **Lombaerts, Dr Thomas.** *Youmin Zhang and Abbas Chamseddine, Fault Tolerant Flight Control Techniques with Application to a Quadrotor UAV Testbed, Automatic Flight Control Systems - Latest Developments.* Rijeka : InTech, 2012. ISBN: 978-953-307-816-8.

37. **V.Cook, Michael.** *Flight Dynamics Principles,.* Oxford : Elsevier, 2007. ISBN: 978-0-7506-6927-6.

38. **James, Glyn.** *Advanced Modern Engineering Mathematics, Third Edition,.* Essex, : Pearson Education Limited, 2004. ISBN-10: 0-13-045425-7.

39. **Weick, Fred E.** *Aircraft Propeller Design,.* New York, : McGraw-Hill Book Company, Inc., 1930.

40. **Drela, Mark.** DC Motor / Propeller Matching, Lab 5 Lecture Notes. *Massachusetts Insitute of Technology.* [Online] 3 March 2005. [Cited: 22 March 2013.] http://web.mit.edu/drela/Public/web/qProp/motorprop.pdf.

41. **Smetana, Frederick O.** *Computer Assisted Analysis of Aircraft Perfromance Stability and Control.* Raleigh : McGraw Hill Book Company, 1984. ISBN 0-07-058441-9.

42. **Veness, Chris.** Calculate distance, bearing and more between Latitude/Longitude points. *Movable Type Scripts.* [Online] 2010. [Cited: 14 June 2012.] http://www.movable-type.co.uk/scripts/latlong.html.

43. **Ardian, Chris.** Ardupilot-mega APM2 Board. *DIY Drones.* [Online] 7 June 2012. [Cited: 24 July 2012.] http://code.google.com/p/ardupilot-mega/wiki/APM2board.

44. **Chris, Adrian.** Ardupilot-mega Official Arduplane repository. *DIY Drones.* [Online] [Cited: 18 July 2012.] http://code.google.com/p/ardupilot-mega/.

45. **Agostino De Marco, Eugene L. Duke, Jon S. Berndt.** *A General Solution to the Aircraft TRim Problem.* Hilton Head, South Carolina : American institute of Aeronautics and astronautics, 2007. Vol. 6703.

46. **C. Ostowari, D. Naik.** *Post-Stall Wind Tunnel Data for NACA 44XX Series Airfoil Sections .* Colorado : U.S. Department of Energy, 1985. Contract No. DE-AC02-83CH10093.

47. **Mostafa Mohammadi, Alireze Mohammad Shahri.** *Adaptive Nonlinear Stabilization Control for a Quadrotor UAV: Theory, Simulation and Experimentation.* Tehran : Springer Science+Buisiness Media, 2013. Vol. 72. DOI 10.1007/s10846-013-9813-y.

Appendix A - Continuation of the Development of the Equations of Motion for a Rigid Non-Symmetric Aircraft

Continuing with the development of the equations of motion described in Chapter 2.5 and rearranging (2.6) results in

$$\dot{r} = \frac{N + \left(I_x - I_y\right)pq + I_{yz}(pr + \dot{q}) - I_{xz}(qr - \dot{p}) - I_{xy}(q^2 - p^2)}{I_z} \tag{A.1}$$

Rearranging (A.1) further results in

$$\dot{r} = \frac{N + \left(I_x - I_y\right)pq + I_{yz}(pr) - I_{xz}(qr) - I_{xy}(q^2 - p^2)}{I_z} + \frac{I_{yz}\dot{q}}{I_z} + \frac{I_{xz}\dot{p}}{I_z} \tag{A.2}$$

To aid in the development of the equations, a substitution is made:

$$A = \frac{N + \left(I_x - I_y\right)pq + I_{yz}(pr) - I_{xz}(qr) - I_{xy}(q^2 - p^2)}{I_z} \tag{A.3}$$

(A.2) therefore becomes

$$\dot{r} = A + \frac{I_{yz}\dot{q}}{I_z} + \frac{I_{xz}\dot{p}}{I_z} \tag{A.4}$$

