CRANFIELD UNIVERSITY

Chang How Lo

Integrated Approaches to handle UAV actuator fault

School of Aerospace, Transport and Manufacturing

PhD

Academic year 2015

Supervisors: Dr Hyo-Sang Shin, Professor Antonios Tsourdos

August 2015

This thesis is submitted in partial fulfillment of the requirements for the Degree of PhD.

Abstract

Unmanned Aerial Vehicles (UAV) has historically shown to be unreliable when compared to their manned counterparts. Part of the reason is they may not be able to afford the redundancies required to handle faults from system or cost constraints. This research explores instances when actuator fault handling may be improved with integrated approaches for small UAVs which have limited actuator redundancy.

The research started with examining the possibility of handling the case where no actuator redundancy remains post fault. Two fault recovery schemes, combing control allocation and hardware means, for a Quad Rotor UAV with no redundancy upon fault event are developed to enable safe emergency landing.

Inspired by the integrated approach, a proposed integrated actuator control scheme is developed, and shown to reduce the magnitude of the error dynamics when input saturation faults occur. Geometrical insights to the proposed actuator scheme are obtained. Simulations using an Aerosonde UAV model with the proposed scheme showed significant improvements to the fault tolerant stuck fault range and improved guidance tracking performance.

While much research literature has previously been focused on the controller to handle actuator faults, fault tolerant guidance schemes may also be utilized to accommodate the fault. One possible advantage of using fault tolerant guidance is that it may consider the fault degradation effects on the overall mission.

A fault tolerant guidance reconfiguration method is developed for a path following mission. The method provides an additional degree of freedom in design, which allows more flexibility to the designer to meet mission requirements.

This research has provided fresh insights into the handling UAV extremal actuator faults through integrated approaches. The impact of this work is to expand on the possibilities a practitioner may have for improving the fault handling capabilities of a UAV.

To my family

Contents

List of Tables

List of Figures

Nomenclature

Symbol /acronym	Description
FTC	Fault Tolerant Control
FDD	Fault Detection and Diagnosis
δ_{rc}	roll virtual command
δ_{pc}	pitch virtual command
δ_{yc}	yaw virtual command
CON	controller
ACT	actuator
δ_i	i^{th} actuator position
$\mathbf{B_{ca}}$	allocation matrix
$\mathbf{P_{ca}}$	deallocation matrix
δ_m	virtual demand
δ_{mc}	virtual demand command
δ	actuator position
$\mathbf{K}$	actuator controller
$\mathbf{G}$	actuator plant
ϕ_c	flight control roll command
Ay_c	lateral acceleration command

Chapter 1

Introduction

1.1 Background

Unmanned Aerial Vehicle (UAV) is gaining popularity nowadays in alleviating the manned air planes in dull, dirty and dangerous missions. The biggest difference between a UAV and the manned aircraft is the human pilot. The weight reduction gained in removing the pilots and their life support systems often result in increased flight endurance, or increased payload carrying capability when compared to conventional aircraft. This presents an attractive and at the same time less costly platform, to decision makers during purchases. Also, the absence of the pilot removes any possibility of having fatalities when used in dirty, dull and dangerous missions over hostile airspace.

However, the removal of the pilot also presents many challenges. When faults occur, the human pilot on-board is capable of making decisions and can control the aircraft to safety. While the UAV has an equivalent ground operator performing the same role in the ground control station (GCS), the lack in tactical feedback and the delays due to communication links are obstacles faced by the ground operator in achieving similar effective responses. Appropriate fault detection and decision making by the on-board flight control computer can aid in reducing the workload, as well as improving the chances of faults becoming systems failures.

While affordable, issues with safety and reliability have hounded the UAV industry. Current UAVs have mishap levels one to two orders higher than manned aircraft levels [2]. This is illustrated in Fig. 1.1. UAV reliability is therefore important because it underlies their affordability (an acquisition issue), their mission availability (an operations and logistics issue), and their acceptance into civil airspace [3]. Ref. [1] concludes with the statement that improving UAV reliability is the single most immediate and long reaching need to ensure their success. With the proliferation of UAVs, and the push for them to operate in non-segregated airspace, there is impetus for the UAVs to improve on their safety and reliability records.

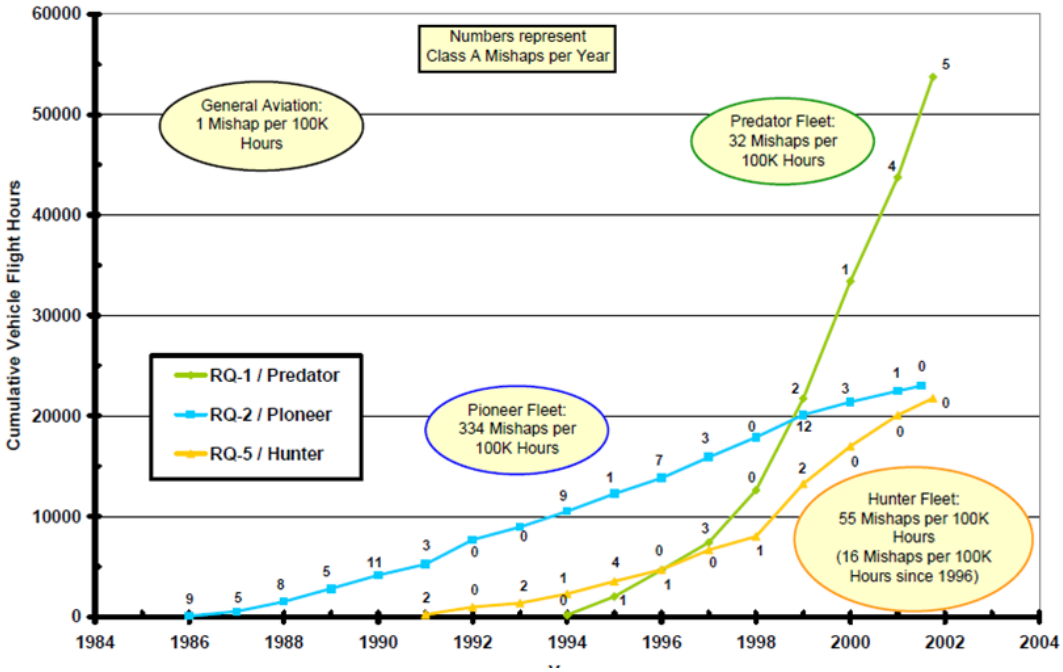

Figure 1.1: UAV failure rates illustrating the high mishap rates of UAVs compared to general aviation aircraft, taken from Ref. [1]

One way to improve the system's reliability is to increase the level of hardware redundancies. However, this approach can greatly increase the cost and complexity of the UAV, which makes them a much less attractive proposition. Moreover, many micro and mini UAVs may not be able to afford the weight or volume of these hardware redundancies without reducing their endurance or payload capability due to their extremely limited physical constraints. Another obvious method may be to use better quality and more reliable components to reduce failure rates. However, this method may not be feasible due to the usually higher costs associated with better quality components, or the technology may not be currently available. An alternative to hardware redundancy is to exploit the analytical redundancy often found in systems.

Due to the complexity in research to complete an active Fault Tolerant Control (FTC) system, many researches have concentrated on either the fault detection, and isolation algorithms or on the reconfiguration control for recovery [4]. A potential problem with these modular approaches during application is they may not be fully optimized to complement each other for overall scheme improvements. For example, switching between models after fault isolation may not

be bumpless [4]. Some Fault Detection and Diagnosis (FDD) techniques are not compatible with some reconfiguration control techniques, as they do not provide sufficient diagnosis details required by the controller for redesign.

Often, the actuator FDD algorithms require full actuator state feedback for fault detection and diagnosis. In Ref. [5], a decentralized FDD tecnique, using only actuator position feedback is combined with control allocation, has been successfully simulated on a Boeing 747 benchmark. Airbus has also described the FDD and FTC practices in Ref. [6], using examples from its latest aircraft A380 airplane.

Although there are many operational UAVs employing some forms of fault tolerant control, few papers have been published on the industrial FTC techniques used on them. Recently, research papers describing complete active FTC designs have been published. An Aerosonde UAV FTC strategy to handle actuator faults for the Aerosonde UAV have been published in Refs. [7] and [8] using Unknown Decoupled Input Observers (UDIO) for fault diagnosis, and Robust Eigenstructure assignment for reconfiguration control. However, guidance is not included in the FTC design, which is necessary for any practical UAV implementation to manage the mission profile after fault. Another research book describes a fault tolerant control and guidance using MMAE for FDI and control allocation (CA) for reconfiguration control Ref. [9]. A separate standalone reconfigurable guidance law is also proposed in the work.

Through these examples, the complexity in implementing a complete fault tolerant control structure for a UAV is apparent. In Ref. [10], a basic review of the possible fault detection techniques to best handle the various potential fault types are presented. More detailed and comprehensive references on the selection of fault tolerant and diagnosis methods are presented in Ref. [11], [12], and [13]. Collectively, they form some of the comprehensive works in helping the reader appreciate and decide on the methods to deploy for their application.

1.2 Motivation

Fault Tolerant Control is a very well established field, with numerous successful applications in many industries [14]. Fault Tolerant Control aims to improve

the ability of a system to tolerate component faults so as to maintain its stability and performance within acceptable/ desired levels [10]. It makes use of existing hardware or analytical redundancies in the system to mitigate the effects of the faults, and prevent them from further deteriorating to become system failure [11].

One might then ask: What are the significant failure modes for UAVs? Does Fault Tolerant Control mitigate the significant failure modes of UAVs? Research done in Ref. [1] showed that flight controls and propulsion failures form more than 50% of the causes for UAV mishaps for US and Israel UAVs, as shown in Fig. 1.2. These significant causal factors may be tackled using Fault Tolerant Control.

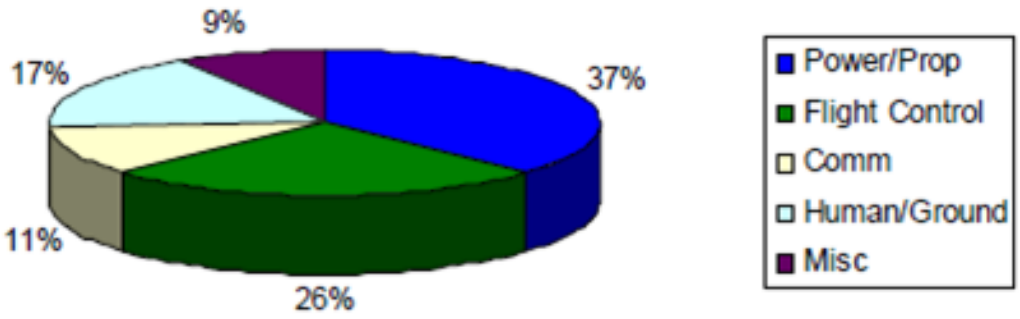

Figure 1.2: Averaged causes of UAV system failures causes, taken from Ref. [1]

Many reasons have contributed to continued interest in this subject. One of them is the rapid improvement in enabling advances in avionics. Examples include shrinking form factors with System on Chip (SOC) designs to enable more hardware redundancies or new architectures with smarter sensors and actuators. Moreover, faster embedded computers with increased memory with reconfigurable computational power, through advances in developments in field programmable gated arrays (FPGAs), are becoming commonplace. These advances allow for many new UAV system designs and concepts which improve the endurance, agility and mission effectiveness.

Due to enabling technological advances in sensing, actuation and computational power, opportunities to improve on small UAVs may be presented by improving fault tolerant strategies to take advantage of these new advances. One approach that is increasingly common in today's world is to integrate functionality and components into integrated systems to take advantage of the technological advances to offer smaller/ more competitive products. An approach which is en-

abled by the smaller sensors is in the integrated vehicle health management field to predict and manage faults before they occur [15]. Prognostics and health management is further enhanced by the minimization of enabling sensors [16], [17], [18].

Increasingly, trends to treat the actuation system control design as a whole system approach have become prevalent. Examples include novel actuator designs like hierarchical actuators consisting of integrated actuators [19], plasma actuators [20], and Micro-Electro-Mechanical Systems (MEMs) based actuators [21]. Some of these actuators are already applied in flight control of UAVs [22], [23]. Due to the different physical architectures, control strategies/ architectures may need to be refined to exploit the new actuator forms and the analytical redundancy within. However, in many existing UAVs, it is often more difficult to add actuators for the UAV compared to flight sensors due to the need for the actuators to be positioned for control effectiveness. Also, electronics minimization may offer the possibility for additional sensors to be available in replacement of current ones should technology advances. Thus, the motivation here is to pursue the study of actuator faults.

Already, integrated approaches have been researched on missile and UAV applications as a means to improve its overall performance. Traditionally, guidance and control algorithms in a flying platform are designed separately. Controller designs are primarily aimed at stabilization of the platform, while guidance aims to bring the platform to meet the mission profile. While decoupling the objectives allow for easier design process, the designs may not be able to fully exploit the full performance capabilities of the platform, due to the need for separation of bandwidths between guidance and control algorithms to ensure stabilization in a cascaded loop design. Many Integrated Guidance and Control (IGC) design approaches combine the guidance and control designs together to remove the conventional bandwidth separation requirement between the guidance and control loops for stability considerations. This enables the designs to improve on their performance due to the tighter coupling achieved between guidance and control. Some prominent examples include missiles requiring high performance effectiveness, like in hit to kill missiles [24], or moving mass actuated kinetic warheads [25]. Other examples include re-entry vehicles [26] and UAVs [27], [28]. These examples showed the superior performance of the integrated designs over

traditional modular designs.

In the area of fault tolerant control research, review papers have suggested at the potential of an improvement in fault detection and handling via integration. In Ref. [4] which reviewed existing fault tolerant control techniques, better integration between fault detection and reconfiguration controller has been suggested as a way to reduce fault deterioration effects. In Ref. [29], current results on fault tolerant guidance had been reviewed as being "not sufficiently explored and needs more methodological work" to complement existing current fault tolerant control techniques. These motivate the author's research direction to be geared towards finding integrated approaches to improve on existing designs. Specifically, can one use combined approaches to improve on fault tolerance performance of a system?

Fault tolerant designs may be complicated by the few physical or analytical redundancies available in the system for many small existing UAVs. Yet, the overall system may need to provide for a prescribed level of safety should faults occur. A strong motivation for research would thus be to look into fault tolerant strategies possibilities, especially for UAVs with no potential analytical redundancy post fault occurrence. This is a practical problem faced by UAV designers in which research insights into the matter will be useful.

The motivation for this research is therefore to focus on performance aspects of handling actuator faults for UAVs. Specifically, two research questions are asked.

1. What can be done if limited redundancy is available in small UAVs?
2. Can integrated/ combined approaches be used to alleviate severe actuator faults?

1.3 Research aim and approach

From the research motivations described in the previous section, the research here aims to investigate and propose ways to improve the handling of actuator faults for UAVs. Two research directions are identified and pursued in this work.

As the fastest control loops in a system lies in the actuation system, the first research aims to investigate the possibility of improving the fault handling from

within the actuation system. An actuator control scheme combining control allocation into the actuation system has been already being proposed [30]. By handling an individual actuator's fault within the actuation system, improved fault handling performance may be achieved. However, the theoretical insights to the scheme have not been fully investigated. The research here aims to establish the properties of the integrated scheme, and show the improvements to handling actuator faults using the integrated scheme.

The second research direction investigates the usage of fault tolerant guidance to bring about improved mission performance with fault occurrence. Usually, a path following mission's guidance may consist of a path planner and a path following algorithm. The research focus here investigates the possibility of accommodating the degraded system's capability through reconfiguring the guidance scheme's parameters. The idea to pursue here is to investigate the possibility of providing flexibility and improving mission success rates while accommodating to the degraded system capability. By utilizing both guidance modules parameters, a proposed reconfiguration is able to provide more flexibility for the designer to handle the fault degradation effects, thus improving mission success rates.

Together, the intent of the research is to improve current small UAVs' ability to handle actuator faults. Small UAVs are often more limited by the amount of redundancy available; hence integrated approaches to synergize between various parts of a fault tolerant scheme are proposed.

1.4 Contributions

Overall, this research has provided fresh insights into the objective of handling UAV extremal actuator faults through combined approaches. The impact of this work is to expand on the possibilities a practitioner may have for improving the fault handling capabilities of a UAV. The main contributions of this work are:

1. A proposed integrated actuator control scheme is shown to reduce the magnitude of the error dynamics compared to a conventional scheme when input saturation faults occur.

2. Geometrical insights to the proposed actuator scheme are shown.

3. Simulations using an Aerosonde UAV model with the integrated actuator control scheme showed significant improvements to the fault tolerant stuck fault range.

4. A proposed fault tolerant guidance reconfiguration method is developed for a path following mission. The method provides an additional degree of freedom in design, which allows more flexibility to the designer to meet mission requirements.

1.5 Research publications

Since the beginning of research, the following publications have been prepared:

1. Chang How Lo, Hyo-Sang Shin, Seungkeun Kim, and Antonios Tsourdos Modeling and Simulation of Fault Tolerant Strategies for A Quad Rotor UAV. In *AIAA Modeling and Simulation Technologies Conference, Minnesota* ,2012.

2. Chang How Lo, Hyo-Sang Shin, Antonios Tsourdos, and Seung-Hwan Kim. Improving the performance of an actuator control scheme during saturation. In *Advances in Aerospace Guidance, Navigation and Control*, pages 15–27. Springer, 2013.

3. C.H. Lo, H.S. Shin, A. Tsourdos, and S.H. Kim. Improving the performance of an actuator control scheme during saturation. In *Euro GNC 2013 – 2nd CEAS Specialist Conference on Guidance, Navigation & Control, Delft*, pages 204–216, 2013.

4. C.H. Lo, H.S. Shin and A. Tsourdos. A Guidance Reconfiguration Strategy to Handle Aileron Actuator Fault. In *MED 2015 – 23rd Mediterranean Conference on Control and Automation, Torremolinos, Spain*

1.6 Organization of thesis

The organisation of this thesis is as follows: The rest of this chapter describes the background and motivation leading to the research objectives of this research.

Following on, Chapter 2 provides a review of the existing literature on the developments in fault tolerant control and integrated guidance and control areas. This provides an extensive background to focus the research done by the author.

In Chapter 3, an integrated actuator control scheme which integrates control allocation and actuator controller together is shown analytically to be superior in performance during saturation or input faults. Using frequency domain analysis, it is shown that the magnitude of error dynamics of the proposed scheme is smaller compared to the conventional scheme. Chapter 4 extends the theoretical analysis of the integrated actuator control scheme. Using discrete time domain analysis, geometrical insights to the improvement in the performance of the integrated scheme over the conventional scheme is obtained.

In Chapter 5, the integrated actuator control scheme is tested with an Aerosonde UAV model with flight control and guidance. The simulation results showed significant improvement in the actuator fault handling range with a classical flight control. Also, guidance tracking error performance is improved.

Ultimately, mission success is the most important measure for any UAV operation. With faults affecting the capability of the system, guidance schemes may be reconfigured to improve on the mission success and safety. Chapter 6 describes a proposed method to reconfigure a guidance scheme to accommodate the degraded system's capability. The new method improves the guidance performance of the system, and provides an additional degree of freedom to the designer to trade between guidance performance and path radius change.

In Chapter 7, conclusions to this work are presented. Here, the work presented is linked to the research aims and directions. Finally, Chapter 8 details future work envisioned by the author to further the presented work.

Chapter 2

Literature Review

2.1 Introduction

Modern flying systems rely on sophisticated control systems to meet performance and safety requirements. At the same time, the increased sophistication in control systems may lead to unsatisfactory performance, even to the extent of instability, when faults in actuators, sensors or other components in the system occur. To address this problem, Fault Tolerant Control (FTC) design methodologies have been researched on as mitigating approaches to tolerate and maintain stability and certain performances with component malfunctions.

Extensive research has been carried out on this topic, leading to numerous Fault Detection and Diagnosis techniques. Many excellent literature reviews had been published to educate on this important topic and progress. These papers range from general reviews [31] [32] [33] [34], to industry specifics like flight control systems [35][29].

In this chapter, a brief review of the concepts of the FTC is introduced. The review is not intended to be exhaustive, but aims to provide a brief insight of some of the main concepts in model based FTC. Besides model based methods, there exists many other methods like data based methods such as neural networks, signal models methods like Fast Fourier Transform (FFT), and Principal Component Analysis (PCA), which are not covered in this review. The aim here is to present an overview on some of the fundamental ideas and current developments leading to the research directions described in Section 1.3.

The organization of this chapter is as follows- basic definition of model based FTC is first described. The main ideas of generating and evaluating residuals in Fault Detection and Diagnosis (FDD) are next presented. This is followed by a description on reconfigurable control techniques. A discussion on the literature survey leading to the research directions taken in this work is then presented. Lastly, the chapter's conclusions are described.

2.2 Model Based Fault Tolerant Control Overview

The review starts off with a definition of fault, failure, Fault Tolerant Control and its components. According to the IFAC Technical Committee SAFEPROCESS definitions [10],

"A fault is an unpermitted deviation of at least one characteristic property or parameter of the system from the acceptable/ usual/ standard condition."

"A failure is a permanent interruption of a system's ability to perform a required function under specified operating conditions."

Critical faults may not cause system failures if they are handled/ mitigated. The purpose of Fault Tolerant Control (FTC) is to prevent a fault from deteriorating to a system failure. Emphasis and research on FTC has always been ongoing due to the potential realisable benefits, especially for safety critical systems.

Generally, faults can be grouped into three types [11]:

- Sensor
- Plant
- Actuator

When sensor faults occur, the plant is not changed. However, the sensor values feedback to the system for control and monitoring will be abnormal. Typical sensor faults include biases, drifts, loss of accuracy, and freezing of value. When these faulty readings are not detected and used for close loop control, degradation in control performance and stability will occur.

Plant faults change the dynamics of the system. An example of plant fault for aircraft is structural damage. In this case, aerodynamic properties of the aircraft or centre of gravity may be changed, causing the dynamics of the system to be changed. [36] presents a comprehensive review on flight vehicle structural damage monitoring and its challenges. Among the various techniques, Integrated Vehicle Health Monitoring (IVHM) techniques have surfaced as an approach to improve on the overall reliability of the system by predicting the emergence of faults before they become critical [37].

Actuator faults change the influence the controller has on the plant as they directly impacts the controllability and performance of the plant. Output actuator faults can be categorised as: stuck, floating, and hardover. Actuators can be stuck due to mechanical jams. They can also fail floating, in which the control surface moves freely according to subjected aerodynamic forces. Loss of power to an actuator may cause this type of fault. The worst actuator failure is considered the runaway or hardover fault. In a runaway fault, the control surface will move towards its maximum mechanical limit position or aerodynamic blowdown limit.

Besides the output actuators faults which are commonly analysed, common problems faced in the actuators are that they may be saturated due to their physical constraints such as voltage, current, or slew rate limit. These problems can result in significant performance degradation of the flight control system, and in the worst case, destabilise the entire system [38]. The design of control systems with constraints on input and state/output variables is currently an active research topic [4]. In the later section, the handling of actuator faults is described.

2.2.1 Fault Tolerant Control Objective

The aim of FTC is to prevent a fault from deteriorating to become a system failure. For flight systems, this also means the protection of the flying platform to ensure it remains within the safe operating condition after fault occurrence. Fig. 2.1 illustrates the concept of FTC enlarging the controllable flight envelope after fault occurrence. The primary task of FTC is to stabilize the system and bring the platform to within a safe controllable operating point post fault. Compared to a nominal control design which does not consider faults, FTC offers a safer operation by initiating recovery. The system's performance may have degraded with faults, but safety is improved.