Now Substituting (A.4) into (2.5) results in

$$I_y\dot{q} + (I_x - I_z)pr + I_{yz}\left(pq - \left(A + \frac{I_{yz}\dot{q}}{I_z} + \frac{I_{xz}\dot{p}}{I_z}\right)\right) + I_{xz}(p^2 - r^2) - I_{xy}(qr + \dot{p}) = M \tag{A.5}$$

Rearranging (A.5) yields

$$\dot{q} = \frac{I_z\left(M - (I_x - I_z)pr - I_{yz}\left(pq - A - \frac{I_{xz}\dot{p}}{I_z}\right) - I_{xz}(p^2 - r^2) + I_{xy}(qr + \dot{p})\right)}{I_yI_z - I_{yz}^2} \tag{A.6}$$

This then becomes

$$\dot{q} = \frac{I_z\left(M - (I_x - I_z)pr - I_{yz}(pq - A) - I_{xz}(p^2 - r^2) + I_{xy}(qr)\right)}{I_y I_z - I_{yz}^2} + \left(\frac{I_{xz}I_{yz} + I_{xy}I_z}{I_y I_z - I_{yz}^2}\right)\dot{p} \tag{A.7}$$

Again, to simplify further development, another substitution is made:

$$B = \frac{I_z\left(M - (I_x - I_z)pr - I_{yz}(pq - A) - I_{xz}(p^2 - r^2) + I_{xy}qr\right)}{I_y I_z - I_{yz}^2} \tag{A.8}$$

Thus (A.7) reduces to (A.9)

$$\dot{q} = B + \left(\frac{I_{xz}I_{yz} + I_{xy}I_z}{I_y I_z - I_{yz}^2}\right)\dot{p} \tag{A.9}$$

Finally, substituting (A.4) into (2.4) results in

$$I_x\dot{p} - \left(I_y - I_z\right)qr + I_{xy}(pr - \dot{q}) - I_{xz}\left(pq + A + \frac{I_{yz}\dot{q}}{I_z} + \frac{I_{xz}\dot{p}}{I_z}\right) + I_{yz}(r^2 - q^2) = L \tag{A.10}$$

After some manipulation is transformed into:

$$\left(\frac{I_x I_z - I_{xz}^2}{I_z}\right)\dot{p} - \left(\frac{I_{xy}I_z + I_{xz}I_{yz}}{I_z}\right)\dot{q} = L + \left(I_y - I_z\right)qr - I_{xy}pr + I_{xz}pq - I_{yz}(r^2 - q^2) + I_{xz}A \tag{A.11}$$

Substituting (A.9) into the equation above yields

$$\begin{aligned}\left(\frac{I_x I_z - I_{xz}^2}{I_z}\right)\dot{p} - \left(\frac{I_{xy}I_z + I_{xz}I_{yz}}{I_z}\right)\left(B + \left(\frac{I_{xz}I_{yz} + I_{xy}I_z}{I_y I_z - I_{yz}^2}\right)\dot{p}\right) \\ = L + \left(I_y - I_z\right)qr - I_{xy}pr + I_{xz}pq - I_{yz}(r^2 - q^2) + I_{xz}A\end{aligned} \tag{A.12}$$

Therefore,

$$\dot{p} = \frac{\left(L + \left(I_y - I_z\right)qr - I_{xy}pr + I_{xz}pq - I_{yz}(r^2 - q^2) + I_{xz}A + B\left(\frac{I_{xy}I_z + I_{xz}I_{yz}}{I_z}\right)\right)}{\left(\frac{I_x I_z - I_{xz}^2}{I_z}\right) - \left(\frac{I_{xy}I_z + I_{xz}I_{yz}}{I_z}\right)\left(\frac{I_{xz}I_{yz} + I_{xy}I_z}{I_y I_z - I_{yz}^2}\right)} \tag{A.13}$$

And finally:

$$\dot{p} = \frac{\left(I_y I_z - I_{yz}^2\right)\left\{\, I_z\left[L + \left(I_y - I_z\right)qr - I_{xy}pr + I_{xz}pq - I_{yz}(r^2 - q^2) + I_{xz}A\right] + B\left(I_{xy}I_z + I_{xz}I_{yz}\right)\right\}}{I_z(I_x I_z - I_{xz}^2)\left(I_y - I_{yz}^2\right) - \left(I_{xy}I_z + I_{xz}I_{yz}\right)\left(I_{xz}I_{yz} + I_{xy}I_z\right)} \tag{A.14}$$

Appendix B - Blade Element Propeller Modelling

Blade element theory has been used extensively in the modelling of propeller blades. The method was first published in 1878 by William Froude and concurrently by Stefan Drzewiecki. [39] The method divides the propeller into a number of chord wise segments as shown in Figure B.1 and examines the forces generated by that segment. These forces are then integrated to determine the total net force acting on the propeller.

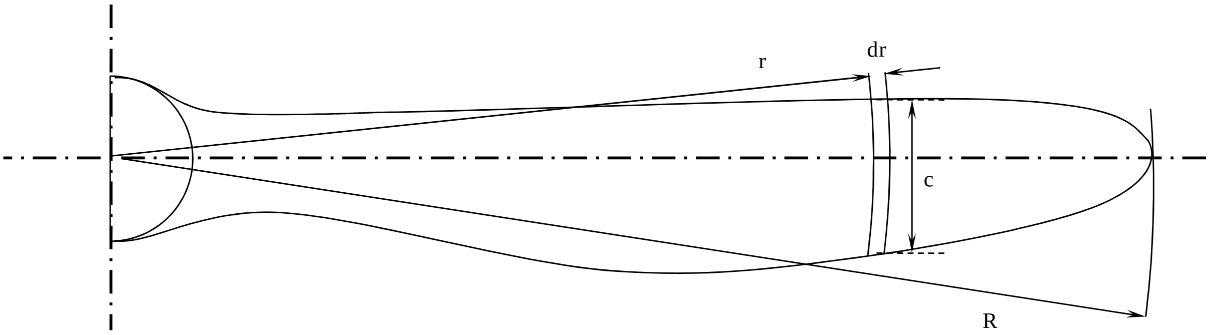


Figure B.1: Blade element theory diagram adapted from [39].

Propeller Geometry

Propellers are typically described by two geometrical properties, namely, the Diameter (D), and the Pitch (P). However, in order to perform a blade element method on a propeller the chord distribution and blade angle distribution is required. This is typically obtained from the manufacturers of the propeller. For the case of the Xoar 23x10 propeller used in this analysis, the geometry was obtained from the work of a previous student, Mr N Moore. The blade angle was determined making the assumption that propeller had uniform geometrical pitch of 10”. This resulted in the geometry shown in Figure B.2.