2.2.2 Redundancy is Key to Fault Tolerant Control

A fault tolerant system requires redundancy in the system to recover from any possible critical fault. Redundancy can come in two main forms- hardware and analytical. The hardware redundancy concept is simple. The idea is to have duplicate sets of subsystems check against each other while remaining functional

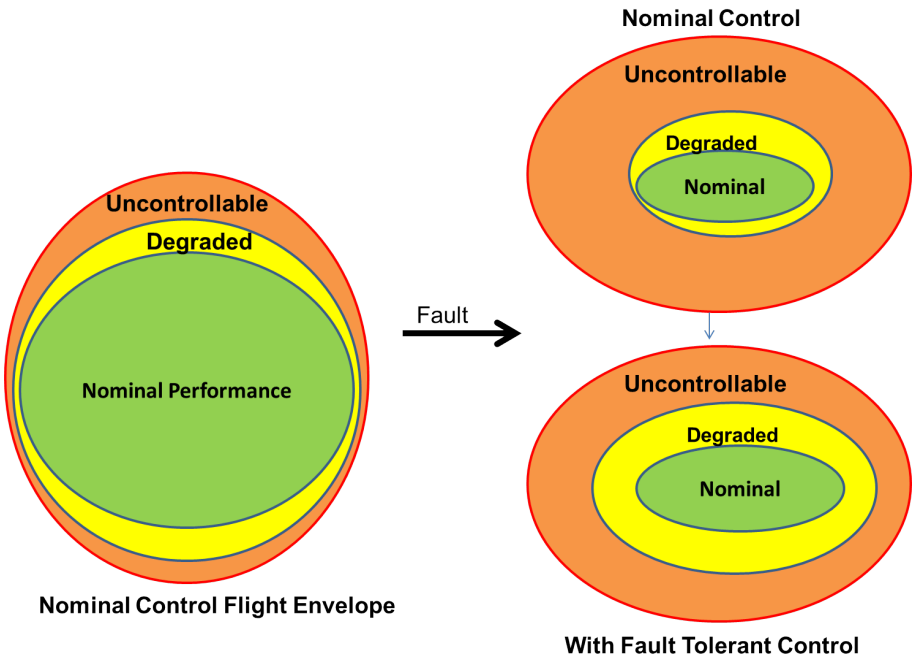

Figure 2.1: Flight Envelope Protection using Fault Tolerant Control

should one subsystem fail. For systems having triple or more hardware redundancies, the subsystems output are compared or voted against each another to select the value for propagation to the next subsystem, as well as to determine faulty subsystems. This is used in almost all big modern civil aircraft due to the need to achieve the required low probabilities of failure levels; examples include the Boeing 777, Airbus A320/330/340/380.

In smaller aircraft or large UAVs like the Predator UAV, as the required Mean Time between Lost (MTBL) probabilities are reduced, dual redundancies may be used to save weight and complexity. In dual redundant avionics systems, one subsystem is made the Master subsystem while the other acts as a backup subsystem (Slave) monitoring the Master. Comprehensive internal and cross subsystem health monitoring is done to sieve out potential faults for switching over.

However, mini/micro UAVs usually cannot afford the luxury of carrying redundant hardware due to weight or cost constraints. Instead, analytical redundancy may be utilized. The concept of analytical redundancy comes from the utilization of coupling connections often found between measurements/ outputs. A mathematical model of the system is used to generate estimates of other mea-

surements as they are inter-related mathematically.

For actuators, coupled interactions between their control variables can allow for controllability in secondary channels. An example would be the usage of rudder to roll the aircraft when ailerons fail. Besides saving weight, analytical redundancy also offers a form of dissimilar redundancy, which is important to avoid common mode failures [39].

2.3 Fault Tolerant Control Types

2.3.1 Passive Fault Tolerant Control

Historically, FTC systems can be classified into two types- passive and active. In passive FTC systems, a fixed controller is designed to tolerate changes of the plant dynamics which changes with the introduction of pre-determined faults. Typically, robust control formulations like H_2, H_∞ and robust eigenstructure assignment methods are used here [40, 41, 42]. During the design process, the faults and their effects on plant dynamics are considered to create a robust controller which will meet the necessary stabilization and performance requirements. No other form of diagnostics or detection methods is required, which also makes this form of design be known as 'reliable' control. While robust to the evaluated fault conditions, the problem with utilizing only passive fault tolerance approach is that robust controllers exist only for certain restricted class of changes caused by faults. This approach is potentially suboptimal for the nominal operating conditions due to the trade-off needed between performance and robustness to faults [43].

2.3.2 Active Fault Tolerant Control

Fig. 2.2 shows an active FTC system architecture. In model based active FTC approaches, a fault detection and diagnosis (FDD) module is first used to actively detect, isolate and identify a fault via the concept of residual generation and evaluation. Upon fault occurrence, a reconfiguration mechanism is used to handle the fault. A nominal mission, guidance and controller will be reconfigured by

the reconfiguration mechanism when faults are detected and diagnosed by the fault detection and diagnosis module. The reconfiguration mechanism may be embedded within the mission, guidance and control algorithms, instead of being a standalone module.

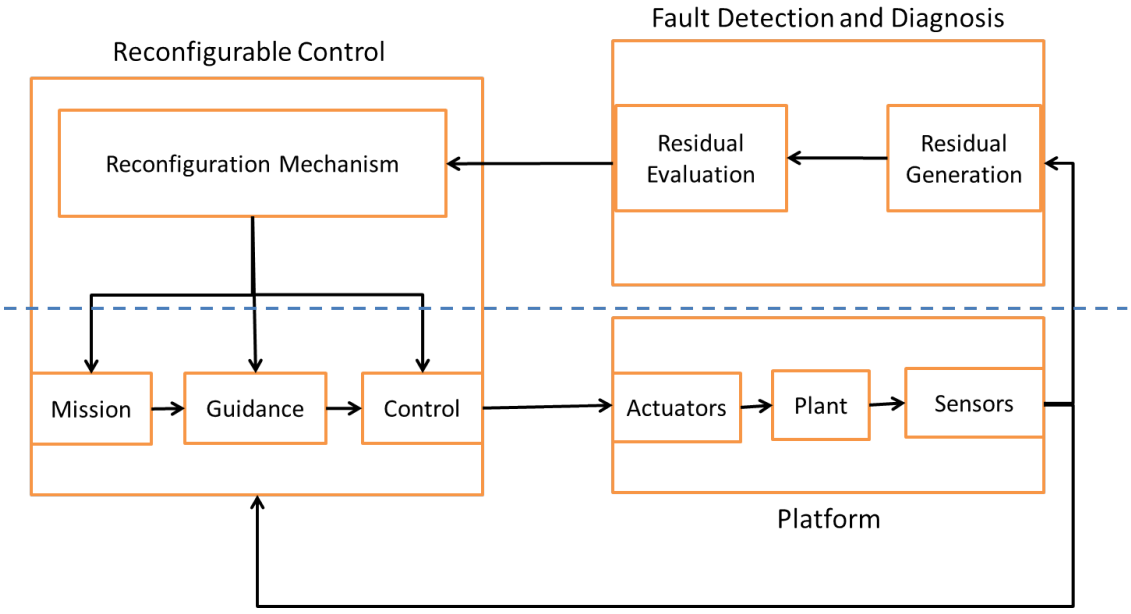

Figure 2.2: Active Fault Tolerant Control System block diagram

Some literature termed this whole process as Fault Detection Isolation Reconfiguration (FDIR)[32] [29], while others referred to it as Fault Detection Diagnosis (FDD) [44] [45] [46] [4].

In an active FTC architecture, the first task is to detect and identify faults. Fig. 2.3 illustrates the main idea in model based approaches for fault detection, consisting of two steps: residual generation and residual evaluation. In residual generation, a model is used to generate the expected output of a fault free plant. The model's output is compared (subtracted) with the actual plant output to generate a residual signal. Next, the residual signal is evaluated to determine the presence of faults with a residual evaluation algorithm.

As the FDD process takes place in the presence of disturbances and process noise, it may be difficult to distinguish between them and faulty behaviour as the plant dynamics may not be 'exciting' for identification. To improve on the fault detection and isolation process, a recent FDD approach is for the algorithm to actively inject probing signals into the null space to aid in the detection and

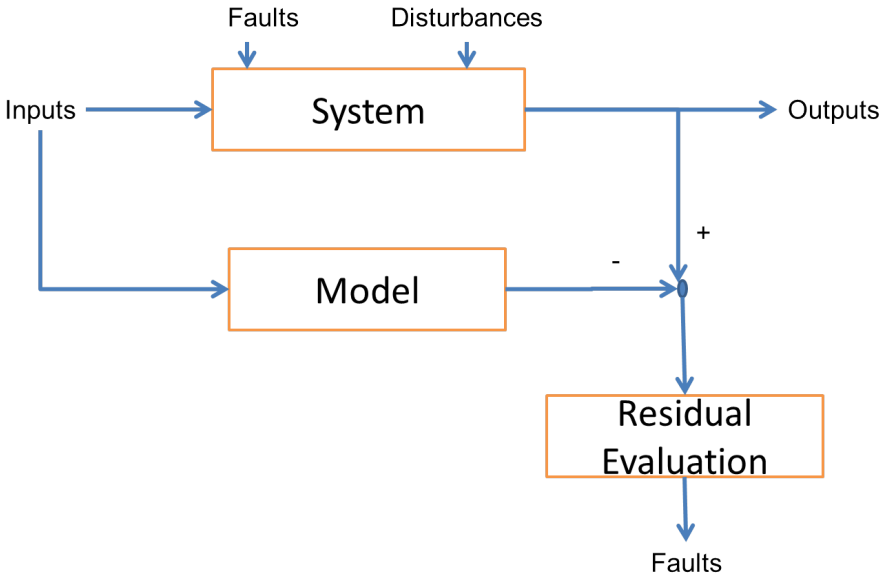

Figure 2.3: Model Based Fault Diagnosis

isolation process. This is known as the active FDD approach, which contrasts with previous FDD passive approaches that do not inject/ create additional dynamics into the plant. However, the active FDD approach adds additional complexity into the system.

Although the active FTC approach promises potentially greater fault detection and handling capabilities when compared to the passive robust FTC approach, the usage of residual generation and evaluation algorithms lead to the possibility of false alarms. On the other hand, a fault detection module may miss detections when faults are present, which may cause system failure.

Generally, a fault detection module will need to achieve high fault detection probability, fast detection time and low false alarm probability. A reliable fault detection algorithm is one with high fault detection probability. A fast detection time is required to ensure there is a good chance of recovering from the fault before it propagates to become system failure, while low false alarms rates increase mission reliability and safety if reconfiguration upon false alarm causes instability. One way to reduce false alarms is to have higher residual thresholds or longer detection confirmation times, which may result in missed detections and longer detection time leading to reduced chances of successful reconfiguration. For any practical implementation, these considerations need to be balanced to ensure

success [43].

For historical reasons and the level of complexity of research, FDD techniques and reconfigurable control are usually cast as two separate research topics. The sections below describe some of the FDD and reconfigurable control techniques.

2.4 Fault Detection and Diagnosis Methods

As mentioned in the earlier section, model based fault detection and isolation methods consist of residual generation and evaluation. There are generally four types of model based residual generation methods [4]:

- Observer based
- Parity Relationship
- Parameter Estimation
- Combination of the above

2.4.1 Residual Generation

2.4.1.1 Observer Based

The idea in observer designs is to use an output error between a measured process output and a model's output to construct a residual. It is assumed with this approach that the model's structure and parameters of the model are precisely known. A model of a linear time invariant process is

$$\dot{\mathbf{x}}(t) = \mathbf{A}\dot{\mathbf{x}}(t) + \mathbf{B}\mathbf{u}(t) \tag{2.1}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \tag{2.2}$$

The state observer's output is then

$$\hat{\dot{\mathbf{x}}}(t) = \mathbf{A}\hat{\dot{\mathbf{x}}}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{H}\mathbf{e}(t) \tag{2.3}$$

$$\mathbf{e}(t) = \mathbf{y}(t) - \mathbf{C}\hat{\mathbf{x}}(t) \tag{2.4}$$

where $\mathbf{e}(t)$ is an output error which acts through the observer gain matrix $\mathbf{H}$ to estimate the states. With the assumption that the system is observable, and stable through the proper design of the $\mathbf{H}$, it can be shown the state error vanishes asymptotically. One common method to design $\mathbf{H}$ is via pole placement. The state estimation poles should be faster than the process poles (high gain), and yet sufficiently slow to manage noise (low gain). The scheme is good for detecting additive faults. However, it is not straight forward to use this scheme for detecting multiplicative faults that changes the process dynamics $\mathbf{A}, \mathbf{B}, \mathbf{C}$, which are elaborated upon in Ref. [10].

Besides managing noise and observer stability, $\mathbf{H}$ can also be chosen to ensure the residuals have a proper structure so as to be sensitive to certain faults for fault isolation purposes [47]. However, it may be difficult to design for complete decoupling structure at times when not all measurements are available.

Besides classical Luenberger observer described above, Kalman filters and their variants are popularly used as a type of optimal observers, where $\mathbf{H}$ is chosen to minimize the effects of stationary stochastic white noise present in the system, with knowledge of their noise covariance. Besides Kalman filters, Sliding mode observers [48] have also been used.

Another method is to simply have a bank of special observers, each with different inputs and outputs. The observer with the omitted inputs or outputs will not show changes to the corresponding input and output faults. When the bank of observers includes all inputs and all outputs, except one input or one output which the fault is modelled for, it is called the generalized observer scheme [13].

The idea of using multiple models in fault detection and isolation has many different variants. Multiple Model Adaptive Estimation (MMAE)[49], Interacting Multiple Models [50], Multiple Model Switching and Tuning (MMST) [51]. The main differences between these formulations lie in the combinational logic for fault isolation and handling.

2.4.1.2 Parity Relations

One method to detect faults for model based fault detection is to compare the difference between the measured signals and the model outputs. The difference

or discrepancy between the two are expressed as residuals. Here, analytical redundancy is characterized in terms of a parity space [52]. One simple method of understanding parity relations is as follows: Consider a state space model of the form

$$\dot{\mathbf{x}}(t) = \mathbf{A}\dot{\mathbf{x}}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{V}\mathbf{v}(t) + \mathbf{L}\mathbf{f}(t) \tag{2.5}$$

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{N}\mathbf{n}(t) + \mathbf{M}\mathbf{f}(t) \tag{2.6}$$

where $\mathbf{n}(t)$ is noise disturbance, and $\mathbf{v}(t)$ is unmeasurable inputs or disturbances. $\mathbf{f}(t)$ describes the fault behaviour, and $\mathbf{N}$, $\mathbf{V}$, $\mathbf{L}$ and $\mathbf{M}$, are their corresponding matrix gains. By differentiating Equation 2.6 repeatedly q times, $q \leq n$, n being the system order leads to

$$\mathbf{Y}(t) = \mathbf{T}x(t) + \mathbf{Q_u}\mathbf{U}(t) \tag{2.7}$$

where

$$\mathbf{Y}(t) = \begin{bmatrix} y(t) \\ \dot{y}(t) \\ \ddot{y}(t) \\ \vdots \\ y^q(t) \end{bmatrix} \tag{2.8}$$

$$\mathbf{U}(t) = \begin{bmatrix} u(t) \\ \dot{u}(t) \\ \ddot{u}(t) \\ \vdots \\ u^q(t) \end{bmatrix} \tag{2.9}$$

$$\mathbf{T} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \mathbf{CA^2} \\ \vdots \\ \mathbf{CA^q} \end{bmatrix} \tag{2.10}$$

$$\mathbf{Q_u} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ \mathbf{CB} & 0 & 0 & \dots & 0 \\ \mathbf{CAB} & \mathbf{CB} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{CA^{q-1}B} & \mathbf{CA^{q-2}B} & \dots & \mathbf{CB} & 0 \end{bmatrix} \tag{2.11}$$

$$\mathbf{Q_v} = \begin{bmatrix} \mathbf{N} & 0 & 0 & \dots & 0 \\ \mathbf{CV} & \mathbf{N} & 0 & \dots & 0 \\ \mathbf{CAV} & \mathbf{CV} & \mathbf{N} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{CA^{q-1}V} & \mathbf{CA^{q-2}V} & \dots & \mathbf{CV} & \mathbf{N} \end{bmatrix} \tag{2.12}$$

$$\mathbf{Q_f} = \begin{bmatrix} M & 0 & 0 & \dots & 0 \\ \mathbf{CL} & 0 & 0 & \dots & 0 \\ \mathbf{CAB} & \mathbf{CB} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{CA^{q-1}} & \mathbf{CA^{q-2}B} & \dots & \mathbf{CB} & 0 \end{bmatrix} \tag{2.13}$$

Using this method, one can see the temporal redundancy in the equations generated at the same instant time t [53] . Multiplying weighting w^T to both sides of Equation 2.5 and selecting w^T such that

$$\mathbf{w^T T} = \mathbf{0} \tag{2.14}$$

$$\mathbf{w^T Q_v} = \mathbf{0} \tag{2.15}$$

The residual vector can be calculated as

$$\mathbf{r}(t) = \mathbf{w^T Y}(t) - \mathbf{w^T Q_u U}(t) \tag{2.16}$$

$$\tag{2.17}$$

Alternatively, the residual can be seen as

$$r(t) = \mathbf{w^T Q_f F}(t) + \mathbf{w^T Q_n N}(t) \tag{2.18}$$

Equation 2.18 shows how the residual is affected by the faults and noise. By choosing different weights satisfying Equation 2.14, a residual vector is formed to create a parity space [54]. The residuals may be enhanced by designing directional or structural residuals, in which the weights are chosen such that different faults trigger different residual patterns. This increases the robustness of the fault detection and isolation process greatly compared to using a single residual value.

However, the higher order derivatives of $y(t)$ are needed for realisation, which cannot be directly obtained due to practical noise issues. State observers may be used for continuous time implementations. Modern digital implementations are easier, since the q operator can be shifted backwards to be causal [10].

2.4.1.3 Parameter Estimation

Parameter Estimation techniques are popularly used in the aerospace industry for flight testing and verification purposes. The parameter estimation algorithms used can be classified into on-line and off-line algorithms, with the off-line maximum likelihood based algorithms being most popularly used for estimation of aircraft coefficients [55]. However, only real time on-line algorithms can be used in the context of fault detection and isolation.

On-line parameter estimation algorithms can be formulated in either time or frequency domain. For time domain techniques, Least Squares type of estimation algorithms like recursive least squares (RLS) [56] have been used. Other examples on this include Kalman Filter types like Extended and Unscented Kalman Filters [57].

One problem with parameter estimation methods used in fault detection is that they are often slow to update the abrupt changes cause by faults due to the practical noise considerations. Exponential forgetting is introduced as one way in recursive least squares to speed up the parameter estimation process.

2.4.1.4 Combinations of Methods

As each of the above described methods come with their own strengths and weakness in methodology, it is common for researchers to combine their strengths during application to enhance the fault detection and isolation capabilities during application. Some examples include [58] and [10].

2.4.2 Residual Evaluation

After a residual is generated, the residuals need to be evaluated for decision on fault occurrence. Residual evaluation methods may be statistical based like Chi-squared testing, Sequential probability ratio testing (SPRT), or generalized likelihood ratio (GLR) testing, to adaptive threshold methods[10]. The stochastic based formulations provide a theoretical foundation for trade off between false alarms and missed detection rates.

Multiple model techniques like Multiple Model Adaptive Estimation (MMAE) [9], Interacting Multiple Models (IMM) [50] [59], and Multiple Model Switching and Tuning (MMST) [60] [61] have also been proposed.

2.5 Reconfigurable Control Methods

The next step after fault confirmation is to reconfigure the controller to handle the fault. Many techniques exist to reconfigure the controller. Unlike model based FDI which can be classified into four main types, almost any control design technique may be used to synthesize a controller, once given the post fault plant dynamics.

Some examples include dynamic inversion [62], adaptive control [63], [64], LMI [7], sliding model control [65],and backstepping [66], just to name a few. Fig. 2.4 shows a possible classification of some of the reconfiguration techniques around.

Different reconfiguration methods have also been studied in Ref. [67]. In the reference, various techniques have been applied on a recreated incident on Boeing 747 Flight 1862. This allows would-be-practitioners a chance to compare and contrast between the various techniques out there. Besides controller re-design, another popular method of handling the fault is to reconfigure the control allocator for fault handling.

2.5.1 Control Allocation

Control allocation is about the assignment of virtual control demands to the physical actuator commands. In aircraft designs, it may be more intuitive to design a controller with outputs in the roll, pitch, and yaw control space than physical actuator space. The task of control allocator is then to distribute these demands to the physical actuators. This is similar to the mechanical linkages in older aircraft, where the actuators are ganged together mechanically to the pilot stick controls. In the digital world, the control allocation method can use this distribution as a means to handle actuator faults easily.

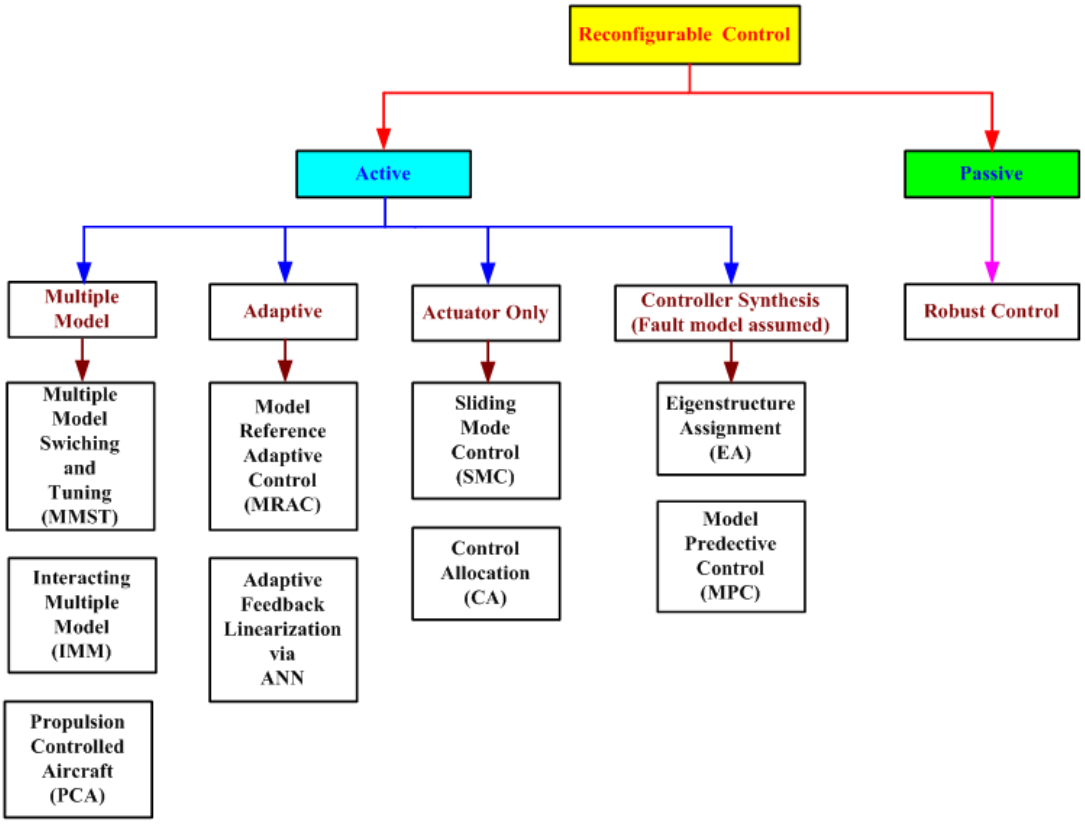

Figure 2.4: Some Reconfigurable control methods

Good overviews on control allocation can be found in [68],[69],[70], and [71]. In Ref. [72], besides fulfilling the mapping demands, control allocation can at the same time be used to optimize the actuator outputs according to various mission profile requirements. Ref. [73] has established an equivalence relationship in using LQR and optimal control allocation. Many of these techniques are real-time optimization based approaches to search for a feasible solution, and a comprehensive review on these techniques is given in Ref. [74].

While optimization based approaches for control allocation may approach the reach of current real time computational implementation capabilities, small Unmanned Aerial Vehicles (UAVs) which may typically be more computationally constrained may still benefit from efficient alleviation techniques. In Ref. [75], a computationally efficient control allocation has been developed based on an explicit set of allocation rules for small UAVs. This is an alternative to optimization based control allocation from aerodynamic considerations. In Ref. [76], weak and strong input redundancy of control systems have been defined, and linear in-line

filters are used to exploit the input redundancy space to alleviate input rate and magnitude saturations. Together, these are some of the available techniques for computationally constrained platforms.

2.5.2 Fault Tolerance Guidance

While much of the past focus of FTC has been on the fault handling using the flight controller, one emerging research area is the usage of guidance schemes for fault accommodation. The concept here is to change the guidance demands to suit the resulting degraded system's capabilities. This may be especially important at the edges of controller reconfiguration capability, as fault tolerant guidance may complement it to improve the overall mission safety and performance with fault occurrence [29]. Recent research using this approach includes [77], [78] and [79].

In path following missions, the guidance scheme usually consists of a path planner and path following modules. While many research works have used either the plan planner or the path following modules individually for fault reconfiguration, one idea which is explored in here to improve the reconfiguration process is the usage of both modules concurrently for extremal faults.