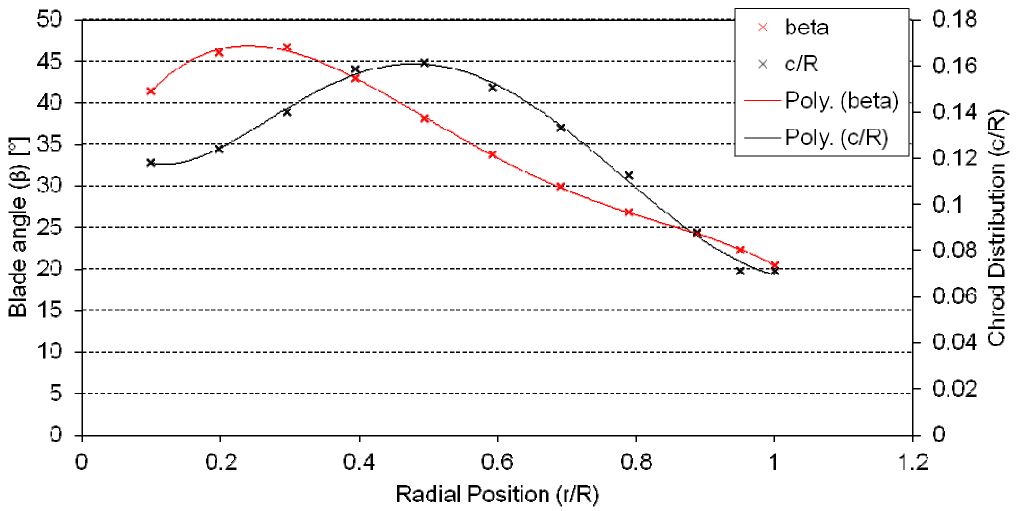


Figure B.2: Geometry of Xoar 23x 10 Propeller

Propeller Aerofoil

Unfortunately, the aerofoil of the propeller was not known from the manufacturer. However, a commonly used aerofoil for model propellers is the NACA 4412. This discrepancy will have a negative impact on the accuracy of the analysis; however, it is thought that in the context of the aircraft as a whole, this effect will be minor. Especially since the focus of this research is on the dynamic behaviour of the aircraft rather than the performance of the aircraft. In many circumstances, particularly at low advance ratio, the blade angle far exceeds the aerofoil's stall angle and, therefore, post stall behaviour is required. This was sourced from [46] and is presented in Figure B.3 and Figure B.4. The lift and drag data was characterised by a set of polynomials, for pre stall and post stall behaviour. The polynomial coefficients used in the blade element code are given in Table B.1.

Table B.1 Polynomial coefficients describing the lift and drag characteristics of the NACA 4412 aerofoil

	Lift coefficient		**Drag coefficient**	
Polynomial coefficient	Alpha<=26	Alpha>26	Alpha<=26	Alpha>26
a0	0.2391	-0.8779	0.0196	-0.4347
a1	0.0947	0.10567	-0.0061	0.0203
a2	0.0007	-0.0016	0.0008	0.0007
a3	-0.0002	6.7013E-06	7.5751E-06	-9.1354E-06
a4	1.7219E-06	-5.0079E-09	-2.4485E-07	2.8181E-08

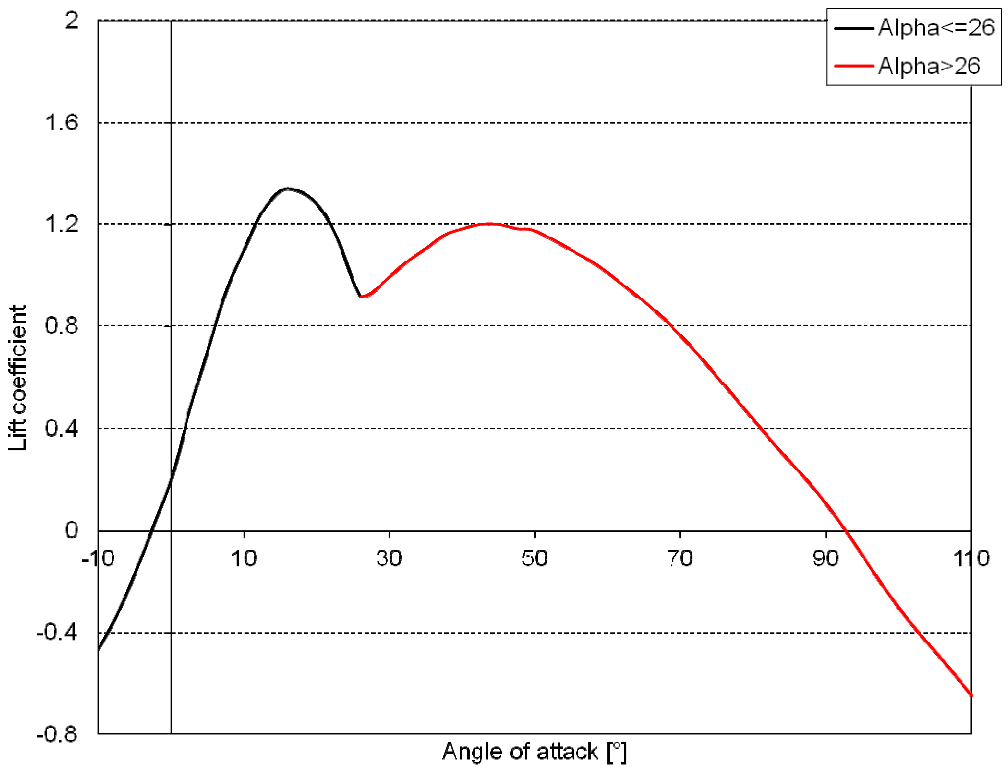


Figure B.3: Lift coefficient variation with angle of attack for the NACA 4412 aerofoil