2.6 Discussion

The literature survey on FTC revealed a well-established field in which extensive work has already been done in the area. A brief overview on FDD has been presented, which shows the process may be divided into the tasks of residual generation and residual evaluation. Analytical redundancy may be exploited to generate residuals to save hardware sensor requirements. However, many practical applications still use physical sensors for diagnosis due to the reliability of the solution obtained. The advancement in sensors minimization technology makes the implementation of hardware redundancy in flight sensors easier for existing UAVs looking to improve on its fault tolerance. This is also the same for structural sensors which are increasingly being used for fault diagnosis purposes.

However, the mechanical parts of the actuation system are relatively more difficult to duplicate. Thus, the author's inclination is to research on the handling

of actuator faults. Fault detection and diagnosis is assumed in this research to be given by sensor hardware which is reliable. These result in a time delay block being used to model this block.

The literature review has provided inspiration for the author to direct the research towards the improvement of handling extremal actuator faults. Speed is crucial to the success in fault handling, as the system is moving towards failure with fault occurrence. In typical flight control systems, the actuators are the innermost and fastest control loop affecting the system dynamics. If actuator faults are handled within the actuators system itself, it would be potentially faster relative to using outer control loops for fault handling. As the research trend towards exploring integrated approaches, the work here explores the possibility of alleviating the fault performance through integration with control allocation at the actuator level.

Lastly, the literature survey has identified fault tolerant guidance as an emerging research direction. One potential advantage here is the ability to accommodate the fault degradation while protecting the mission. Many research works have focused on the reconfiguration of either path planning or path following algorithms. Following up on the integrated approach idea, this work explores the possible usage of the two modules of guidance for reconfiguration to provide better fault handling capabilities compared to using either one alone.

Together, the ideas derived from the review inspire the research approaches taken in this work to improve on the ability of systems to handle extremal actuator faults.

2.7 Conclusion

A brief literature review on FTC is presented as background for the research. An overview of FTC explaining its primary task of ensuring the flying platform remains within its operating envelope is described. The need for redundancy as the main requirement for FTC to work is elaborated. Passive and active FTC as the two main types of FTC are described, with active FTC having a fault detection isolation module and reconfigurable control for realization. FDD and reconfigurable control techniques are presented. Control allocation is introduced as a possible

reconfiguration technique to handle actuator faults. An emerging research area is on the guidance being utilized to accommodate the fault degradation. A possible advantage of using fault tolerant guidance is the ability to consider and improve on the overall mission safety and performance. A discussion based the insights from the literature survey reveals the research directions taken in this work to further the research on the handling extremal actuator faults on UAVs.

Chapter 3

Improving the performance of an actuation system when subjected to saturation fault

3.1 Introduction

The performance of the actuation system plays a decisive role in determining the performance of the flight control system, especially for a highly manoeuvrable air vehicles [30]. The vehicles are generally controlled by fins of which defection produces aerodynamic force. However, in classical autopilot design, the autopilot produces virtual roll, pitch and yaw moment demands [80] instead of physical fin deflection commands. These moment demands are then 'mixed'or allocated by an control allocation algorithm to generate individual actuator commands. The actuator commands will typically be tracked by their individual actuator controllers. Common problems faced in the actuators are that they may be saturated due to their physical constraints such as voltage, current, or slew rate limit. These problems can result in the significant performance degradation of the flight control system, or worse still, destabilize the entire system.

Many approaches have been proposed to tackle this problem. One method to handling actuator fault for an over-actuated system is to treat it as a control allocation problem. Methods such as redistributed pesudo inverse [68], dynamic control allocation [81], quadratic programming techniques [82], and direct allocation [83] have been devised as optimization based approaches to handle the constraints. Fault handling and reconfiguration control during actuator faults can also be considered [84], [85]. Many of these techniques are real-time optimization based approaches to search for a feasible solution, and a comprehensive review on these techniques is given in Ref. [74].

While optimization based approaches for control allocation may approach the reach of current real time computational implementation capabilities, small

Unmanned Aerial Vehicles (UAVs) which may typically be more computationally constrained may still benefit from efficient alleviation techniques. In Ref. [75], a computationally efficient control allocation has been developed based on an explicit set of allocation rule for small UAVs as an alternative to optimization based control allocation from aerodynamic considerations. In Ref. [76], weak and strong input redundancy of control systems have been defined, and linear in-line filters are used to exploit the input redundancy space to alleviate input rate and magnitude saturations have been proposed. Together, these are some of the possible available techniques for computationally constrained platforms.

Increasingly, trends to treat the actuation system control design as a whole system approach have become increasingly important. Concepts such as novel actuator designs like hierarchical actuators consisting of integrated actuators from smaller ones [19], plasma actuators [20], and MEMs based actuators [21] have surfaced. Some of these actuators are already applied in flight control of UAVs [22], [23]. Due to the different physical architectures, control strategies may need to be tweaked to exploit the new actuator forms and the analytical redundancy within.

In an integrated approach, Ref. [30] proposed an alternative actuator control scheme which alleviates the actuator saturation/ faults for a four tail fin controlled missile. The integrated actuator control scheme regulates the error in the autopilot demand space instead of actuator space to utilize the analytically redundant actuator. The paper presented simulation results showing the improved performance compared to the conventional actuator control scheme during actuator saturation.

This chapter aims to extend on the work in Ref. [30] using frequency domain analysis methods. First, the integrated actuator control scheme is shown to be equivalent to a conventional scheme in unsaturated operation region. Next, the superior performance of the integrated scheme is shown for one and two actuator saturation situations with available actuator redundancy and controllability. Simulation results verifying the analysis are then shown.

3.2 Description of Control Scheme

Fig. 4.1 shows a typical control architecture of an aerial vehicle. The flight control computer computes the guidance and flight control demands using the received flight sensors data. Usually, when the flight control demand outputs may be more intuitive to be synthesized in the moment virtual demand commands δ_{mc} . When there are more actuators than virtual demands, control allocation may be used to synthesize or distribute the virtual demands into individual actuator commands δ.

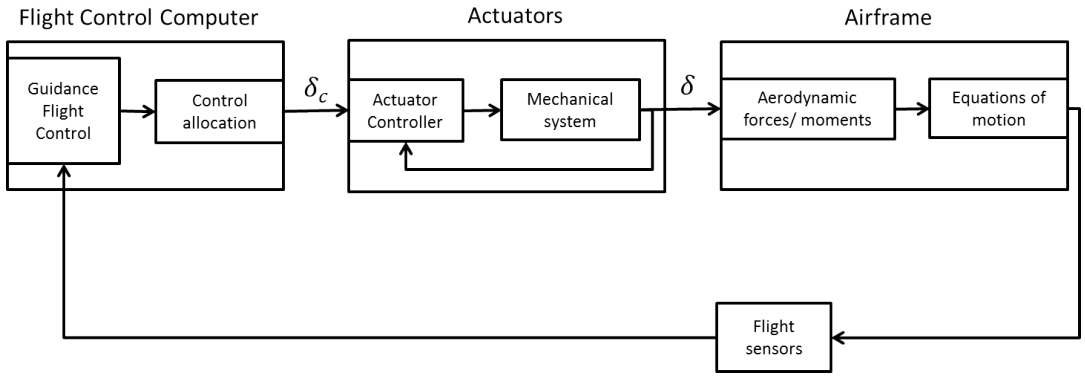

Figure 3.1: A typical vehicle autopilot scheme with the actuation system block.

3.2.1 Conventional Actuator Control Scheme

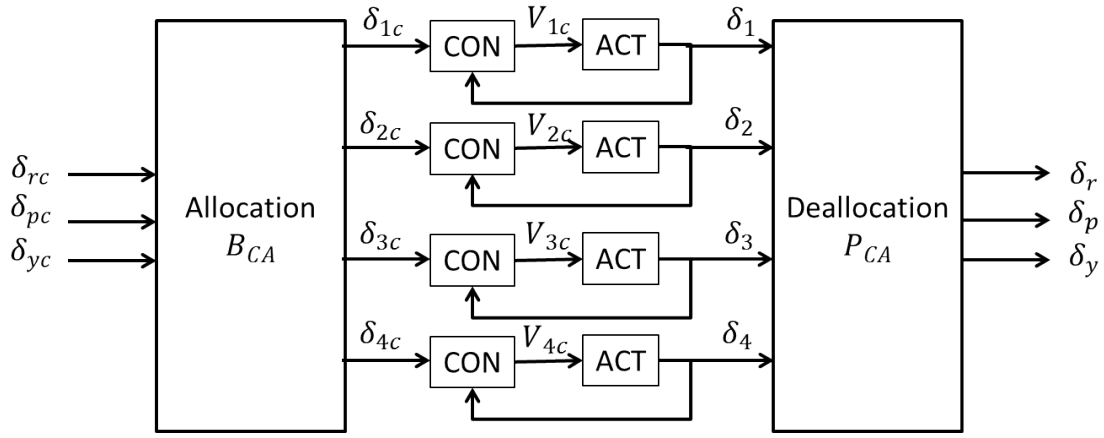

Figure 3.2: Conventional Actuator Control Architecture

Here, the signal flow between the control allocation and the actuation system is elaborated. Fig. 3.2 shows a conventional actuator control architecture for a four tail fin controlled UAV. First, an autopilot generates a virtual command $\delta_{mc} = \begin{bmatrix} \delta_{pc} & \delta_{rc} & \delta_{yc} \end{bmatrix}^T$ in the roll, pitch and yaw moment space to the actuators. The virtual command is allocated or mixed by $\mathbf{B_{ca}}$ to generate each individual actuator command δ_{ic}. Each actuator controller K_i computes the voltage command V_{ic} to their actuator plant. Each actuator plant considers of a linear actuator model G_i, which is preceded by a physical voltage constraint K_{is}. The resulting actuator positions are measured, and the achieved autopilot demand response can be found by deallocating the individual actuator response by matrix $\mathbf{P_{ca}}$.

The closed loop transfer function for the conventional actuator control scheme for the autopilot demands is

$$\delta_{\mathbf{m}} = \mathbf{P_{ca}GK}\left(\mathbf{I} + \mathbf{GK}\right)^{-1}\mathbf{B_{ca}}\delta_{\mathbf{mc}} \tag{3.1}$$

The four actuators can be combined mathematically as $\mathbf{G} = diag(G_i)$ and $\mathbf{K_s} = diag(K_{is})$, with i's values being 1 to 4. K_{is} is the standard saturation function to describe the physical voltage input constraint, V_c. In frequency domain, the nonlinear saturation function can be represented as

$$K_{is} = 1, \qquad |V_c| \leq V_{max} \tag{3.2}$$

$$0 < K_{is} < 1, \qquad |V_c| > V_{max} \tag{3.3}$$

It is assumed there is no unstable pole zero cancellation in the system. For a four tail fin UAV under consideration here, the control allocation matrix $\mathbf{B_{ca}}$ is

$$\mathbf{B_{ca}} = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & -1 & -1 \end{bmatrix} \tag{3.4}$$

and the deallocation matrix P_{ca} being

$$\delta_{\mathbf{m}} = \mathbf{P_{ca}}\,\delta$$

$$\begin{bmatrix} \delta_r \\ \delta_p \\ \delta_y \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 \end{bmatrix}\begin{bmatrix} \delta_1 \\ \delta_2 \\ \delta_3 \\ \delta_4 \end{bmatrix} \tag{3.5}$$

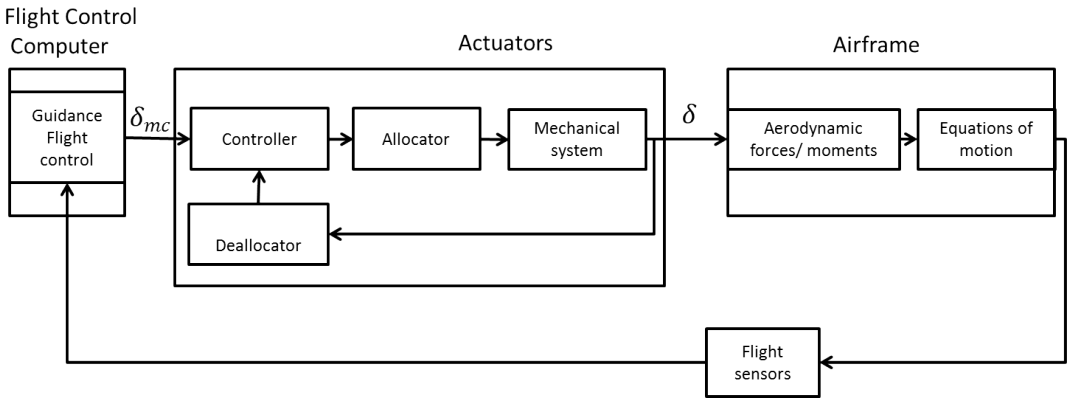

Figure 3.3: Integrated Actuator Control Architecture

with $\mathbf{P_{ca}B_{ca}} = \mathbf{I}$, or $\mathbf{B_{ca}}$ being the pseudo inverse solution of $\mathbf{P_{ca}}$. The control deallocation matrix is obtained by considering the resultant torque generated by each actuator's position from the aerodynamics point of view.

3.2.2 Integrated Actuator Control Scheme

In the integrated actuator control scheme, the idea is to shift the allocator from the flight computer block to the actuation block to combine the actuators as shown in Fig. 3.3. Fig. 3.4 elaborates on the integrated actuator control scheme first proposed in [30]. Here, the controller regulates in the autopilot demand space before control allocation. This contrasts with the conventional control scheme regulating in each individual actuator command. The closed loop transfer function of the integrated scheme can be derived as

$$\delta_{\mathbf{m}}(s) = \mathbf{P_{ca}GK_sB_{ca}K}\left(\mathbf{I} + \mathbf{P_{ca}GK_sB_{ca}K}\right)^{-1}\delta_{\mathbf{mc}}(s) \tag{3.6}$$

3.2.3 Comparison of Control Schemes

The comparison of the two control schemes is divided into 3 cases: linear (non-saturating) case, 1 actuator saturation fault case, and > 1 actuator saturation fault case.

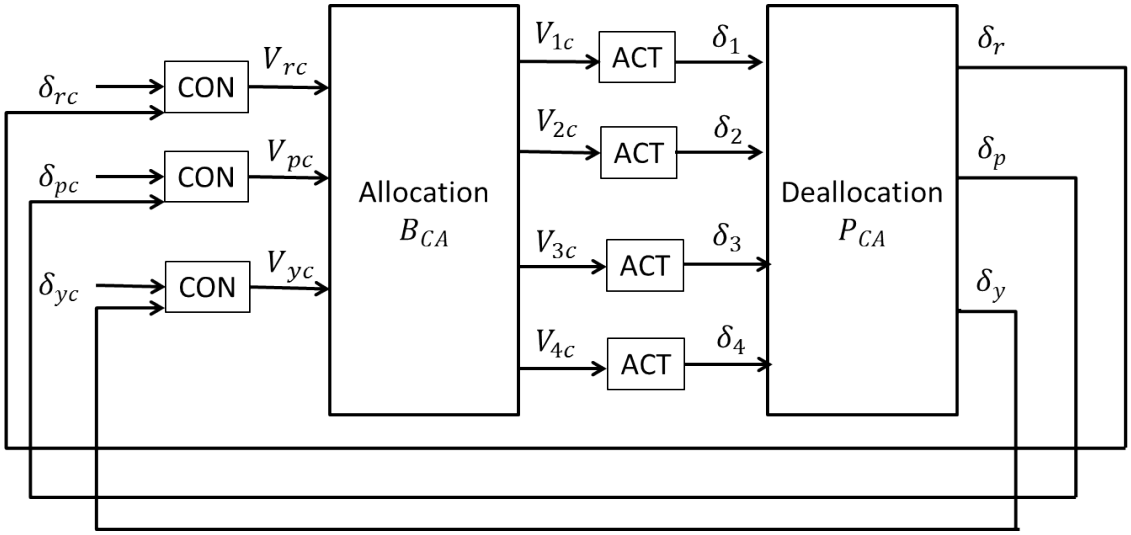

Figure 3.4: Integrated actuator control scheme

3.2.3.1 Non saturation case

First, the actuators and their controllers are assumed similar, that is, $G_i = G$ and $K_i = K$. In the case when there is no saturation fault, K_{is} become unity gain in Equations 3.1 and 3.6, that is,

$$\mathbf{K_s} = \mathbf{I} \tag{3.7}$$

Substituting Eqn. 3.7 into Eqn. 3.1 results in

$$\begin{aligned}\delta_{\mathbf{m}}(\mathbf{s}) &= \mathbf{P_{ca}GK_sK}\left(\mathbf{I} + \mathbf{GK_sK}\right)^{-1}\mathbf{B_{ca}}\delta_{\mathbf{mc}}(\mathbf{s}) \\ &= \mathbf{P_{ca}GK}\left(\mathbf{I} + \mathbf{GK}\right)^{-1}\mathbf{B_{ca}}\delta_{\mathbf{mc}}(\mathbf{s})\end{aligned} \tag{3.8}$$

Realise $\mathbf{GK}$ and $(\mathbf{I} + \mathbf{GK})^{-1}$ are diagonal matrices. This means the inverse of the of the term $(\mathbf{I} + \mathbf{GK})^{-1}$ is simply the inverse of the individual diagonal terms. Using this information, and substituting Eqn. 3.4 and Eqn 4.52 into Eqn. 3.8,

$$\delta_{\mathbf{m}}(\mathbf{s}) = \frac{GK}{1+GK}\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}\delta_{\mathbf{mc}}(\mathbf{s}) \tag{3.9}$$

Similarly, the same approach is used to analyse the integrated scheme. Substituting Eqns. 3.7, 3.4 and 4.52 into Eqn 3.6, and noting $\mathbf{G}$, $\mathbf{K}$ are diagonal, Eqn. 3.6 becomes Eqn. 3.9.

The significance of this case is to show the two control schemes are equivalent during nominal operation, and prove the improvement in performance of the integrated scheme shown later is not due to increase in gain used. In practice, this may reduce the amount of design analysis needed to convert from the integrated scheme for any existing actuator scheme.

3.2.3.2 One actuator saturation fault case

Next, the analysis is extended to the case with 1 actuator saturation fault. First, assume actuator 4 is saturated. This can be represented by

$$\mathbf{K_s} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & K_{4s} \end{bmatrix} \tag{3.10}$$

Substituting Equation 3.10 into Equation 3.6 for the integrated scheme, and rearranging with Equation 4.52 reveals

$$\begin{aligned} \delta_r &= \left(\frac{GK}{1+GK}\right)\delta_{rc} + \left(\frac{K_{4s}-1}{K\,(G+3\,G\,K_{4s})+4}\right)\begin{bmatrix} 1 \\ -1 \\ -1 \end{bmatrix}^T \begin{bmatrix} \delta_{rc} \\ \delta_{pc} \\ \delta_{yc} \end{bmatrix} \\ &= \left(\frac{GK}{1+GK}\right)\delta_{rc} + \frac{1}{4}\left(\frac{K_{4s}-1}{K\,(G+3\,G\,K_{4s})+4}\right)\delta_{4c} \end{aligned} \tag{3.11}$$

$$\begin{aligned} \delta_p &= \left(\frac{GK}{1+GK}\right)\delta_{pc} + \left(\frac{K_{4s}-1}{K\,(G+3\,G\,K_{4s})+4}\right)\begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix}^T \begin{bmatrix} \delta_{rc} \\ \delta_{pc} \\ \delta_{yc} \end{bmatrix} \\ &= \left(\frac{GK}{1+GK}\right)\delta_{pc} - \frac{1}{4}\left(\frac{K_{4s}-1}{K\,(G+3\,G\,K_{4s})+4}\right)\delta_{4c} \end{aligned} \tag{3.12}$$

$$\begin{aligned} \delta_y &= \left(\frac{GK}{1+GK}\right)\delta_{yc} + \left(\frac{K_{4s}-1}{K\,(G+3\,G\,K_{4s})+4}\right)\begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix}^T \begin{bmatrix} \delta_{rc} \\ \delta_{pc} \\ \delta_{yc} \end{bmatrix} \\ &= \left(\frac{GK}{1+GK}\right)\delta_{yc} - \frac{1}{4}\left(\frac{K_{4s}-1}{K\,(G+3\,G\,K_{4s})+4}\right)\delta_{4c} \end{aligned} \tag{3.13}$$

Equations 3.11 to 3.13 show the effect of actuator saturation fault. The first term on the right hand side of these equations is the nominal unsaturated performance of the actuator, while the second term is the additional dynamics introduced by the saturated actuator. For the conventional scheme, similar relationships can be obtained by manipulating Equation 3.1 in the same manner to obtain:

$$\begin{aligned}\delta_r &= \left(\frac{GK}{1+GK}\right)\delta_{rc} + \left(\frac{K_{4s}-1}{4GKK_{4s}+4}\right)\begin{bmatrix}1\\-1\\-1\end{bmatrix}^T\begin{bmatrix}\delta_{rc}\\\delta_{pc}\\\delta_{yc}\end{bmatrix}\\ &= \left(\frac{GK}{1+GK}\right)\delta_{rc} + \frac{1}{4}\left(\frac{K_{4s}-1}{4GKK_{4s}+4}\right)\delta_{4c}\end{aligned} \tag{3.14}$$

$$\begin{aligned}\delta_p &= \left(\frac{GK}{1+GK}\right)\delta_{pc} + \left(\frac{K_{4s}-1}{4GKK_{4s}+4}\right)\begin{bmatrix}-1\\1\\1\end{bmatrix}^T\begin{bmatrix}\delta_{rc}\\\delta_{pc}\\\delta_{yc}\end{bmatrix}\\ &= \left(\frac{GK}{1+GK}\right)\delta_{pc} - \frac{1}{4}\left(\frac{K_{4s}-1}{4GKK_{4s}+4}\right)\delta_{4c}\end{aligned} \tag{3.15}$$

$$\begin{aligned}\delta_y &= \left(\frac{GK}{1+GK}\right)\delta_{yc} + \left(\frac{K_{4s}-1}{4GKK_{4s}+4}\right)\begin{bmatrix}-1\\1\\1\end{bmatrix}^T\begin{bmatrix}\delta_{rc}\\\delta_{pc}\\\delta_{yc}\end{bmatrix}\\ &= \left(\frac{GK}{1+GK}\right)\delta_{yc} - \frac{1}{4}\left(\frac{K_{4s}-1}{4GKK_{4s}+4}\right)\delta_{4c}\end{aligned} \tag{3.16}$$

Similarly, the Actuator 4 saturation fault effects can be accounted for the conventional actuator scheme. The second term on the right hand side of Equations 3.14 to 3.16 is the detrimental contribution by the saturation.

Comparing between the integrated and conventional schemes, it can be seen the numerator of the saturation dynamics for both schemes are the same at $(K_{4s}-1)$ from Equations 3.11 to 3.16.

Now, comparing the denominator of conventional and integrated scheme, it can be seen the conventional scheme's denominator is $4GKK_{4s}+4$, while the integrated scheme's denominator is $GK+3GKK_{4s}+4$. To compare between the two terms, let $G(s) = a + bj$.

For the conventional scheme, the magnitude of the denominator is

$$|4GKK_{4s}+4| = \left((4aKK_{4s}+4)^2 + (4bKK_{4s})^2\right)^{0.5} \tag{3.17}$$

In comparison, the integrated scheme's denominator is

$$|GK + 3GKK_{4s} + 4| = \left((aK + 3aKK_{4s} + 4)^2 + (bK + 3bKK_{4s})^2\right)^{0.5} \tag{3.18}$$

By comparing term by term between Eqn. 3.17 and Eqn. 3.18, and noting

$$|4K_{4s}| < |1 + 3K_{4s}| \tag{3.19}$$

as $0 < K_{4s} < 1$ during actuator saturation or fault.This implies the magnitude of the effect caused by the saturation term for the conventional scheme will be bigger than the integrated scheme. The alleviation in actuator saturation in the integrated scheme in this case is due to the availability of actuator redundancy, which enables the controller to increase the commands of the other non-saturated actuators.

Next, the saturations on the other actuators are considered. For the single actuator saturation case, Equations 3.11 to 3.13 can be generalised for the integrated scheme to be

$$\delta_{\mathbf{m}}(s) = \frac{GK}{1 + GK} \left(\mathbf{I} + \mathbf{P_{ca}} \mathbf{D} \mathbf{B_{ca}}\right) \delta_{\mathbf{mc}}(s) \tag{3.20}$$

where

$$\mathbf{D} = diag\left[4\left(\frac{K_{is} - 1}{K\,G + 3\,G\,K\,K_{is} + 4}\right)\right] \tag{3.21}$$

with the appropriate actuator being saturated, and the other K_{is} being equal to 1.

3.2.4 Two actuators saturation

The remaining two unsaturated actuators provide only two degrees of freedom for fulfilling three virtual commands. During saturation, the saturated actuators would effectively be open loop in performance. To map this to the control allocation matrix $\mathbf{B_{ca}}$ by making the row representing the saturated actuators zero, 6 possible combinations matrices are obtained:

$$\mathbf{B_{1,2}} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & -1 & 1 \\ 1 & -1 & -1 \end{bmatrix} \tag{3.22}$$

$$\mathbf{B_{1,3}} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \\ 1 & -1 & -1 \end{bmatrix} \tag{3.23}$$

$$\mathbf{B_{1,4}} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 1 & 1 \\ 1 & -1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.24}$$

$$\mathbf{B_{2,3}} = \begin{bmatrix} 1 & 1 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & -1 & -1 \end{bmatrix} \tag{3.25}$$

$$\mathbf{B_{2,4}} = \begin{bmatrix} 1 & 1 & -1 \\ 0 & 0 & 0 \\ 1 & -1 & 1 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.26}$$

$$\mathbf{B_{3,4}} = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.27}$$

This means only one of the three autopilot commands will be controllable, while the remaining two commands will have to "share" the remaining actuator. Without losing generality, assume the third and fourth actuator is saturated, that is

$$K_s = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & K_{3s} & 0 \\ 0 & 0 & 0 & K_{4s} \end{bmatrix} \tag{3.28}$$

The integrated scheme output is

$$\begin{bmatrix} \delta_r \\ \delta_p \\ \delta_y \end{bmatrix} = \frac{GK}{1+GK}\left(\begin{bmatrix} \delta_{rc} \\ \delta_{pc} \\ \delta_{yc} \end{bmatrix} + \frac{1}{\chi}\begin{bmatrix} \alpha & -\alpha & \beta \\ -\alpha & \alpha & -\beta \\ \beta & -\beta & \gamma \end{bmatrix}\begin{bmatrix} \delta_{rc} \\ \delta_{pc} \\ \delta_{yc} \end{bmatrix}\right) \tag{3.29}$$

where

$$\alpha = K_{3s} + K_{4s} + GK\left(K_{3s}K_{4s} - 1\right) - 2 \tag{3.30}$$

$$\beta = (GK+1)\left(K_{3s} - K_{4s}\right) \tag{3.31}$$

$$\gamma = K_{3s} + K_{4s} + GK\left(2K_{3s}K_{4s} - K_{4s} - K_{3s}\right) - 2 \tag{3.32}$$

$$\chi = G^2K^2\left(K_{3s} + K_{4s} + 2K_{3s}K_{4s}\right) + GK\left(1 + 3K_{3s}\right) + GK\left(1 + 3K_{4s}\right) + 4 \tag{3.33}$$

Similarly, the conventional scheme relationship can be worked out to be

$$\begin{bmatrix} \delta_r \\ \delta_p \\ \delta_y \end{bmatrix} = \frac{GK}{1+GK}\left(\begin{bmatrix} \delta_{rc} \\ \delta_{pc} \\ \delta_{yc} \end{bmatrix} + \frac{1}{\eta}\begin{bmatrix} \gamma & -\gamma & \beta \\ -\gamma & \gamma & -\beta \\ \beta & -\beta & \gamma \end{bmatrix}\begin{bmatrix} \delta_{rc} \\ \delta_{pc} \\ \delta_{yc} \end{bmatrix}\right) \tag{3.34}$$

where

$$\eta = 4G^2K^2K_{3s}K_{4s} + 4GKK_{3s} + 4GKK_{4s} + 4 \tag{3.35}$$

Equations 3.29 and 3.34 are now compared to assess the performance between the two schemes. In both equations, the first term is the nominal unsaturated performance, while the second term represent the saturation dynamics by the two actuators. Comparing the denominator of the second term χ and η, it can be deduced

$$\begin{aligned} \left(1 + 3K_{3s}\right) &> 4K_{3s} \\ \left(1 + 3K_{4s}\right) &> 4K_{4s} \\ \left(K_{3s} + K_{4s} + 2K_{3s}K_{4s}\right) &> 4K_{3s}K_{4s} \end{aligned} \tag{3.36}$$

with $K_{3s} < 1$ and $K_{4s} < 1$ during saturated. Let $G = a + bj$. The real and imaginary parts of χ and η can be expanded as

$$\begin{aligned} \chi =& (a^2 - b^2)K^2(K_{3s} + K_{4s} + 2K_{3s}K_{4s}) + aK(1 + 3K_{3s}) + aK(1 + 3K_{4s}) + 4 \\ &+ j\left(2abK^2(K^2(K_{3s} + K_{4s} + 2K_{3s}K_{4s}) + bK(1 + 3K_{3s}) + bK(1 + 3K_{4s})\right) \end{aligned} \tag{3.37}$$

$$\begin{aligned}\eta =&(a^2 - b^2)K^2(4K_{3s}K_{4s} + 4aKK_{3s} + 4aKK_{4s} + 4 \\ &+ j\left(2abK^2(4K_{3s}K_{4s}) + 4bKK_{3s} + 4bKK_{3s}\right)\end{aligned} \tag{3.38}$$

Assuming $a^2 > b^2$, and using Eqn. 3.36, the real and imaginary parts of Eqns. 3.37 and 3.38 can be compared in magnitude to obtain $\chi > \eta$, or

$$\left|\frac{1}{\chi} < \frac{1}{\eta}\right| \tag{3.39}$$

which means the integrated scheme has a bigger denominator than conventional scheme. This implies that the magnitude of the undesired saturation dynamics of the integrated scheme is smaller for the numerator terms, β, which are common across both schemes. Comparing these terms and Equation 3.27, it is seen the controllable virtual demand is the one which the integrated scheme will be better off than the conventional scheme. This is expected; one needs controllability to improve on the performance.

Next, the second term numerator in Equations 3.29 and 3.34 are compared for those terms that are different. Specifically, α from the integrated scheme is compared to the γ for the conventional scheme. Removing the common terms from both terms, and noting the relationship $|(K_{3s} - 1)(K_{4s-1})| < 1$ when $0 < K_{3s} < 1$ and $0 < K_{4s} < 1$ during saturation, results in

$$|\alpha| > |\gamma| \tag{3.40}$$

which shows the numerator of the integrated scheme to be bigger in magnitude than the conventional scheme. With a larger numerator and denominator for the integrated scheme, it is unclear which scheme is better. Referring to Equation 3.27, the channels which exhibit this characteristic are the ones without controllability. Thus, it is hard for any scheme to be superior to another without controllability.

The comparison for the two actuator saturation case shows the integrated scheme having reduced magnitude of saturation dynamics over conventional scheme for the distinct virtual demand with controllability, while unclear performance comparison results are obtained for the other two virtual demands. For the other 5 possible different combinations of two actuator saturations, similar results are obtained. In the next section, simulations are presented providing additional insights into the comparison.

3.3 Simulation

3.3.1 Simulation Model

The actuator model used here is from the actuator model described in [30]. A sinusoidal input at 0.3 rad/sec is used as input virtual command for all three virtual demands when conducting all the simulations for consistency.

3.3.2 Simulation Results

The nominal simulation results are first shown in Fig. 3.5. The two control schemes perform in the exact same manner as predicted in theoretical analysis presented earlier.

To simulate the effect of actuator fault, the maximum voltage input level for an actuator is reduced by 90 percent of its original value. This can also be thought of as the actuator having a fault to reduce its effectiveness.

Fig. 3.6 shows the simulation results with Actuator 2 voltage range reduced. The integrated scheme is able to track the virtual demand relatively well for all three virtual demands in Fig. 3.6. In contrast, the conventional scheme shown with green dotted line exhibits poor tracking consistent across the three virtual demands. Fig. 3.7 shows the individual actuator responses for the same simulation. It can be seen the integrated scheme is able to exploit the available actuator capability when actuator 2 is saturated to improve the performance over the conventional scheme. This also implies the need for the remaining healthy unsaturated actuators to have remaining actuator margins in physical constraints for the integrated scheme to exploit.

Next, Actuator 1 and 2 are clipped in voltage to induce voltage saturation with the same simulation profile, and the simulation results are shown in Figs. 3.8 and 3.7. It can be seen from Fig. 3.8 virtual demand responses for roll and pitch between the two schemes are similar. This is explained from Equation 3.22, where it can be seen the roll and pitch channels cannot be controlled separately. For the yaw channel which is distinct, the integrated scheme's yaw channel response is superior to the conventional scheme. This is expected from the earlier section

analysis which shows the superiority of the integrated scheme over the conventional scheme when possible.

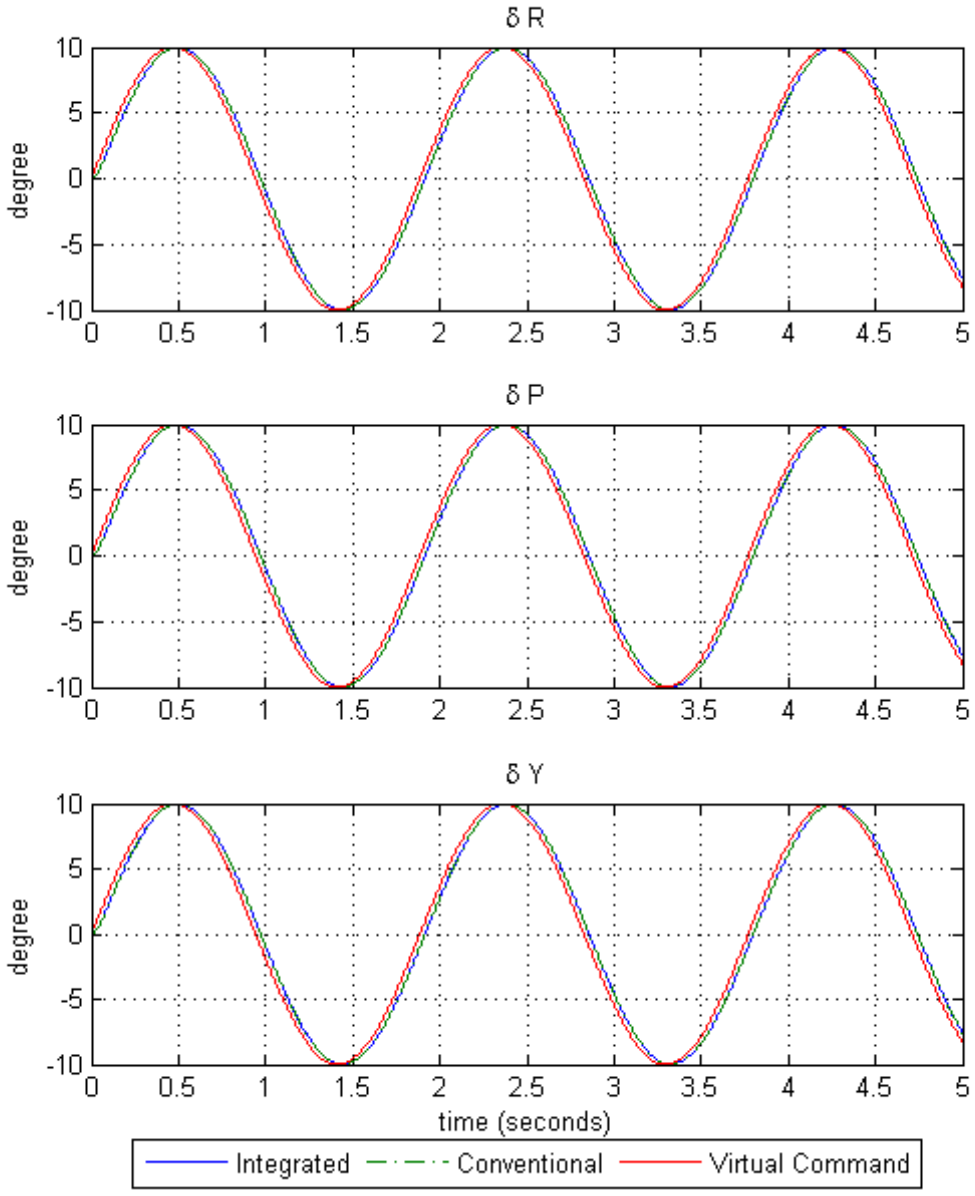

Figure 3.5: Simulation results showing the 2 control schemes having exact responses when there is no actuator saturation events.

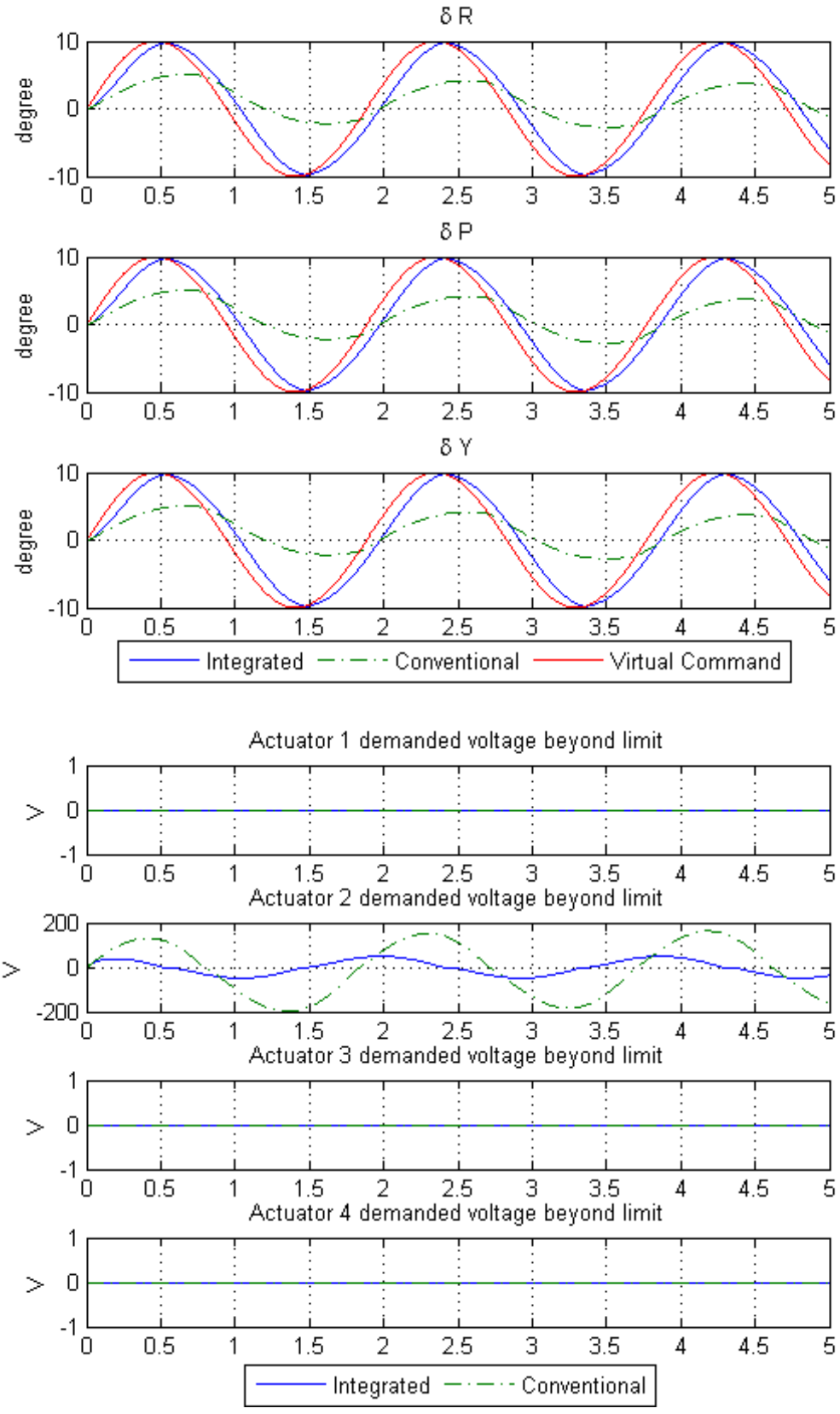

Figure 3.6: Simulation results for 1 actuator saturation case.

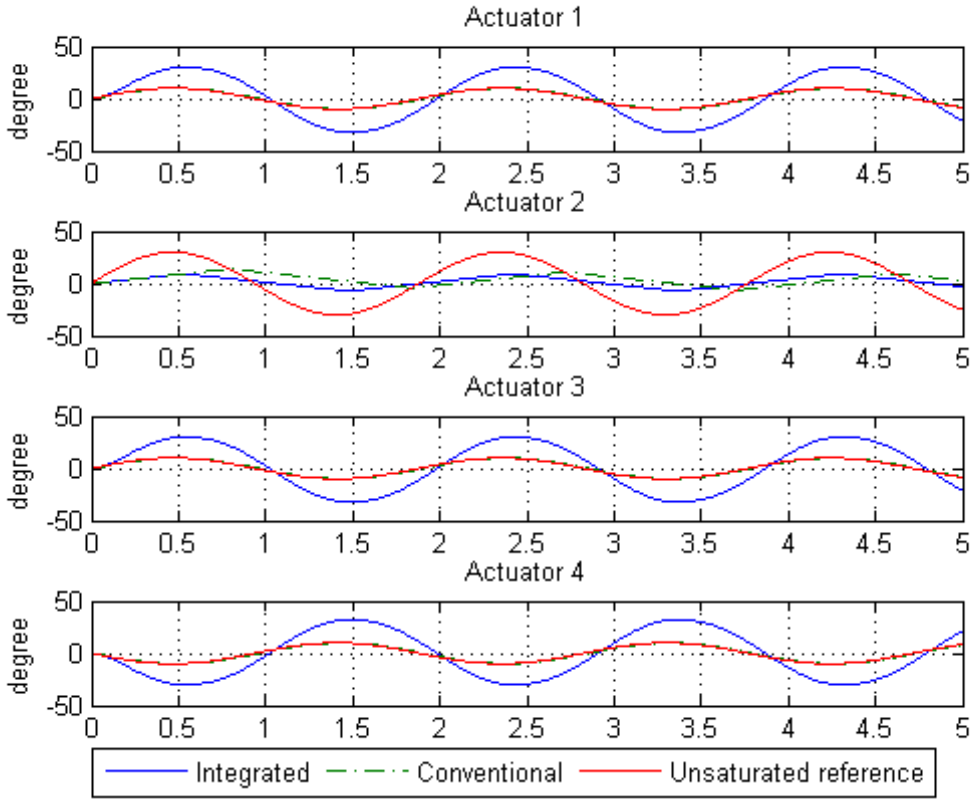

Figure 3.7: Individual actuator responses for 1 actuator saturation case.

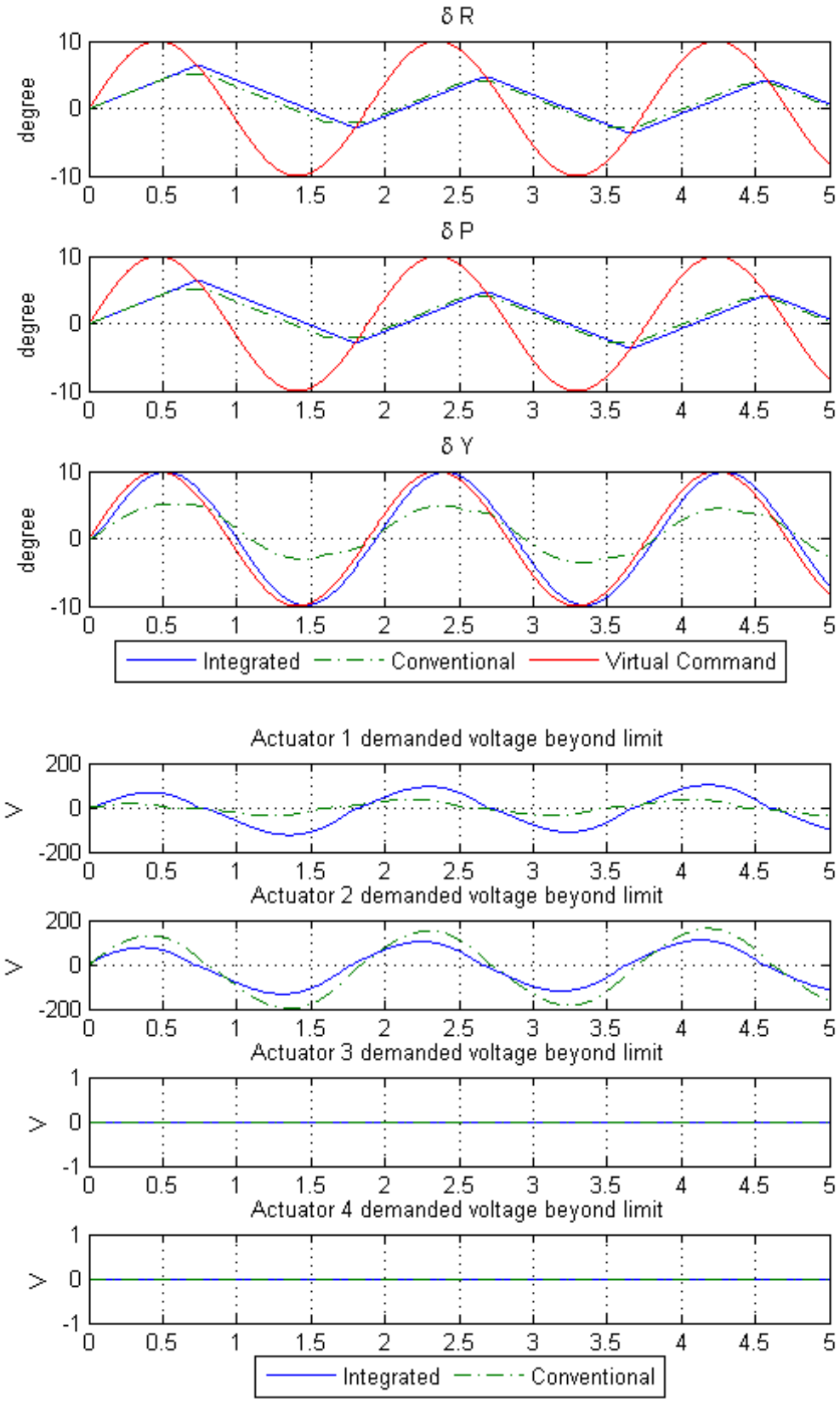

Figure 3.8: Simulation results for Actuator 1 and 2 saturation case.

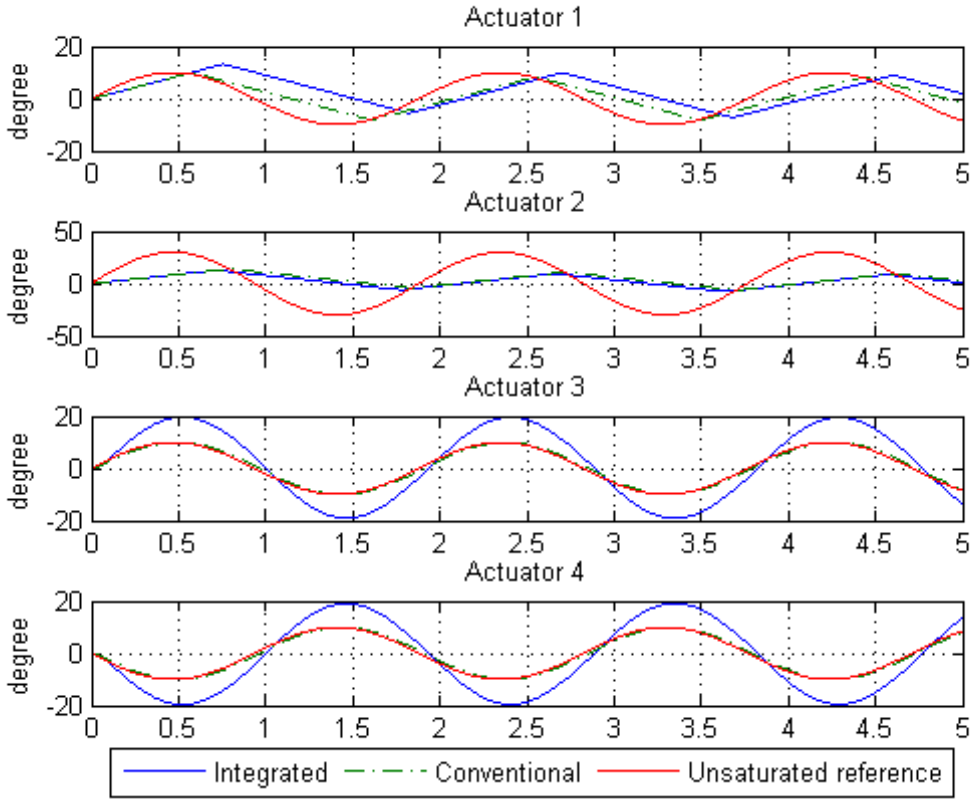

Figure 3.9: Individual actuator response for Actuator 1 and 2 saturation case.