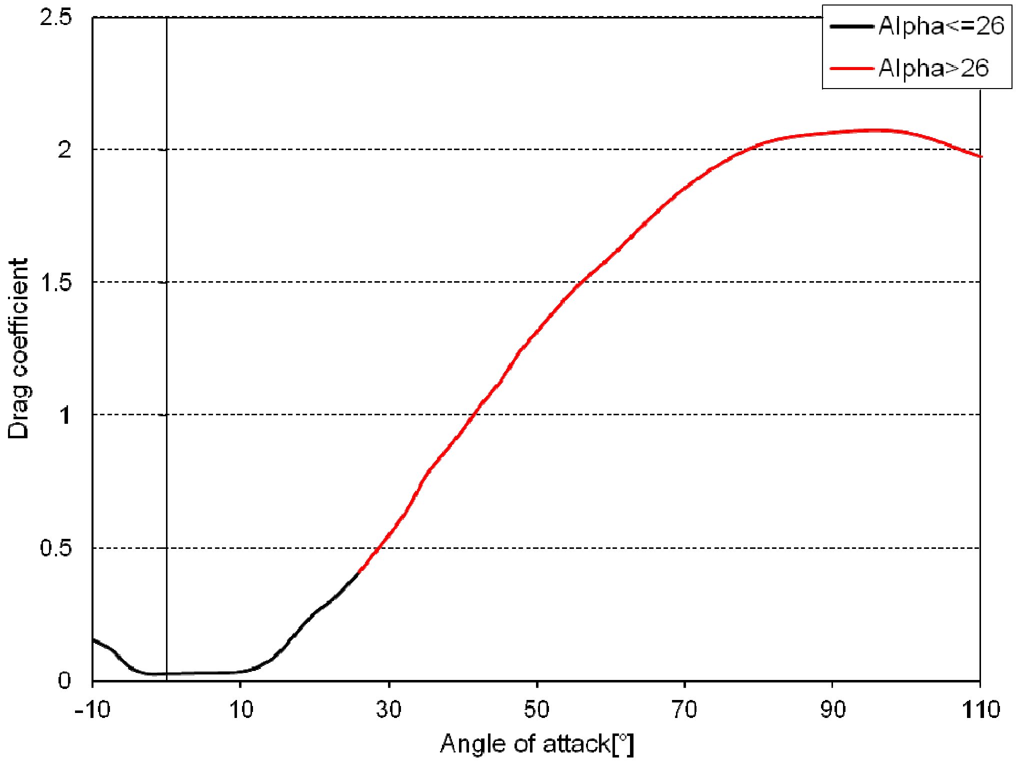