3.4 Conclusion

In this chapter, an integrated actuator control scheme is presented. Analytical analysis of an integrated actuator control scheme over conventional control scheme is conducted for a tail-fin controlled UAV. By directly regulating in the autopilot demand space, the change in the control variables over conventional actuator control schemes improves the tracking performance during actuator saturation when there is available actuator redundancy available. The integrated scheme is shown to be superior in performance for one actuator fault case with available actuator redundancy.

In the event when two actuators are faulty, the integrated scheme's performance continues to be superior in than the conventional scheme for the virtual demand channel which is controllable. The next chapter presents a more generalized analysis of the integrated scheme's analysis.

Chapter 4

Discrete domain analysis on the integrated actuator control scheme

4.1 Introduction

The previous chapter described the superior performance of an integrated control allocation and actuator control scheme using frequency domain analysis on a four tail fin UAV. To utilize the analytically redundant actuator, the integrated scheme regulates the error in the virtual moment space rather than the physical fin deflection space. An advantage of this scheme is that it improves the performance of the actuator system during saturation fault, with a smaller magnitude of error dynamics compared with conventional scheme [86], while keeping the computational requirements low for implementation.

In this chapter, the previous chapter's analysis is generalized using an alternative discrete domain approach. Using this approach, the characteristics of the integrated scheme and conventional scheme are investigated, and new geometrical insights into the scheme are obtained. One of the main focuses of this chapter is to theoretically prove the superior performance of the integrated scheme over the conventional scheme when practical input saturation fault is encountered, and circumstances when it will be similar. A drawback of the integrated scheme potentially having larger steady state actuator positions is addressed. Two possible solutions mitigating this issue are also proposed in this chapter. Simulation results verifying the analysis are shown.

4.2 Conventional control scheme

Fig. 4.1 shows a typical control architecture of an aerial vehicle. The flight computer computes the guidance and flight control laws to generate the desired virtual moment demands. In over-actuated systems, the desired virtual demands are allocated to individual actuators in the actuation system. Each actuator controller

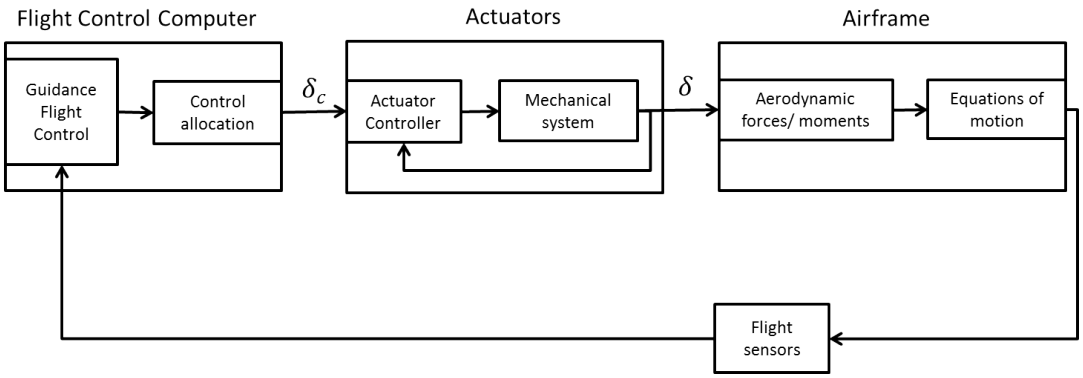

Figure 4.1: A typical vehicle autopilot scheme with the actuation system block.

controls its actuator motor to achieve the desired actuator command with their sensors. In an aerial vehicle model, the actuator positions may be translated into forces and moments, before being used to update the equations of motion of a rigid body.

$$\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\,\mathbf{u} \tag{4.1}$$

where $\mathbf{x}$ is the vehicle states, and $\mathbf{u}$ consists of the actuator positions and throttle. In cases where there is actuator redundancy, control allocation techniques may be used to map the desired virtual demands $\delta_{\mathbf{mc}}$ generated by the flight control to the actuator commands $\delta_{\mathbf{c}}$

$$\delta_{mc} = \mathbf{P_{ca}}\delta_{\mathbf{c}} \tag{4.2}$$

where $\mathbf{P_{ca}}$ is the control effectiveness matrix. It is assumed the variation of $\mathbf{P_{ca}}$ is small about a trim point. One allocation method is to minimise the cost function

$$J = \sum_{i=1}^{p} {\delta_i}^2 \tag{4.3}$$

where there are p actuators in the system with q virtual demands. Solving Eqn (4.3) results in the allocator $\mathbf{B_{ca}}$ as follows:

$$\mathbf{B_{ca}} = \mathbf{P_{ca}^T}\left(\mathbf{P_{ca}}\mathbf{P_{ca}^T}\right)^{-1} \tag{4.4}$$

$$\mathbf{P_{ca}}\mathbf{B_{ca}} = \mathbf{I_q} \tag{4.5}$$

where $\mathbf{B_{ca}}$ is the pseudo inverse control allocator.

A conventional control allocation and actuator control scheme is represented in Fig. 4.2. First, the desired virtual demand $\delta_{\mathbf{mc}}$ is generated by the system's

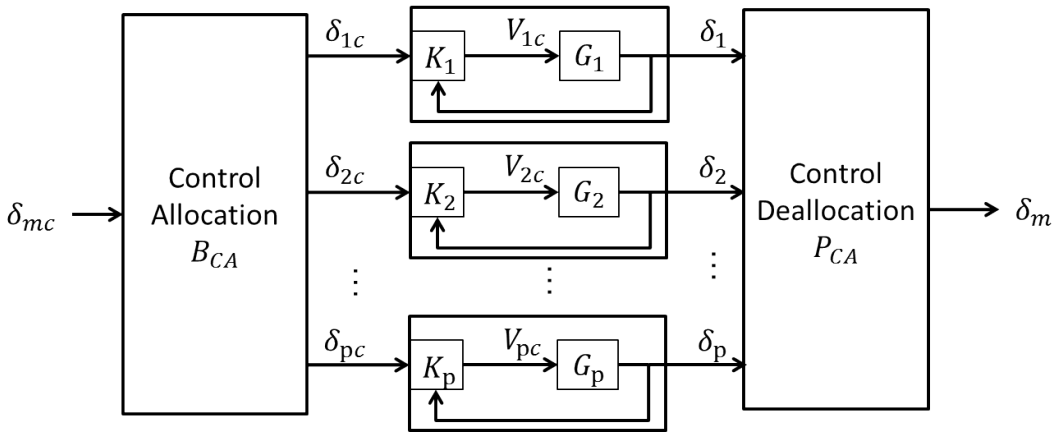

Figure 4.2: A conventional actuator control scheme.

control law, which is allocated to the individual actuators as commands by a linear allocator $\mathbf{B_{ca}}$.

In the actuation system, a discrete domain model of an i^{th} actuator with input saturation can be described as

$$x_{ik+1} = A_i x_{ik} + B_i sat\,(u_{ik}) \tag{4.6}$$

$$y_{ik} = C_i x_{ik} \tag{4.7}$$

$$u_{ik} = K_i\,(\delta_{ic} - y_{ik}) \tag{4.8}$$

where *sat*() refers to the standard saturation function:

$$\text{sat}(u_{ik}) := \begin{cases} \overline{u_i} & u_{ik} > \overline{u_i} \\ u_{ik} & \underline{u_i} \le u_{ik} \le \overline{u_i} \\ \underline{u_i} & u_{ik} < \underline{u_i} \end{cases} \tag{4.9}$$

with $A_i \in \mathbb{R}^{n\times n}, B_i \in \mathbb{R}^{n\times m}, C_i \in \mathbb{R}^{l\times n}$ being the individual actuator state space model parameters. n, m, l are the number of states, control inputs, and sensor outputs of the individual actuator respectively. $K_i = [k_{i1}, k_{i2} \ldots, k_{il}]$ is the actuator controller.

Let p be the number of actuators in the system with q desired virtual demands, $\delta_{\mathbf{mc}} \in \mathbb{R}^q$. The pseudo inverse control allocation and actuator system (termed

conventional scheme) can be represented as

$$\mathbf{x_{k+1}} = \mathbf{A_p x_k} + \mathbf{B_p} \text{sat}\left(\mathbf{K_o}\left(\mathbf{B_{ca}}\delta_{\mathbf{mc}} - \mathbf{C_p x_k}\right)\right) \tag{4.10}$$

$$\mathbf{y_k} = \mathbf{C_p x_k} \tag{4.11}$$

$$\delta_{\mathbf{m}} = \mathbf{P_{ca} C_p x_k} \tag{4.12}$$

$$= \mathbf{P_{ca} y_k} \tag{4.13}$$

where $\mathbf{x_k}$ is the k^{th} sample of the states of the system. $\mathbf{A_p} = diag(A_1, \ldots A_p)$, $\mathbf{B_p} = diag(B_1, \ldots B_p)$, $\mathbf{C_p} = diag(C_1, \ldots C_p)$. $\mathbf{K_o} = diag(K_1, \ldots K_p)$ represents the actuator controllers which are decoupled from one another. The control allocation matrix $\mathbf{B_{ca}} \in \mathbb{R}^{p \times q}$ is assumed to have full column rank. It is also assumed that there are more actuators than virtual demands in the system, $p > q$, or analytical redundancy in the actuation system. This implies there exist non-unique actuator $\mathbf{y_k}$ combinations which produce the same virtual demand $\delta_{\mathbf{m}}$ in the virtual demand space.

4.3 Integrated actuator control scheme

In the integrated scheme, the idea is to shift the allocator from the flight computer block to the actuation block to combine the actuators as shown in Fig. 4.3. Fig. 4.4 shows the integrated actuator control control scheme in detail. First, the actuator controller acts on the virtual demand error, which is presented to the pseudo inverse control allocation algorithm $\mathbf{B_{ca}}$ for allocating to the individual actuators.

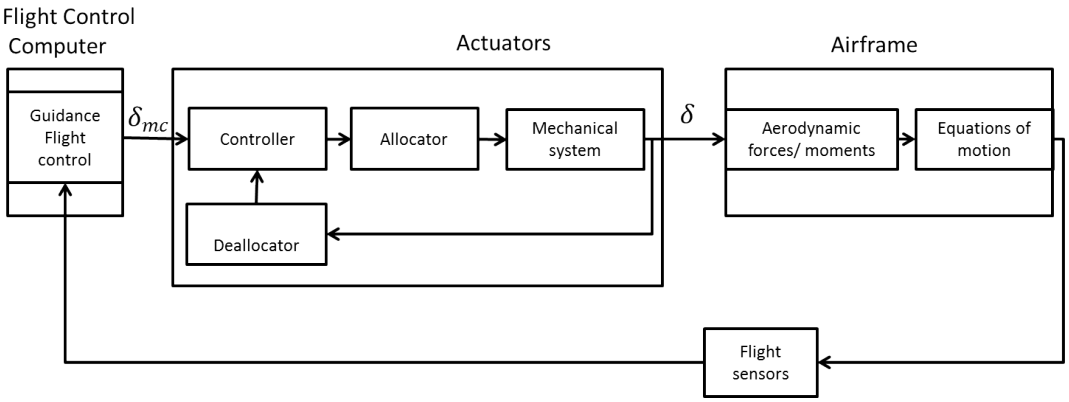

Figure 4.3: Integrated Actuator Control Architecture

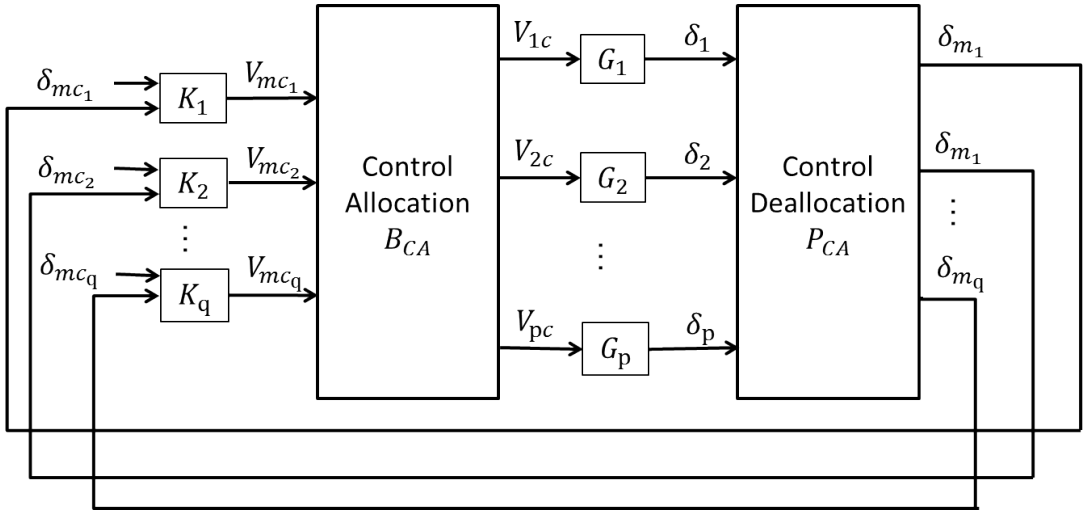

Figure 4.4: Integrated Actuator Control Architecture in detail

The actuator responses are next deallocated by $\mathbf{P_{ca}}$ for transforming the actuator responses to the virtual demand $\delta_\mathbf{m}$ space to be used for control. A constant $\mathbf{P_{ca}}$ is first considered, which can usually be obtained from trim analysis, as we do not consider faults which drastically change the aerodynamic coefficients here. The integrated actuator control scheme can be expressed as

$$\mathbf{x_{k+1}} = \mathbf{A_p x_k} + \mathbf{B_p} \text{sat}\left(\mathbf{B_{ca} K_n}\left(\delta_\mathbf{mc} - \mathbf{P_{ca} C_p x_k}\right)\right) \tag{4.14}$$

with $\mathbf{K_n} = diag(K_1, K_2 \ldots K_q)$ and K_i being the individual actuator controller.

In the case when an actuator control may have more than one input or output $m > 1$ or $l > 1$, the control allocation and deallocation matrices have to be scaled accordingly for dimension consistency. For instance, when $l > 1$, the relationship between $\mathbf{B_{ca}}$ and $\mathbf{B^l_{ca}}$ is described in Eqns. 4.15 and 4.16 below

$$\mathbf{B_{ca}} = \begin{bmatrix} b_{11} & b_{12} & \ldots & b_{1q} \\ b_{22} & b_{22} & \ldots & b_{2q} \\ \vdots & & \ddots & \vdots \\ b_{p1} & b_{p2} & \ldots & b_{pq} \end{bmatrix} \tag{4.15}$$

$$\mathbf{B_{ca}^{l}} = \mathbf{B_{ca}} \otimes \mathbf{I_{l}} \tag{4.16}$$

$$= \begin{bmatrix} b_{11}I_l & b_{12}I_l & \dots & b_{1q}I_l \\ b_{22}I_l & b_{22}I_l & \dots & b_{2q}I_l \\ \vdots & & \ddots & \vdots \\ b_{p1}I_l & b_{p2}I_l & \dots & b_{pq}I_l \end{bmatrix} \tag{4.17}$$

with I_l being the identity matrix of dimension l. Similarly,

$$\mathbf{B_{ca}^{m}} = \begin{bmatrix} b_{11}I_m & b_{12}I_m & \dots & b_{1q}I_m \\ b_{22}I_m & b_{22}I_m & \dots & b_{2q}I_m \\ \vdots & & \ddots & \vdots \\ b_{p1}I_m & b_{p2}I_m & \dots & b_{pq}I_m \end{bmatrix} \tag{4.18}$$

In this work, the scaling of the $\mathbf{B_{ca}}$ and $\mathbf{P_{ca}}$ is implicitly assumed, unless illustrated during matrix manipulations for clarity.

4.4 Properties of the integrated actuator control scheme

4.4.1 Linear Equivalence in the Virtual Demand space

The two control schemes are shown to be equivalent to each other in the virtual demand space when operating in the linear region. By removing the saturation functions in Eqns. (4.10) and (4.14), the linear region for the conventional scheme can be described as

$$\mathbf{x_{k+1}} = \mathbf{A_p x_k} + \mathbf{B_p K_o} \left(\mathbf{B_{ca}^{l}} \delta_{\mathbf{mc}} - \mathbf{y_k} \right) \tag{4.19}$$

Similarly, the integrated control scheme can be expressed as

$$\mathbf{x_{k+1}} = \mathbf{A_p x_k} + \mathbf{B_p B_{ca}^{m} K_n} \left(\delta_{\mathbf{mc}} - \delta_{\mathbf{m}} \right) \tag{4.20}$$

where $\mathbf{B_{ca}^{l}}$ and $\mathbf{B_{ca}^{m}}$ are defined in Eqns. (4.16) and (4.18) respectively.

Proposition 1. *Suppose the individual actuator plants and controllers are the same, the two control schemes are equivalent in the virtual demand space in fault/ saturation free scenarios.*

Proof. First, assume the individual actuator plants and controllers are the same. This assumption may not be very restrictive for some UAVs types like quadrotor UAVs due to symmetry. In some other small UAV designs, the system designer may choose the same actuators due to small differences in loads or demands. With this, the plant matrices can be partitioned in block diagonalized forms as follows

$$\begin{aligned}\mathbf{A_p} &= diag(A_1, \ldots, A_p) \\ &=: diag(A)^p \end{aligned} \tag{4.21}$$

$$\begin{aligned}\mathbf{B_p} &= diag(B_1, B_2, \ldots, B_p) \\ &=: diag(B)^p \end{aligned} \tag{4.22}$$

$$\begin{aligned}\mathbf{C_p} &= diag(C_1, C_2, \ldots, C_p) \\ &=: diag(C)^p \end{aligned} \tag{4.23}$$

$$\begin{aligned}\mathbf{K_o} &= diag(K_1, K_2, \ldots, K_p) \\ &=: diag(K)^p \end{aligned} \tag{4.24}$$

$$\begin{aligned}\mathbf{K_n} &= diag(K_1, K_2, \ldots, K_q) \\ &=: diag(K)^q \end{aligned} \tag{4.25}$$

Similarly, $\mathbf{B_{ca}}$ can be block partitioned into the appropriate $p \times q$ partitions as follows

$$\mathbf{B_{ca}} = \begin{bmatrix} b_{11} & b_{12} & \ldots & b_{1q} \\ b_{22} & b_{22} & \ldots & b_{2q} \\ \vdots & & \ddots & \vdots \\ b_{p1} & b_{p2} & \ldots & b_{pq} \end{bmatrix} \tag{4.26}$$

Putting Eqns. (4.22), (4.25) and (4.26) into Eqn (4.20) yields

$$\begin{aligned}\mathbf{x_{k+1}} =& \mathbf{A_p x_k} + \mathbf{B_p B_{ca}^m K_n}\,(\delta_{\mathbf{mc}} - \delta_{\mathbf{m}}) \\ =& \mathbf{A_p x_k} + \begin{bmatrix} Bb_{11}K & Bb_{12}K & \dots & Bb_{1q}K \\ Bb_{22}K & Bb_{22}K & \dots & Bb_{2q}K \\ \vdots & & \ddots & \vdots \\ Bb_{p1}K & Bb_{p2}K & \dots & Bb_{pq}K \end{bmatrix} (\delta_{\mathbf{mc}} - \delta_{\mathbf{m}})\end{aligned} \tag{4.27}$$

Noticing in Eqn (4.27) that the terms formed in the matrix multiplication of $\mathbf{B_p B_{ca}^m K_n}$ has common BK terms, a change in the matrix order can be done as follows

$$\begin{aligned}\mathbf{x_{k+1}} =& \mathbf{A_p x_k} + diag(BK)^p \mathbf{B_{ca}^l}(\delta_{\mathbf{mc}} - \delta_{\mathbf{m}}) \\ =& \mathbf{A_p x_k} + \mathbf{B_p K_o B_{ca}^l}(\delta_{\mathbf{mc}} - \delta_{\mathbf{m}})\end{aligned} \tag{4.28}$$

Now, multiply $\mathbf{P_{ca}^l C_p}$ to both sides of Eqn (4.28) to transform it to the virtual demand space results in

$$\begin{aligned}&\mathbf{P_{ca}^l C_p x_{k+1}} \\ &= \mathbf{P_{ca}^l C_p A_p x_k} + \mathbf{P_{ca}^l} diag(CBK)^p \mathbf{B_{ca}^l}\,(\delta_{\mathbf{mc}} - \delta_{\mathbf{m}}) \\ &= \mathbf{P_{ca}^l C_p A_p x_k} + \mathbf{P_{ca}^l} diag(CBK)^p \mathbf{B_{ca}^l}\,(\delta_{\mathbf{mc}} - \delta_{\mathbf{m}}) \\ &= \mathbf{P_{ca}^l C_p A_p x_k} + diag(CBK)^q \mathbf{P_{ca}^l B_{ca}^l}\,(\delta_{\mathbf{mc}} - \delta_{\mathbf{m}}) \\ &= \mathbf{P_{ca}^l C_p A_p x_k} + diag(CBK)^q (\delta_{\mathbf{mc}} - \delta_{\mathbf{m}})\end{aligned} \tag{4.29}$$

Similarly for the conventional scheme, multiplying $\mathbf{P_{ca}^l C_p}$ to both sides of Eqn (4.19) results in

$$\begin{aligned}&\mathbf{P_{ca}^l C_p x_{k+1}} \\ &= \mathbf{P_{ca}^l C_p A_p x_k} + \mathbf{P_{ca}^l C_p B_p K_o}\left(\mathbf{B_{ca}^l}\delta_{\mathbf{mc}} - \mathbf{y_k}\right) \\ &= \mathbf{P_{ca}^l C_p A_p x_k} + \mathbf{P_{ca}^l} diag(CBK)^p\left(\mathbf{B_{ca}^l}\delta_{\mathbf{mc}} - \mathbf{y_k}\right) \\ &= \mathbf{P_{ca}^l C_p A_p x_k} + diag(CBK)^q \mathbf{P_{ca}^l}\left(\mathbf{B_{ca}^l}\delta_{\mathbf{mc}} - \mathbf{y_k}\right) \\ &= \mathbf{P_{ca}^l C_p A_p x_k} + diag(CBK)^q\left(\delta_{\mathbf{mc}} - \mathbf{P_{ca}^l y_k}\right) \\ &= \mathbf{P_{ca}^l C_p A_p x_k} + diag(CBK)^q (\delta_{\mathbf{mc}} - \delta_{\mathbf{m}})\end{aligned} \tag{4.30}$$

Observing that Eqns (4.29) and (4.30) are equal, the linear equivalence between the two control schemes in the virtual demand space is shown. □

Remark 1. The analysis here reaffirms the previous analysis made in the frequency domain, where it is shown the two schemes are equivalent when linear.

4.4.2 Orthogonal Control Update direction

The previous section has shown the two control schemes are equivalent in the virtual demand space while operating linearly. However, the two control schemes do not update the control signal in the same direction. This section describes the differences between the two control schemes. The control update direction for the integrated scheme is shown to be orthogonal to the actuator solution space, while the conventional scheme's direction is towards the desired virtual demand point in the actuator space.

As it is assumed the system has analytical redundancy in the actuator space, there exists an actuator solution set Ω_{δ_m} where the actuators have the same virtual demand $\delta_{\mathbf{m}}$.

Definition 1. Let the actuator solution set $\Omega_{\delta_m} \in \mathbb{R}^p$ describe all the possible actuator combinations $\mathbf{y_k}$ which is mapped to an equivalent virtual demand $\delta_{\mathbf{m}}$:

$$\Omega_{\delta_m} =: \{\mathbf{y_k} \in \mathbb{R}^p | \mathbf{P_{ca}} \mathbf{y_k} = \delta_{\mathbf{m}}\} \tag{4.31}$$

Similarly, let Ω_{δ_m} be the set of desired actuator outputs $\mathbf{y_{ds}}$ which maps to the desired virtual demand δ_{mc} in the virtual demand space:

$$\Omega_{\delta_{mc}} =: \{\mathbf{y_{ds}} \in \mathbb{R}^p | \mathbf{P_{ca}} \mathbf{y_{ds}} = \delta_{\mathbf{mc}}\} \tag{4.32}$$

Lemma 1. $\mathbf{B_{ca}}\delta_{\mathbf{mc}}$ *is orthogonal to the desired actuator solution space* $\Omega_{\delta_{mc}}$.

Proof. Let $\mathbf{y_{1s}}$ and $\mathbf{y_{2s}}$ be two distinct actuator outputs fulfilling the desired virtual demand. They fulfil the following relationships:

$$\mathbf{P_{ca}} \mathbf{y_{1s}} = \delta_{\mathbf{mc}} \tag{4.33}$$

$$\mathbf{P_{ca}} \mathbf{y_{2s}} = \delta_{\mathbf{mc}} \tag{4.34}$$

The vector $(\mathbf{y_{2s}} - \mathbf{y_{1s}})$ lies on the desired actuator set $\Omega_{\delta_{mc}}$. The inner product between $\mathbf{y_{2s}} - \mathbf{y_{1s}}$ and $\mathbf{B_{ca}}\delta_{\mathbf{mc}}$ is given by

$$\begin{aligned}
&\langle \mathbf{y_{2s}} - \mathbf{y_{1s}}, \mathbf{B_{ca}}\delta_{\mathbf{mc}} \rangle \\
&= (\mathbf{y_{2s}} - \mathbf{y_{1s}})^{\mathbf{T}} \mathbf{B_{ca}}\mathbf{P_{ca}}\mathbf{y_{ds}} \\
&= \mathbf{y_{2s}^T}\mathbf{B_{ca}}\mathbf{P_{ca}}\mathbf{y_{ds}} - \mathbf{y_{1s}^T}\mathbf{B_{ca}}\mathbf{P_{ca}}\mathbf{y_{ds}}
\end{aligned} \tag{4.35}$$

Since $\mathbf{B_{ca}}\mathbf{P_{ca}}$ is symmetrical due to the pseudo inverse relationship, $(\mathbf{B_{ca}}\mathbf{P_{ca}})^T = \mathbf{B_{ca}}\mathbf{P_{ca}}$[87] . From this and Eqn (4.32), we have,

$$\begin{aligned}
&\langle \mathbf{y_{2s}} - \mathbf{y_{1s}}, \mathbf{B_{ca}}\delta_{\mathbf{mc}} \rangle \\
&= \mathbf{y_{2s}^T} (\mathbf{B_{ca}}\mathbf{P_{ca}})^{\mathbf{T}} \mathbf{y_{ds}} - \mathbf{y_{1s}^T} (\mathbf{B_{ca}}\mathbf{P_{ca}})^{\mathbf{T}} \mathbf{y_{ds}} \\
&= (\mathbf{B_{ca}}\mathbf{P_{ca}}\mathbf{y_{2s}})^{\mathbf{T}} \mathbf{y_{ds}} - (\mathbf{B_{ca}}\mathbf{P_{ca}}\mathbf{y_{1s}})^{\mathbf{T}} \mathbf{y_{ds}} \\
&= (\mathbf{B_{ca}}\delta_{\mathbf{mc}})^{\mathbf{T}} \mathbf{y_{ds}} - (\mathbf{B_{ca}}\delta_{\mathbf{mc}})^{\mathbf{T}} \mathbf{y_{ds}} \\
&= \mathbf{0}
\end{aligned} \tag{4.36}$$

This implies that $\mathbf{B_{ca}}\delta_{\mathbf{mc}}$ and the desired actuator solution space $\Omega_{\delta_{mc}}$ is orthogonal. □

Lemma 2. $\mathbf{B_{ca}}\mathbf{P_{ca}}$ *is an orthogonal projector of* $\mathbf{y_k}$ *onto* $\mathbf{B_{ca}}\delta_{\mathbf{mc}}$.

Proof. Let $\mathbf{y_1}$ be an arbitrary actuator output. Showing the vector $\mathbf{y_1} - \mathbf{B_{ca}}\mathbf{P_{ca}}\mathbf{y_1}$ is normal to $\mathbf{B_{ca}}\delta_{\mathbf{mc}}$ completes the proof. Taking the inner product between the two vectors,

$$\begin{aligned}
&\langle \mathbf{y_1} - \mathbf{B_{ca}}\mathbf{P_{ca}}\mathbf{y_1}, \mathbf{B_{ca}}\delta_{\mathbf{mc}} \rangle \\
&= \mathbf{y_1^T}\mathbf{B_{ca}}\mathbf{P_{ca}}\mathbf{y_{ds}} - \mathbf{y_1^T} (\mathbf{B_{ca}}\mathbf{P_{ca}})^{\mathbf{T}} \mathbf{B_{ca}}\mathbf{P_{ca}}\mathbf{y_{ds}} \\
&= \mathbf{y_1^T}\mathbf{B_{ca}}\mathbf{P_{ca}}\mathbf{y_{ds}} - \mathbf{y_1^T}\mathbf{B_{ca}}\mathbf{P_{ca}}\mathbf{B_{ca}}\mathbf{P_{ca}}\mathbf{y_{ds}} \\
&= \mathbf{y_1^T}\mathbf{B_{ca}}\mathbf{P_{ca}}\mathbf{y_{ds}} - \mathbf{y_1^T}\mathbf{B_{ca}}\mathbf{P_{ca}}\mathbf{y_{ds}} \\
&= \mathbf{0}
\end{aligned} \tag{4.37}$$

Since $\mathbf{y_1}$ is an arbitrary actuator output, Lemma 2 is proved. □

Using these results, the update direction differences between the two schemes can be compared.

Theorem 1. *The conventional scheme updates the control signal in the direction towards the allocated desired virtual demand point* $\mathbf{B_{ca}}\delta_{\mathbf{mc}}$ *in the actuator space when unsaturated.*

Proof. The conventional scheme's unsaturated controller output is

$$\begin{aligned}\mathbf{u_k} &= \mathbf{K_n}\left(\mathbf{B_{ca}}\delta_{\mathbf{mc}} - \mathbf{y_k}\right) \\ &= diag(K)^p\mathbf{K_n}\left(\mathbf{B_{ca}}\delta_{\mathbf{mc}} - \mathbf{y_k}\right)\end{aligned} \tag{4.38}$$

From Eqn (4.38), the control signal's update direction is characterised by $\mathbf{B_{ca}}\delta_{\mathbf{mc}} - \mathbf{y_k}$, which is towards the $\mathbf{B_{ca}}\delta_{\mathbf{mc}}$. The actuator gains K are diagonal and assumed similar and do not affect the update direction. □

Theorem 2. *The integrated actuator scheme's control signal* $\mathbf{u_k}$ *is orthogonal to the desired virtual demand set* $\Omega_{\delta_{mc}}$ *when unsaturated.*

Proof. When unsaturated, the integrated scheme's control update u_k is

$$\mathbf{u_k} = \mathbf{B_{ca}^m}\mathbf{K_n}\left(\delta_{\mathbf{mc}} - \mathbf{P_{ca}^l}\mathbf{y_k}\right) \tag{4.39}$$

$$= diag(K)^p\mathbf{B_{ca}^l}\delta_{\mathbf{mc}} - diag(K)^p\mathbf{B_{ca}^l}\mathbf{P_{ca}^l}\mathbf{y_k} \tag{4.40}$$

Lemma 1 and Lemma 2 showed the first and second part of the integrated scheme's update in Eqn (4.39) to be orthogonal to the desired virtual demand space. By superposition, the theorem is proven. □

In comparison, the conventional scheme updates the control signal in the direction towards the allocated desired virtual demand point $\mathbf{B_{ca}}\delta_{\mathbf{mc}}$ in the actuator space. This is graphically illustrated in Fig. 4.5 for a simple example of two actuators serving a single virtual demand. When the initial position is not along the orthogonal direction from zero, the integrated scheme updates the actuators to move orthogonally towards the desired virtual demand space. On the other hand, the figure shows the two control schemes directions are equivalent when the states are along the $\mathbf{B_{ca}}\delta_{\mathbf{m}}$.

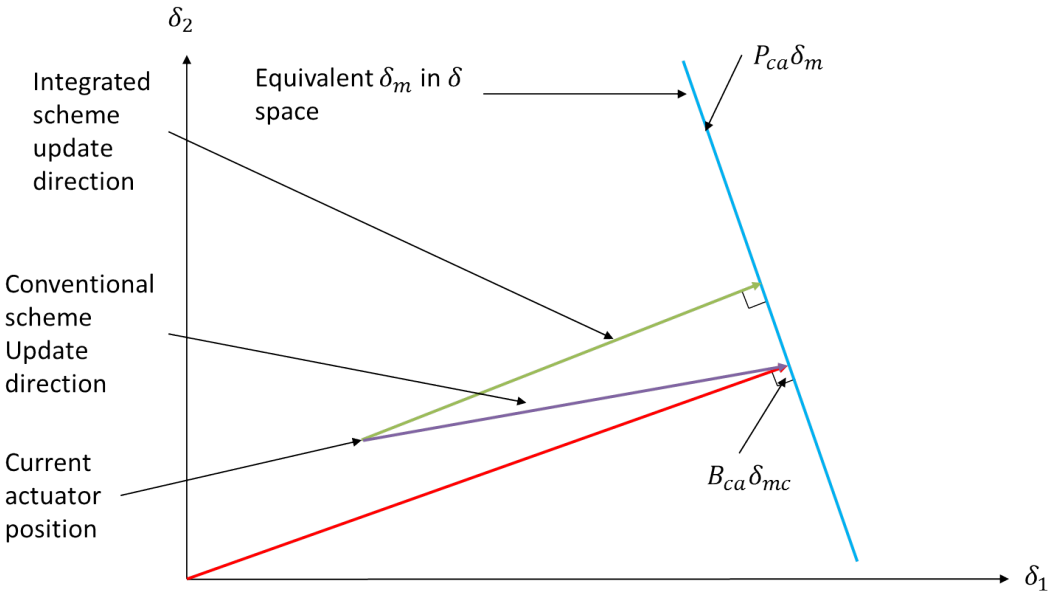

Figure 4.5: Differences in control signal update direction between the integrated and conventional actuator control schemes consisting of two actuators and one virtual demand for illustration. The integrated scheme updates orthogonally to the desired virtual demand space, while the conventional scheme points the control vector towards directly towards $\mathbf{B_{ca}}\delta_{\mathbf{mc}}$.

4.4.3 Output stuck fault behaviour

Consider the effect of an input stuck fault to one of the actuators in the system. The i^{th} faulty actuator's output can be modelled as

$$y_i = y_f \tag{4.41}$$

where y_f is an arbitrary stuck position of the faulty actuator. With the assumption the actuation controller minimizes the error when possible with the output fault, the following observations can be made about the reachability of the virtual demand space by the two schemes. In fault free cases, it is assumed the individual actuators are exponentially stable and tracks the reference command. This implies the error regulated by the control scheme tends towards zero with time, or

$$\lim_{k \to \text{inf}} y_{ik} = r_i \tag{4.42}$$

where r_i is the i^{th} individual actuator reference command, $\mathbf{r} = \mathbf{B_{ca}}\delta_{\mathbf{mc}}$.

Proposition 2. *If the i^{th} faulty actuator is stuck, there exists a steady state error in the virtual demand space for the conventional scheme.*

Proof. For the conventional scheme, the virtual demand error due to the stuck fault may be described as

$$\begin{aligned}\lim_{k\to\inf} \delta_{mk} &= \mathbf{P_{ca}} \begin{bmatrix} y_1 \\ \vdots \\ y_f \end{bmatrix} \\ &= \delta_{mc} + \mathbf{P_{ca}} \begin{bmatrix} 0 \\ \vdots \\ y_f - r_i \end{bmatrix}\end{aligned} \tag{4.43}$$

□

The steady state virtual demand error for the conventional scheme is the second term in the Eqn (4.43). In contrast, the actuator controller in the integrated scheme regulates the virtual demand error. This implies in fault free cases, when the assumption the integrated scheme's controller is stable and regulating the error is valid, the virtual demand error tends towards zero. Mathematically, this means

$$\lim_{k\to\inf} \mathbf{B_{ca}} \left(\delta_{\mathbf{mc}} - \mathbf{P_{ca}} \mathbf{y_k}\right) \to \mathbf{0} \tag{4.44}$$

Proposition 3. *If the i^{th} faulty actuator is stuck, there exists no steady state error in the virtual demand space for the integrated scheme when there is sufficient actuator redundancy.*

Proof. In the stuck fault case,

$$\begin{aligned}
\lim_{k\to\inf} \mathbf{B_{ca}}\mathbf{P_{ca}} \begin{bmatrix} y_{1k} \\ \vdots \\ y_f \end{bmatrix} &= \mathbf{B_{ca}}\mathbf{P_{ca}} \begin{bmatrix} r_1 \\ \vdots \\ y_f \end{bmatrix} \\
&= \begin{bmatrix} bp_{11} & \dots & bp_{1(p-1)} & bp_{1i} \\ bp_{21} & \vdots & bp_{2(p-1)} & bp_{2i} \\ bp_{i1} & \dots & bp_{i(p-1)} & bp_{ii} \end{bmatrix} \begin{bmatrix} r_1 \\ \vdots \\ y_f \end{bmatrix} \\
&= \begin{bmatrix} r_1 - bp_{1i}y_f \\ \vdots \\ r_{p-1} - bp_{(p-1)i}y_f \\ y_f \end{bmatrix} \\
&= \mathbf{r'}
\end{aligned} \tag{4.45}$$

where bp_{ij} is the i^{th} and j^{th} row and column of the matrix $\mathbf{B_{ca}}\mathbf{P_{ca}}$. If the off diagonal terms of $\mathbf{B_{ca}}\mathbf{P_{ca}}$ terms are non-zero with respect to the faulty actuator, the stuck fault value is coupled into the other actuators via the matrix's column. This can be viewed as a change in the nominal reference command for the healthy actuators caused by the fault. With the change in reference command to $\mathbf{r'}$, it can be shown with sufficient actuator redundancy

$$\lim_{k\to\inf} \mathbf{P_{ca}} \left(B_{ca}\delta_{mc} - r'\right) = \mathbf{0} \tag{4.46}$$

□

This result makes the integrated scheme useful for handling actuator stuck faults compared to the conventional scheme, as it iteratively updates the actuators in the presence of fault to reduce the virtual demand error. This alleviates the fault degradation effects felt by the overall control system to possibly enable better performance by the outer control loops. Also, as the actuation system often has higher bandwidth compared to the outer control loops, the fault alleviation process may be faster compared to one implemented in an outer control loop.

4.4.3.1 Possible drawback and possible solutions

When the integrated scheme converges to the desired virtual demand faster than the conventional scheme, the drawback of this performance comes from the over-

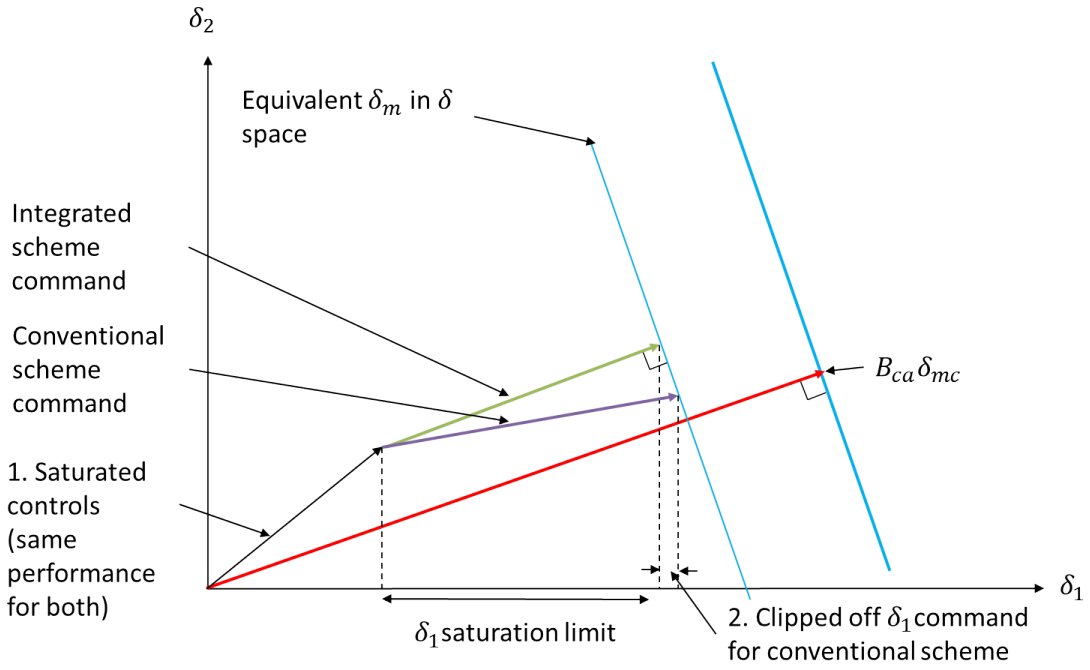

Figure 4.6: Figure illustrating the differences between the integrated and conventional scheme when actuators are input saturated.

all larger magnitude of final actuator positions. This can be seen in Fig. 4.6. The conventional scheme aims toward the unconstrained virtual demand point ($\mathbf{B_{ca}}\delta_{\mathbf{mc}}$), which is minimal distance from origin satisfying the required virtual demand. This results in the smaller actuator trim demand when compared to the integrated scheme. A possible larger trim may not be desirable for some applications. For instance, flight vehicles may suffer from more trim drag when the trimmed actuator positions are larger.

Depending on the requirements of the system to be designed, it may either be desirable to achieve the desired virtual demand in the fastest time, or achieve the minimum actuator position requirement. One possible way to fine tune between the two requirements is to combine the two control schemes via a weighting gain to balance the two demands as follows:

$$u = \alpha u_{integrated} + (1 - \alpha) u_{conventional} \tag{4.47}$$

where $0 \leq \alpha \leq 1$. Eqn. 4.47 will result in a control update direction between the two control schemes, which may be used for fine-tuning to meet a control system's requirements. This solution may be useful to achieve faster results than

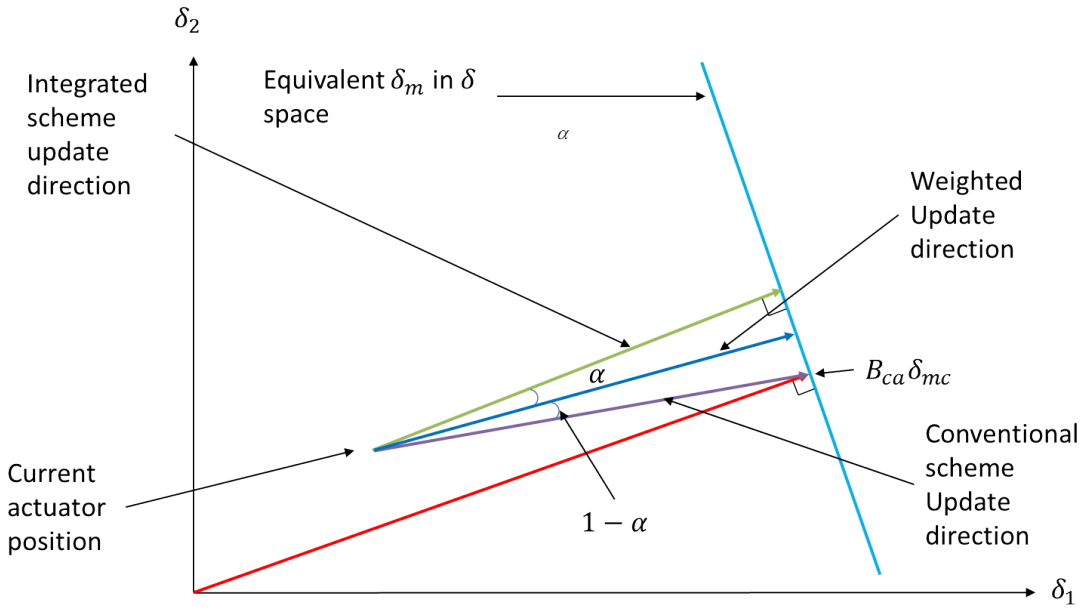

Figure 4.7: Possible solutions to alleviate the drawback of the integrated scheme, which may possibly result in increased final actuator magnitude. The first solution is to combine the two schemes for tuning the actuator update directions for design flexibility.

conventional scheme, while handling possible actuator magnitude limitations. Fig. 4.7 provides a graphical interpretation to the idea.

Another possible method to alleviate the integrated scheme's larger final actuator demands is to first orthogonally reach the desired virtual demand using the integrated scheme, then travel along the null space to reduce the actuator demand using the conventional scheme. This effectively marries the best properties between the two schemes. The real time computational requirement needed for this solution is minimal, since one of the schemes is active only at a time, when compared to the first integrated solution. Fig. 4.8 shows the described two-step control graphically. One can also view this as an adaptive case of the first solution, with $\alpha = 1$ for the first step, before switching to $\alpha = 0$ for the second step.

In the next section, simulations are presented to verify the theoretical analysis.

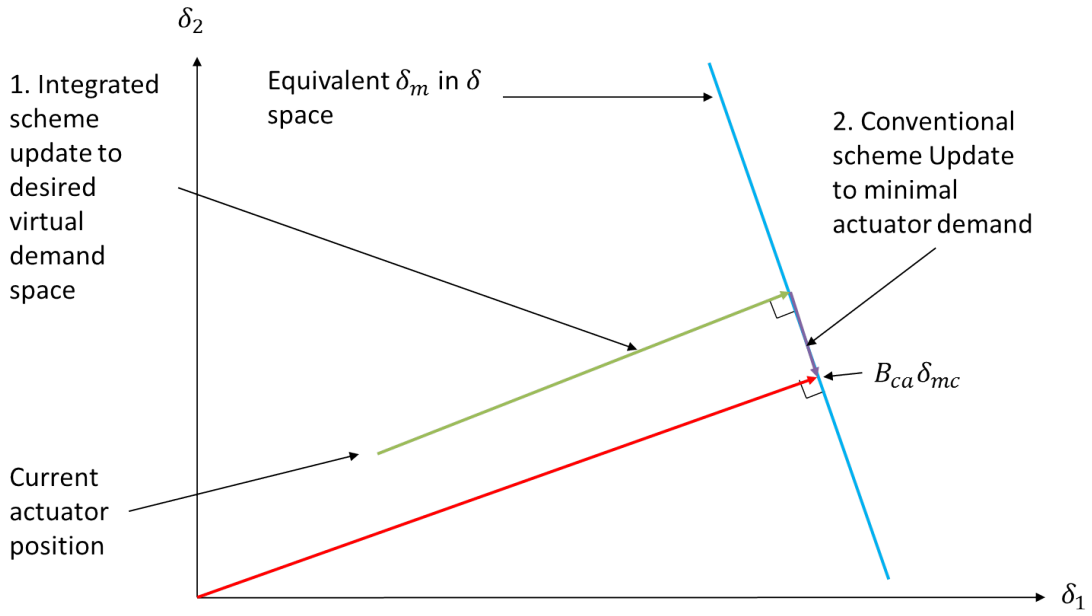

Figure 4.8: Possible solutions to alleviate the drawback of the integrated scheme, which may possibly result in increased final actuator magnitude: The second solution is to first reach the desired virtual demand using the integrated scheme, and subsequently converge to the minimal actuator magnitude position using the conventional scheme.

4.4.4 Simulations and Discussion

A simple example consisting of 2 actuators and 1 virtual demand actuator system is used to illustrate the properties of the integrated scheme described earlier. The actuator plant used in the simulation is described in Ref. [30], which is shown in Figure 4.9. The plant is a typical DC motor with gearing, and explained in Ref. [88]. The parameters used in are detailed in Table 4.1.

A cascaded two loop controller shown in Figure 4.10 is used to provide good control performance of the actuator plant from Ref. [30]. The control law is

$$V_c(t) = K_p\left(\delta_c(t) - \delta(t)\right) - K_d\left(\dot{\delta}(t)\right) \tag{4.48}$$

where δ_{ci} is the commanded deflection angle, and $K_p = 6$ and $K_d = 0.02$ are gains for the control law. The overall simulation diagram is shown in Figs. 4.11 and 4.12, where *CON* represents the controller, and *ACT* the actuator model.