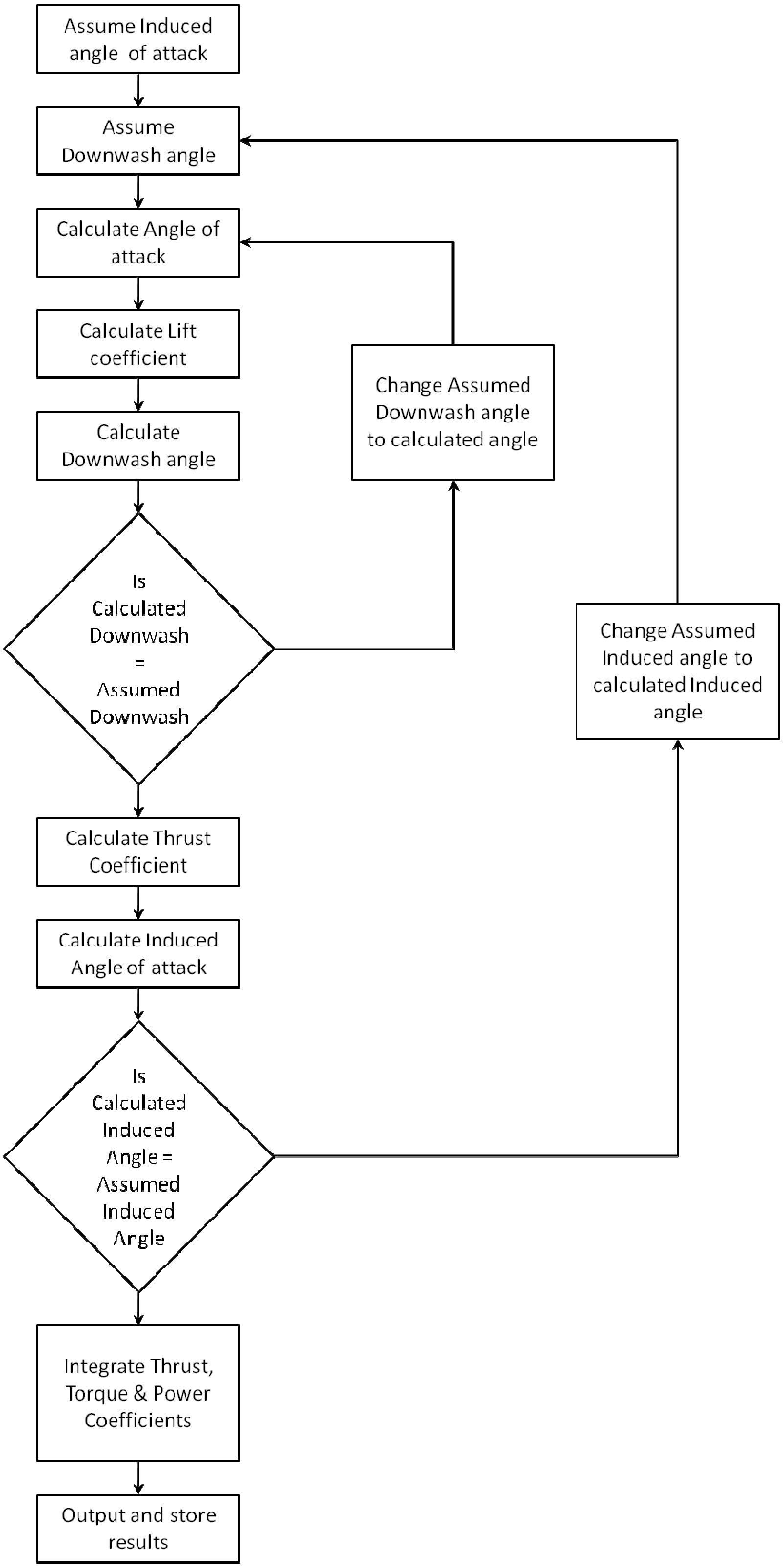