Let the control allocation matrix $\mathbf{B_{ca}}$ and corresponding deallocation matrix

Table 4.1: Actuator plant parameters used in the simulation

Parameters	Values	Parameters	Values
K_T	0.303125	K_B	$5.7333x10^{-4}$
L	0.35×10^{-3}	H	0
R	0.933	V_{lim}	28
J	8.5354×10^{-7}	N	274
B	2.0835×10^{-6}		

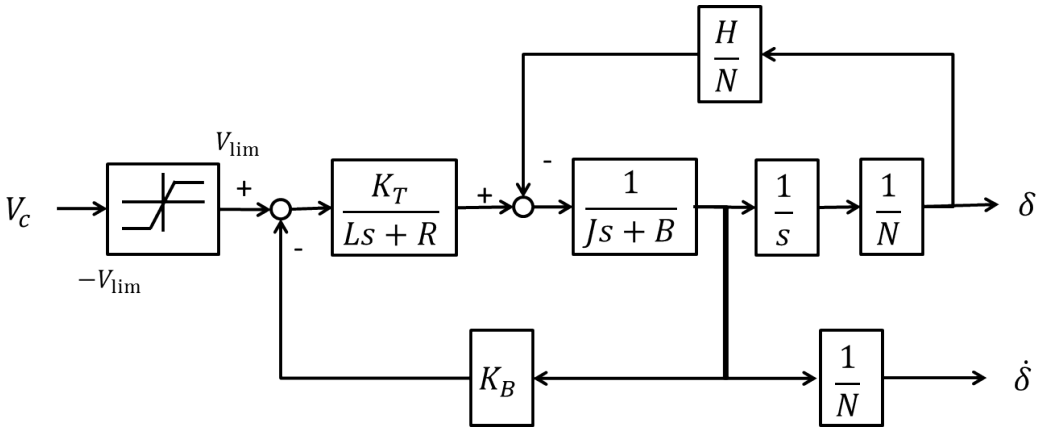

Figure 4.9: Actuator plant model used in the simulation

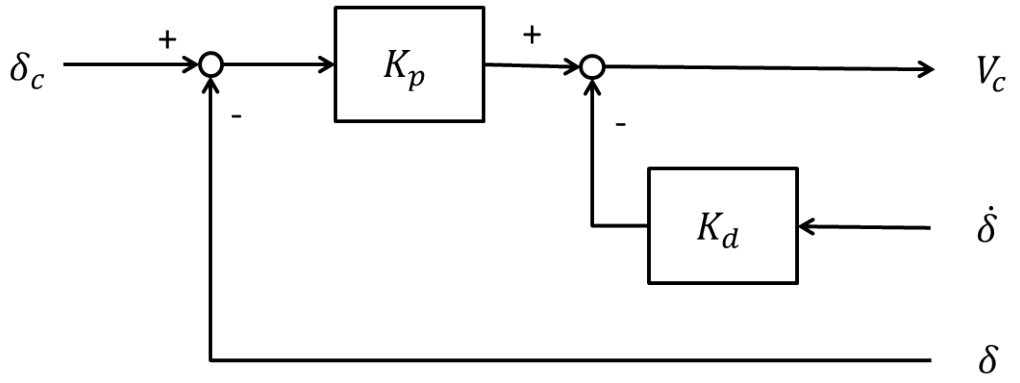

Figure 4.10: Controller used for controlling the actuator plant used in the simulation

$\mathbf{P_{ca}}$ be

$$\mathbf{B_{ca}} = [\ 1 \quad 0.5\ ]^T \tag{4.49}$$

$$\mathbf{P_{ca}} = [\ 0.8 \quad 0.4\ ] \tag{4.50}$$

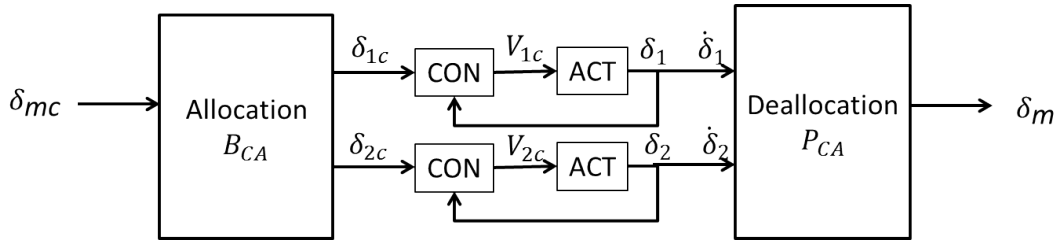

Figure 4.11: Overall simulation diagram setup for the conventional actuator control scheme

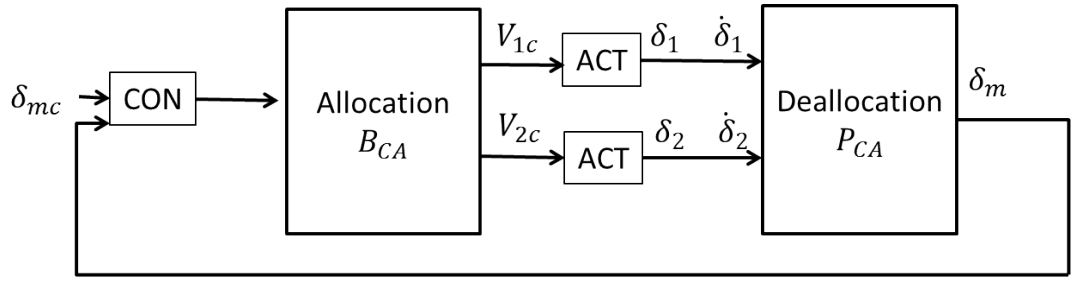

Figure 4.12: Overall simulation diagram setup for the integrated actuator control scheme

4.4.4.1 Unconstrained simulations

First, the differences between the unconstrained control update directions between the two control schemes are illustrated. The initial positions of 0 deg and 10 deg is set for Actuators 1 and 2 respectively. In Fig. 4.13, the unconstrained versions of the two control schemes having the same virtual demand response is shown. However, the actuator position plot in Fig. 4.14 shows the two schemes update very differently to achieve the desired virtual demand. The conventional scheme updates the actuators to the desired virtual demand point $\mathbf{B_{ca}}\delta_{\mathbf{mc}}$, while the integrated scheme updates the actuators orthogonally to the desired virtual demand (line) space $\mathbf{P_{ca}}\delta_{\mathbf{m}}$. Fig. 4.15 shows the individual actuator positions of between the two control schemes, which are different.

The next set of unconstrained simulations set the initial conditions for the actuators are set to zero for both actuators, which is a point along the vector $\mathbf{B_{ca}}\delta_{\mathbf{mc}}$. From the analysis in previous section, the actuator responses profile for both control schemes will coincide due to the geometry. This is verified in Fig. 4.16 and Fig. 4.17, where the virtual demand plots show the equivalence between the two schemes. This is further reinforced in Fig. 4.18, which shows the individual

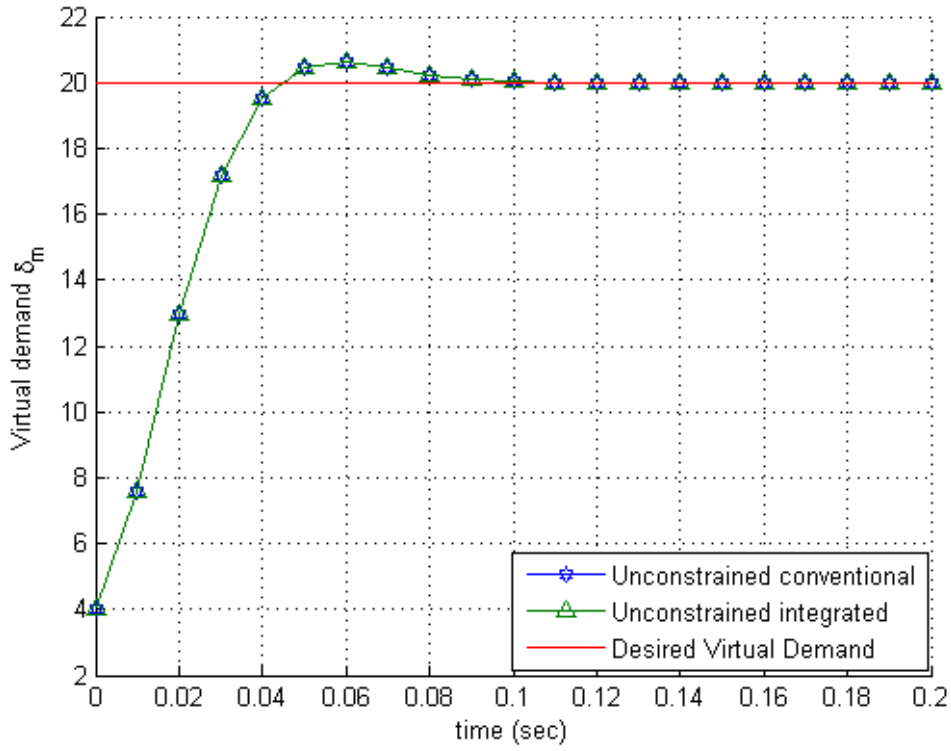

Figure 4.13: Simulation results showing without input saturation constraints for the two schemes. The two control schemes are seen to have exactly the same virtual demand response, as proven in the theoretical analysis presented earlier.

actuator positions being the same for the two control schemes.

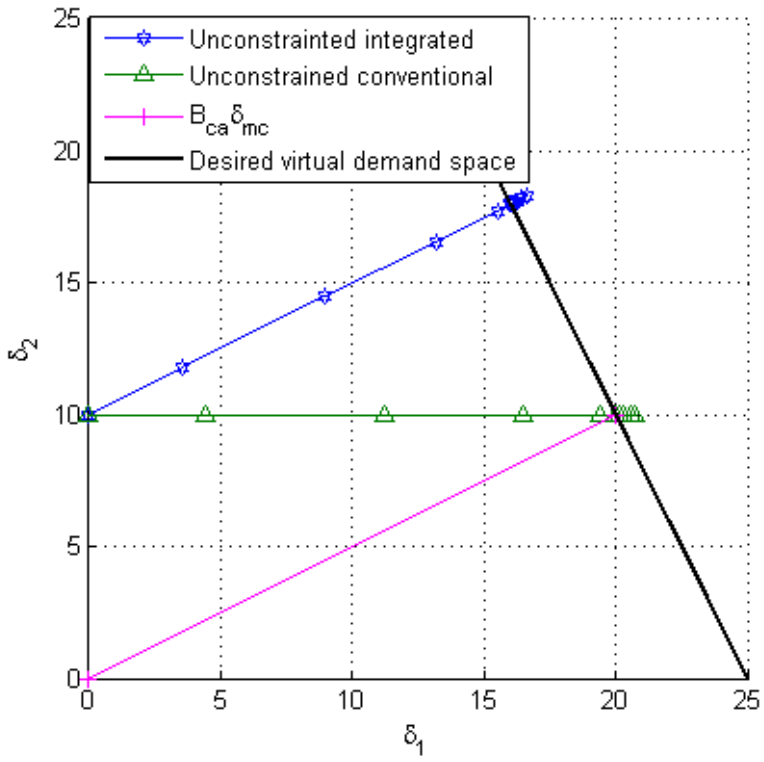

Figure 4.14: Simulation results showing without input saturation constraints for the two schemes with initial conditions not on $\mathbf{B_{ca}}\delta_{mc}$ vector. The integrated scheme is shown to update orthogonally towards the desired virtual demand line (space), while the conventional scheme updates towards the specific actuator position $\mathbf{B_{ca}}\delta_{\mathbf{mc}}$.

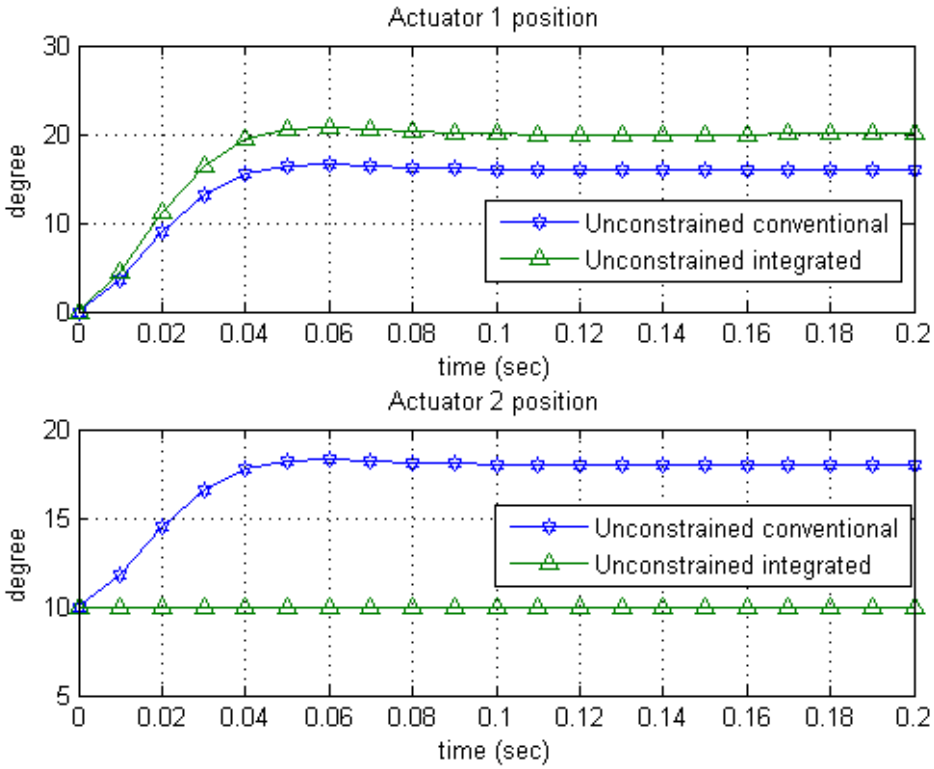

Figure 4.15: Actuator position plots for simulations without input saturation constraints for the two schemes with initial conditions not on $\mathbf{B_{ca}}\delta_{mc}$ vector. The actuator positions are different between the two control schemes due to the update direction differences.

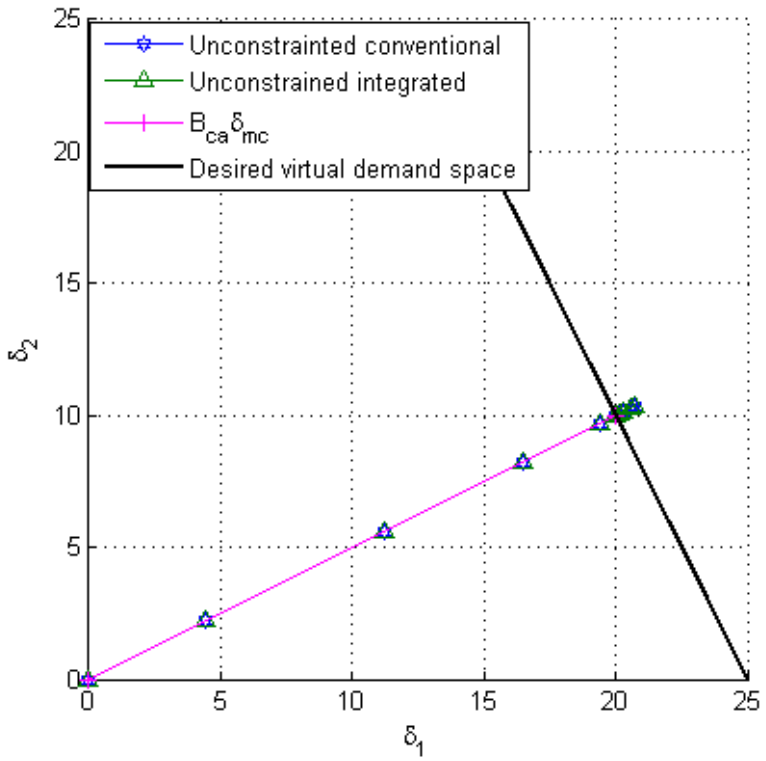

Figure 4.16: Simulation results showing without input saturation constraints for the two schemes with zero initial conditions (which is on $\mathbf{B_{ca}}\delta_{mc}$ vector): It can be seen the two control schemes trajectories are equivalent to each other.

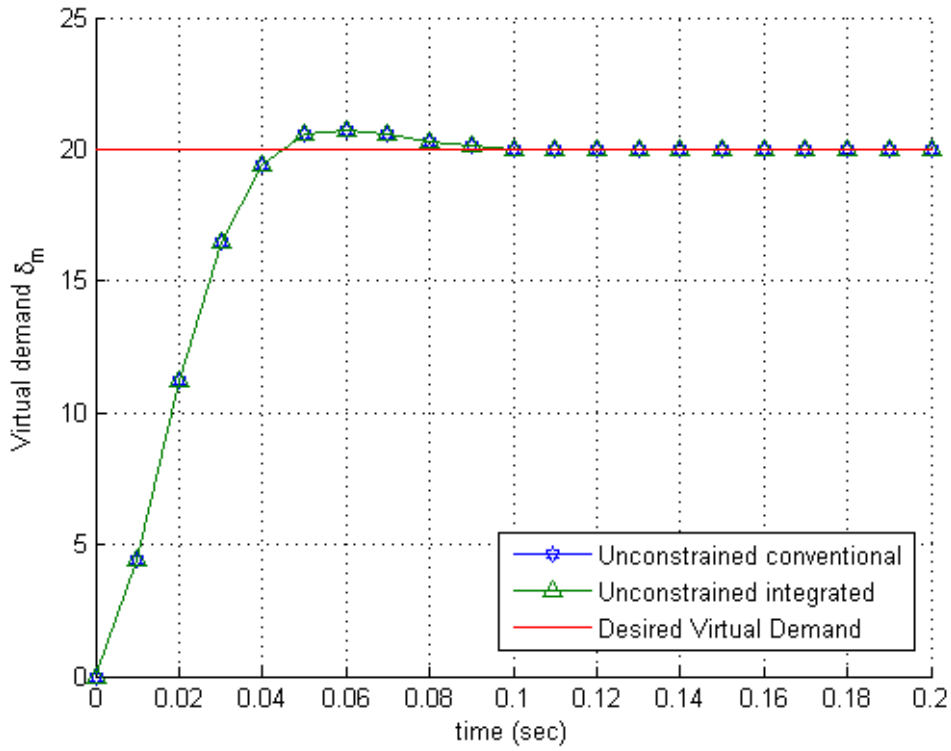

Figure 4.17: Simulation results showing without input saturation constraints for the two schemes, with zero initial conditions. The two control schemes are seen to have exactly the same virtual demand response, as proven in the theoretical analysis presented earlier.

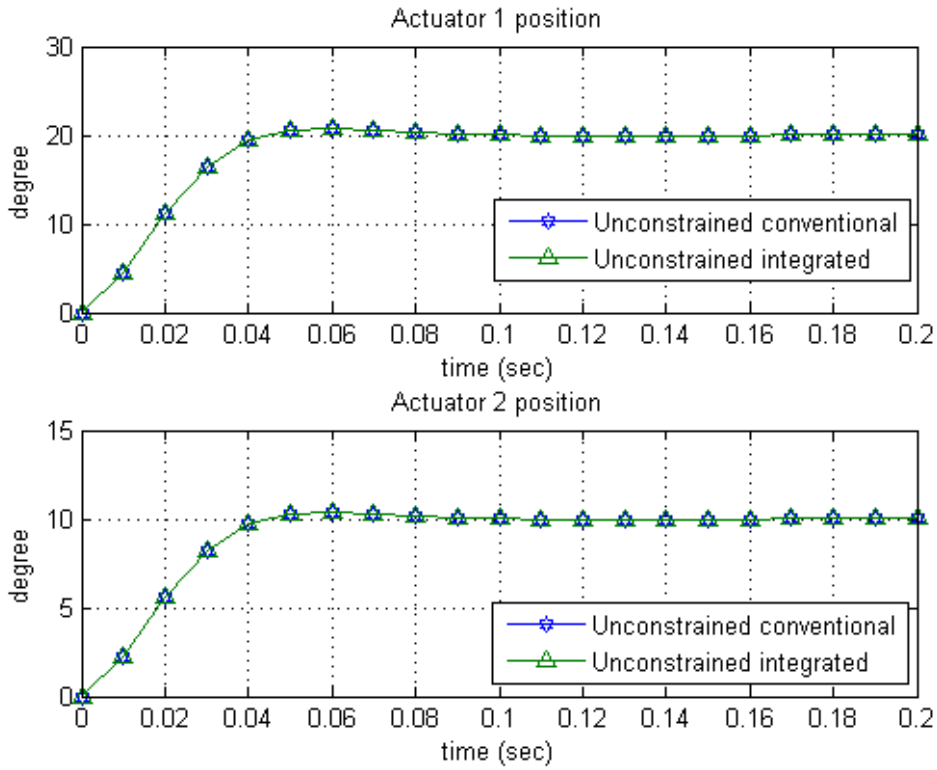

Figure 4.18: Actuator position plots for simulations without input saturation constraints for the two schemes with initial conditions on $\mathbf{B_{ca}}\delta_{mc}$ vector. The actuator positions are equivalent between the two control schemes.

4.4.4.2 Input saturation simulations

Next, the control schemes are subjected to input saturation constraints. With zero initial conditions, the simulation results for the input constrained integrated and conventional schemes are shown for a step virtual demand input in Fig. 4.19. For comparison, the unconstrained solution is also plotted. The integrated scheme is able to reach the desired virtual demand faster when compared to the conventional scheme. Calculating the sum of the absolute error between the desired virtual demand and response as a measure, the integrated scheme has a lower score of 115.6 against the conventional scheme of 122.

Fig. 4.20 shows the relative actuator positions of the two control schemes as they achieve the desired virtual demand. First, both actuators are saturated and thus limited to move at 45 deg in the phase plot for both control schemes. As the actuators desaturate when the virtual demand error reduces, the integrated scheme updates orthogonally towards the desired virtual demand space, while the conventional scheme is seen to move towards the original desired virtual demand point to make it slower when compared to the integrated scheme.

Fig. 4.21 shows the actuators' position responses for the simulation. The conventional scheme's steady state value is the same as the unconstrained response, while the sum of the integrated scheme's steady state or trim values can be seen to be greater than the conventional scheme. This is the potential drawback of the integrated scheme, which may not be desirable in certain applications.

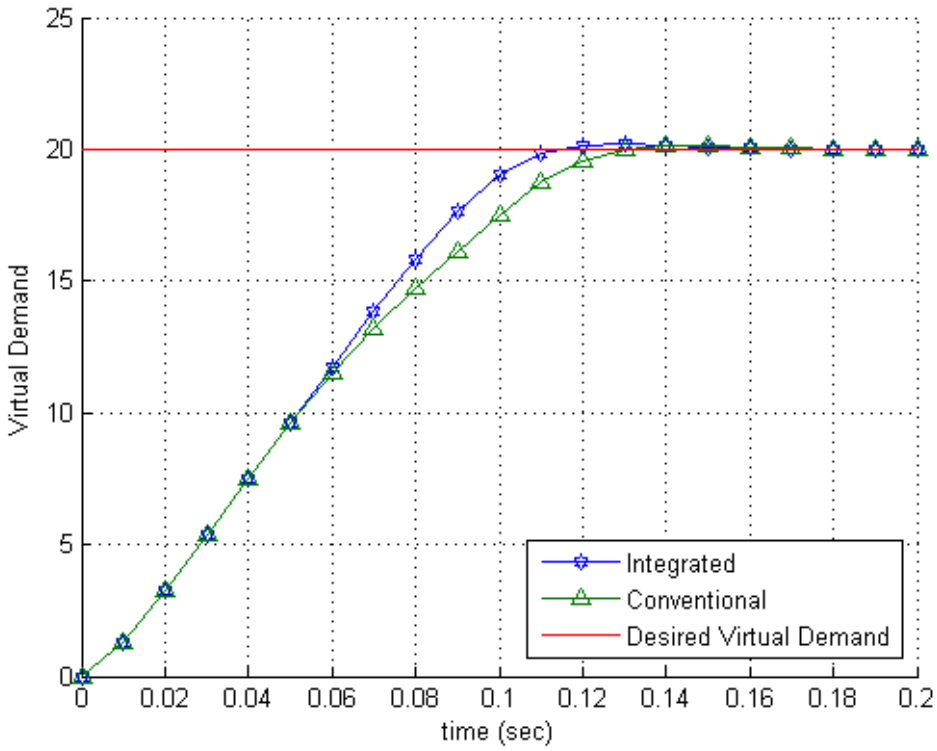

Figure 4.19: Simulation results showing the virtual demand response and actuator positions for both control schemes with input control saturation constraints. The integrated scheme can be seen to achieve the desired virtual demand response faster compared to the conventional scheme.