Figure B.4: Drag coefficient variation with angle of attack for the NACA 4412 aerofoil

The blade element code developed followed a highly iterative logic as shown in Figure B.5. The code was run for each increment in advance ratio from static conditions until the windmill state of the propeller was attained.



Assume Induced angle of attack
Assume Downwash angle
Calculate Angle of attack
Calculate Lift coefficient
Calculate Downwash angle
Is Calculated Downwash = Assumed Downwash
Change Assumed Downwash angle to calculated angle
Calculate Thrust Coefficient
Calculate Induced Angle of attack
Is Calculated Induced Angle = Assumed Induced Angle
Change Assumed Induced angle to calculated Induced angle
Integrate Thrust, Torque & Power Coefficients
Output and store results

Figure B.5: Blade element prediction code logic diagram

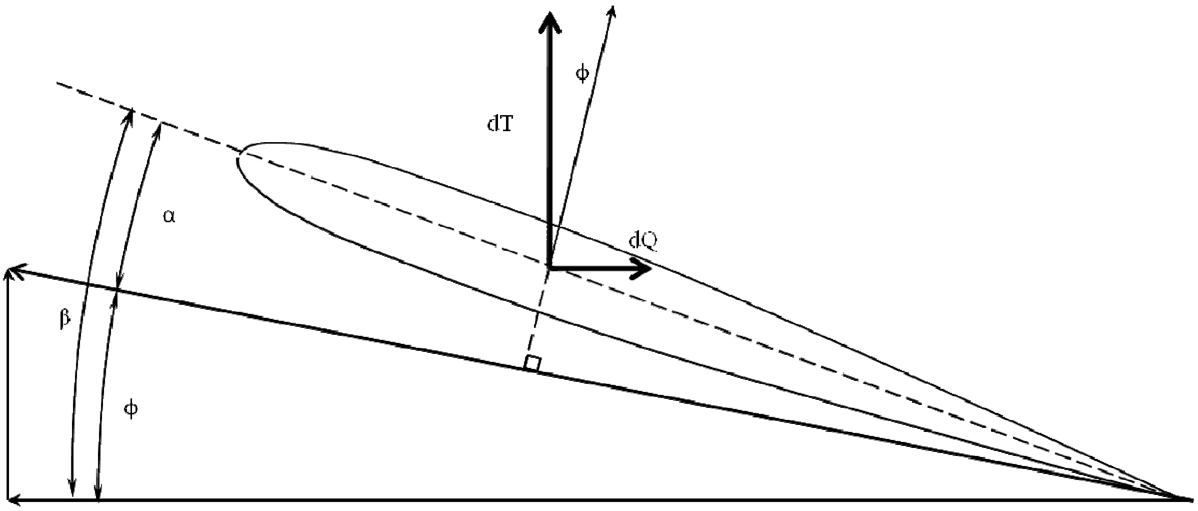


Figure B.6: Aerofoil element used in the blade element prediction code

Using Figure B.6, the angle of attack of the blade element is

$$\alpha = \beta - \phi - \alpha_i - \varepsilon \qquad \text{(B.1)}$$

where β is the geometric blade angle, ϕ is the path angle, α_i is the induced angle of attack, and ε is the downwash angle from the previous blade.

The path angle is found using

$$\phi = \arctan\left(\frac{V}{2\pi r n}\right) \qquad \text{(B.2)}$$

Making use of the advance ratio

$$J = \frac{V}{nD} \qquad \text{(B.3)}$$

This can be transformed into

$$\phi = \arctan\left(\frac{J}{r\pi}\right) \qquad \text{(B.4)}$$

This then relates the angle of attack of the aerofoil section to advance ratio rather than velocity. The induced angle is determined through an iterative scheme as it varies with the amount of thrust produced. Initially; however, an arbitrary value for induced angle and downwash are chosen, allowing for the determination of angle of attack. The lift coefficient is determined using the polynomials of Figure B.3. and Figure B.4. Downwash is determined using an approximation from

Prandtl lifting line theory and assuming that a ideal lift distribution is present. Thus the downwash can be simply expressed as

$$\varepsilon = \frac{CL}{\pi ARe} \tag{B.5}$$

This unfortunately also affects the angle of attack, and thus the downwash is iterated, until a converged solution occurs. Thereafter, the thrust coefficient can be estimated

$$dCT = \frac{1}{2}(J^2D^2 + 4\pi^2r^2)c\big(clcos(\gamma) - cdsin(\gamma)\big) \tag{B.6}$$

This in turn affects the induced angle of attack given by

$$\alpha_i \ = \ Arctan\left(\frac{\frac{1}{2}\sqrt{\left(J^2D^2 \ + \ \left(\frac{D^4}{\pi r}\right)dCT\right)} - \frac{JD}{2}}{\sqrt{\left((J^2D^2) + \ 4\pi^2r^2\right)}}\right) \tag{B.7}$$

where $\gamma = \ \phi + \alpha_i + \varepsilon$.

The procedure is iterated again until the induced angle of attack has converged, at which point, dCT and dCQ are calculated using (B.6) and (B.8):

$$dCQ = \frac{1}{2}(J^2D^2 + 4\pi^2r^2)c\big(clsin(\gamma) - cdcos(\gamma)\big) \tag{B.8}$$

The power coefficient and efficiency can be determined using

$$dCP = 2\pi dCQ \tag{B.9}$$

These quantities are then integrated numerically, using Simpson's integration to obtain the total thrust and torque coefficients of the propeller.

Appendix C - Digital Appendix

This appendix (attached CD) contains all of the wind tunnel data, the flight dynamic code and a full set of results for the simulations run.