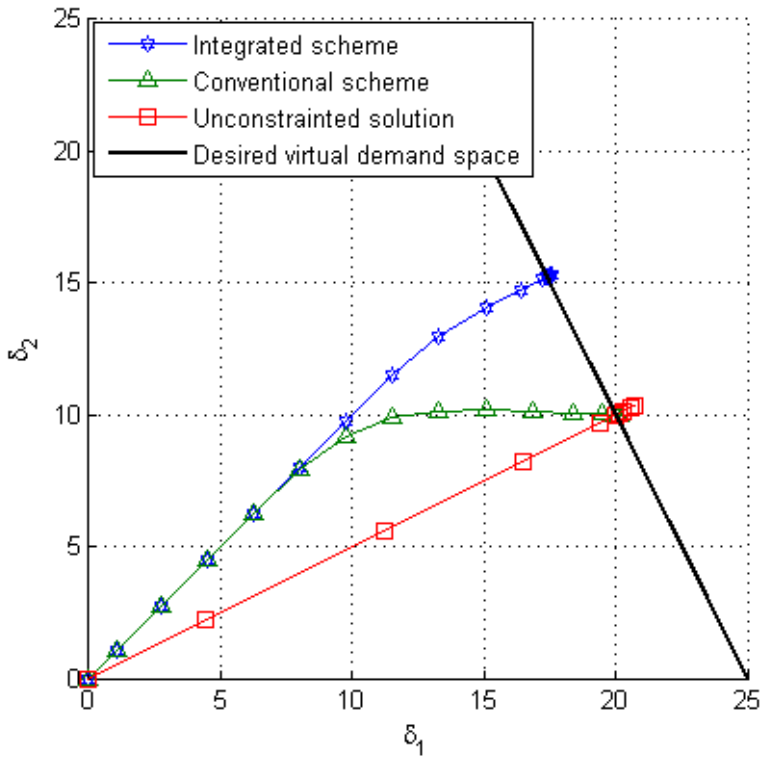

Figure 4.20: Simulation results showing the two actuator positions relative to the virtual demand space. The blue line shows the integrated scheme updating perpendicularly to the black virtual demand line after desaturating, while the green line shows the conventional scheme updating the actuator positions towards the desired virtual demand point similar to the unconstrained version.

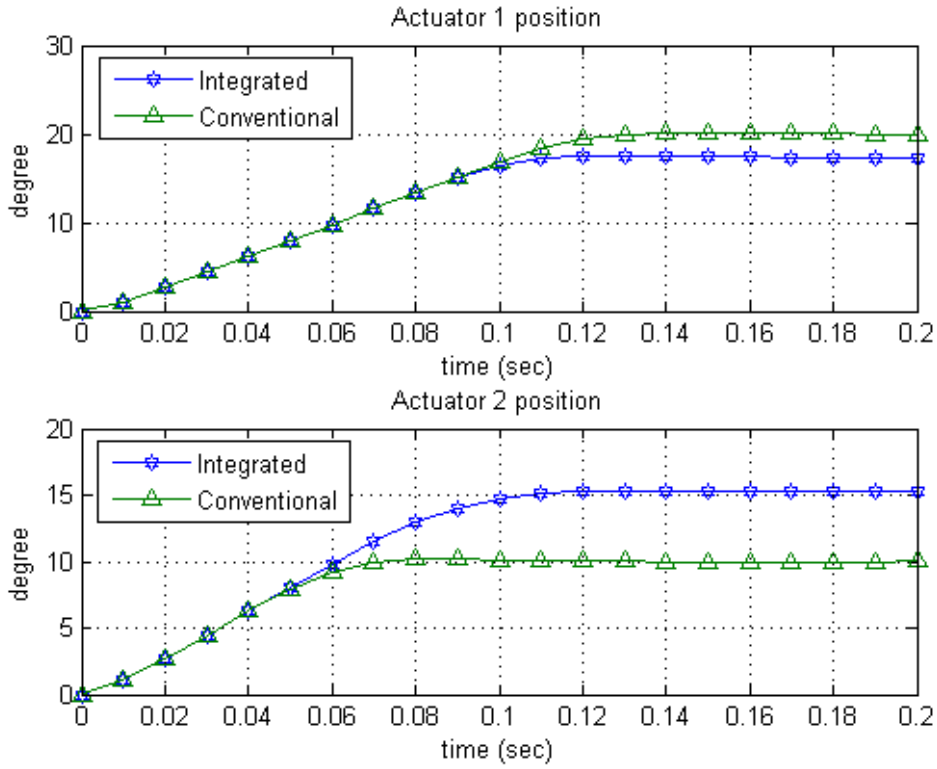

Figure 4.21: Simulation results showing the actuator position responses for both control schemes. The conventional scheme's steady state value is the same as the unconstrained response, while the sum of the integrated scheme's steady state values can be seen to be greater than the conventional scheme. This is the potential drawback of the integrated scheme.

4.4.4.3 Weighted control scheme simulations

Next, simulations showing the two integrated possible solutions to reduce the steady state actuator values of the integrated scheme are shown. One possible method to allow flexibility to the designer is to blend the two control schemes with a weighting function as described in Eqn. 4.47 to tune between speed and reducing trim values to meet system requirements. The phase plot in Fig. 4.22 illustrates the concept for the same simulation setup, with equal weights between the two control schemes with $\alpha = 0.5$. The path taken by the weighted solution is intermediate between the two control schemes. When the weighted solution closes towards the desired virtual demand, the conventional scheme update within it dominates to move the actuators to the unconstrained desired virtual demand point, which can also be viewed in Fig. 4.24. This alleviates the drawback of the larger actuator trim values experienced by using only the integrated scheme, at the expense of some rise time as seen in Fig. 4.23.

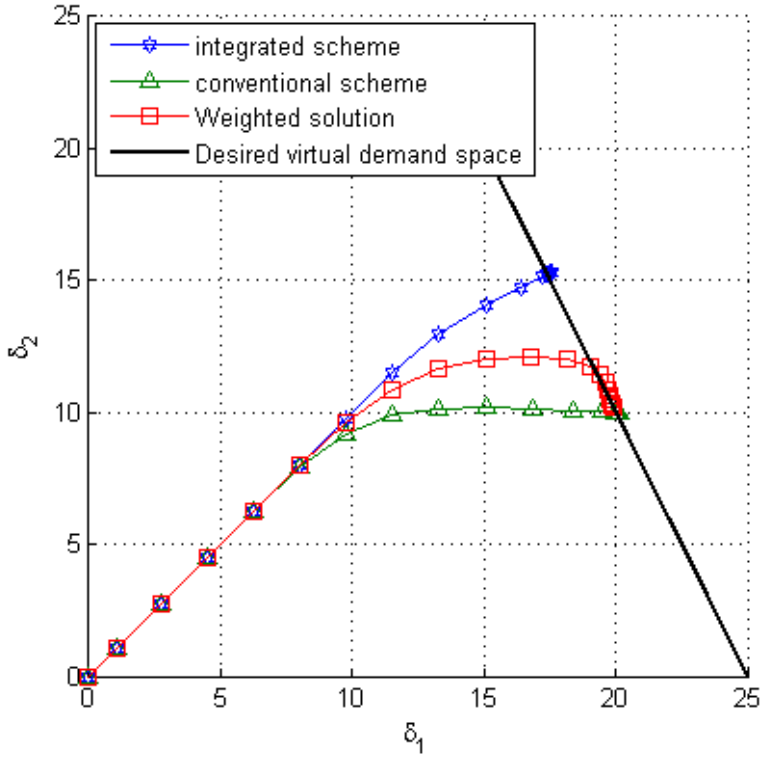

Figure 4.22: Simulation results showing the phase plot comparison between the integrated, conventional and weighted solution. The weighted solution offers a possibility in trading between performance and reducing the actuator steady state values.

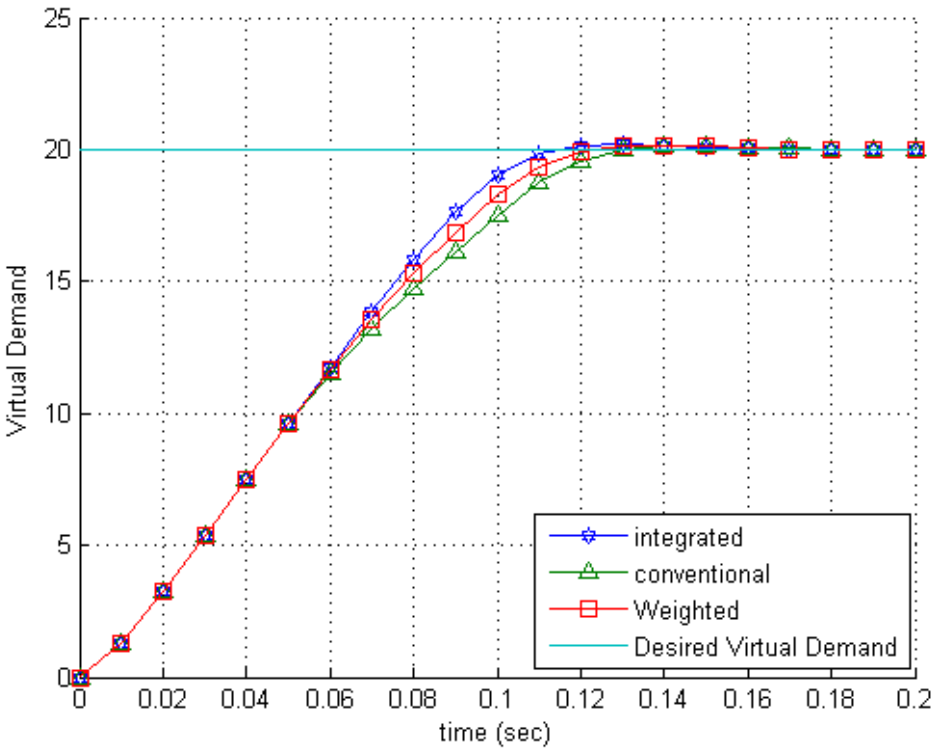

Figure 4.23: Virtual demand step response for the integrated, conventional and weighted solution. The weighted solution's rise time is seen to be faster than the conventional yet slower than the integrated scheme.

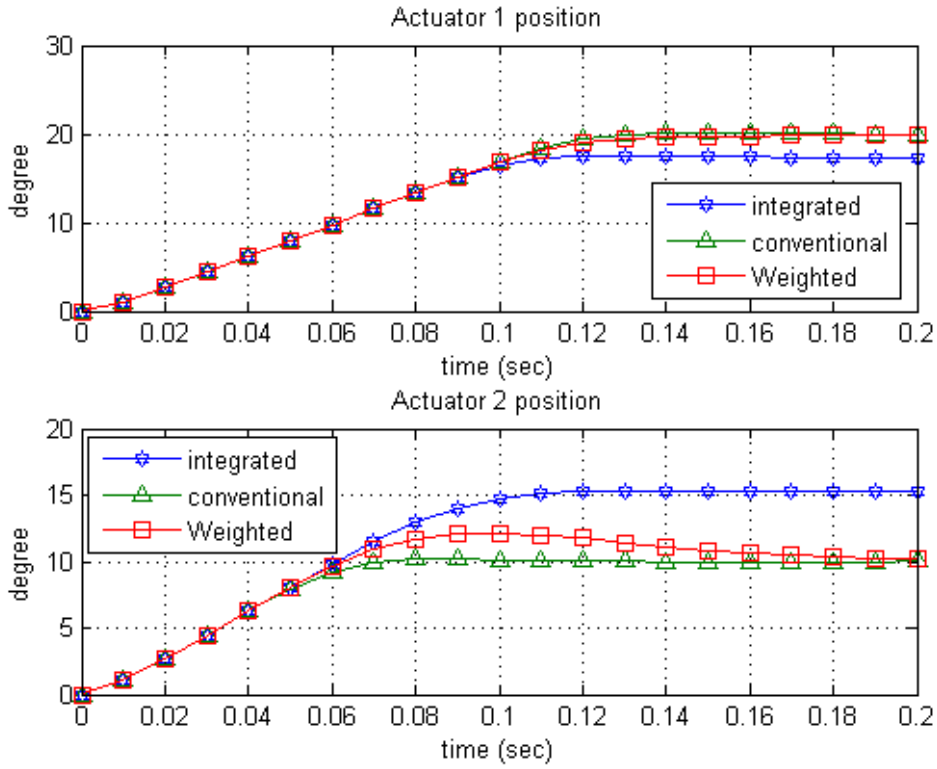

Figure 4.24: Simulation results showing the individual actuator responses for the weight solution, with the integrated and conventional schemes for comparison: The weighted solution is seen to converge to the conventional steady state values. This verifies the integrated weighted solution alleviates the larger steady state (trim) values of the integrated scheme.

4.4.4.4 Two step control scheme simulations

Fig. 4.25 show the actuator trajectory for the second solution to alleviate the larger trim values in the integrated scheme. Here, we can see the two-step solution at work. The first step of the solution is to use the integrated scheme to reach the desired virtual demand in the fastest manner. For the simulation, it is done by checking for the condition when the absolute virtual demand error is less than 0.1. When this is fulfilled, the conventional scheme is deployed to bring the actuators positions to the unconstrained solution. Fig. 4.26 verifies the virtual demand response of the solution to be equal to the integrated scheme, while Fig. 4.27 shows the actuators decreasing to the conventional scheme's steady state values by the end of the simulation.

Overall, the two potential solutions are verified to work. They present flexibility in the choice of tuning options for the control engineer to meet a control system's requirements.

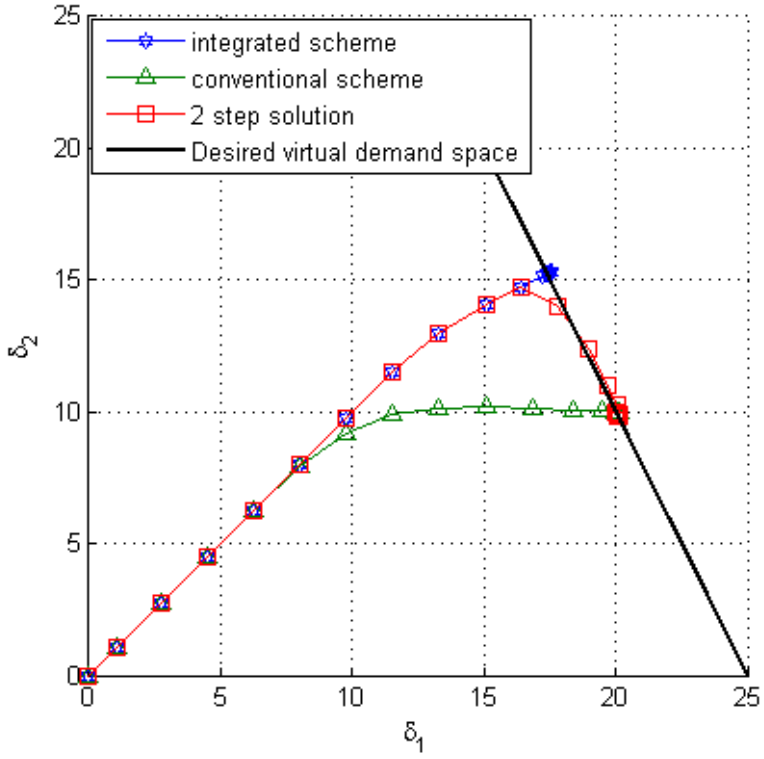

Figure 4.25: Simulation results showing the relative actuator trajectories of the two-step solution integrated to achieve the same steady state values of the conventional scheme with the response of the integrated scheme: From the plot, the two-step solution uses the integrated scheme's control to reach the desired virtual demand before switching to the conventional scheme's control to achieve the minimal actuator steady state values.

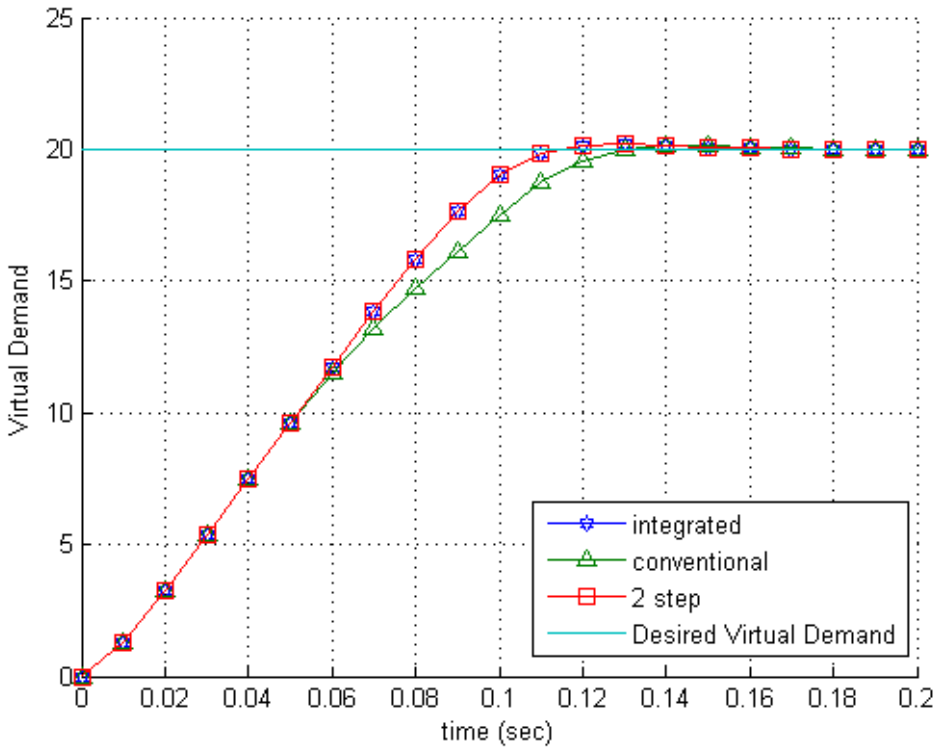

Figure 4.26: Simulation results showing the virtual demand response of the two-step solution to alleviate the larger steady state actuator values for the integrated scheme: Here, the proposed two-step solution is seen to preserve the same virtual demand rise time as the integrated scheme.

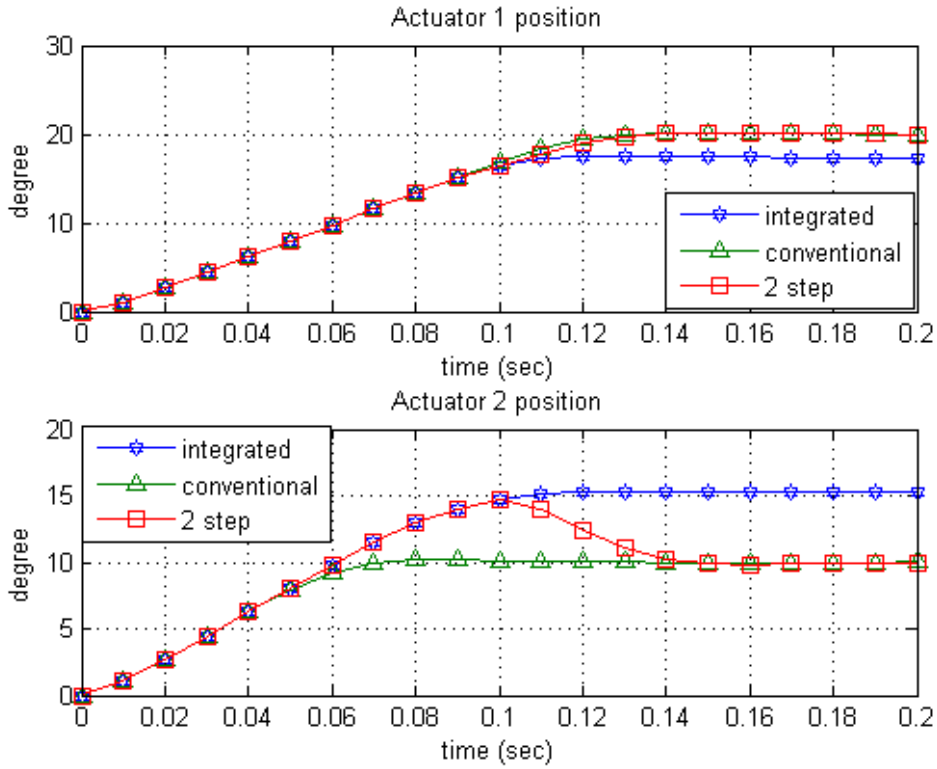

Figure 4.27: Simulation results showing the actuator positions of the two-step solution. The final actuator values of the two-step solution is seen to be the same as the conventional scheme.

4.4.4.5 An tail fin controlled UAV example

A practical simulation example of a four tail fin UAV example from Ref. [30] is illustrated here. Previous work on this example using frequency domain error analysis proved the magnitude of the error dynamics of the integrated scheme is smaller than the conventional scheme. Using the new geometrical interpretation results presented in the previous section, an alternative perspective on the simulation results is presented here.

The system consists of three virtual demand inputs δ_p, δ_r and δ_y which represents the control systems pitch, roll and yaw moment demands. The control allocation matrix $\mathbf{B_{ca}}$ to relate this to the actuator commands is

$$\mathbf{B_{ca}} = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 1 & 1 \\ 1 & -1 & 1 \\ 1 & -1 & -1 \end{bmatrix} \tag{4.51}$$

while the control deallocation matrix $\mathbf{P_{ca}}$ is

$$\mathbf{P_{ca}} = \frac{1}{4}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ -1 & 1 & 1 & -1 \end{bmatrix} \tag{4.52}$$

With four actuators and three virtual demands, and with $\mathbf{B_{ca}}$ having full column rank, there is still one available analytical redundancy in the system to be exploited. The individual actuator model and parameters used in this simulation is identical to the previous simulation.

First, the actuator position output constraint is removed to observe the effect of control input saturation. A step change of 10 deg in desired virtual demand in all three axes acts is simulated. In Fig. 4.28, it can be seen the integrated scheme is faster than the conventional scheme in reaching the setpoint for all three virtual demands. To achieve this desired virtual demand for the conventional scheme, Actuators 1 to 4 would have actuator commands of 10, 30, 10 and -10 degrees respectively obtained from $\mathbf{B_{ca}}\delta_{\mathbf{mc}}$. Thus, Actuator 2 control input constraints would be the limiting factor in achieving the desired virtual demand. Fig. 4.29 shows the individual actuator position movements to achieve the desired virtual

demand. Looking at the integrated scheme's actuator responses, it can be seen that the other actuators' responses have been increased in magnitude to alleviate Actuator 2's demand. This alleviation results in an overall faster response.

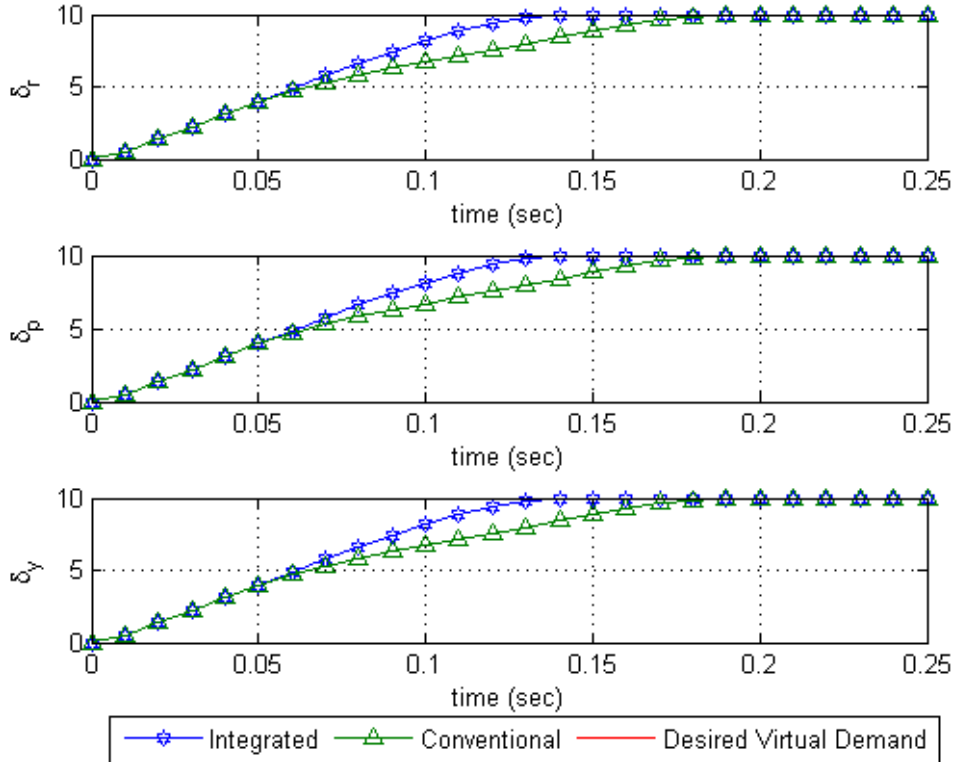

Figure 4.28: Simulation results of a four actuators with three virtual demands actuation system with control input constraints. The integrated scheme is seen to be faster in all three virtual demand response when compared with the conventional scheme.

The next simulation imposes an additional actuator position limit of 20 deg on each actuator. Fig. 4.30 shows the virtual demand responses with the actuator constraint. Comparing it with Fig. 4.28, the conventional scheme has a steady state error caused by the actuator position constraint, while the integrated scheme is able to reduce the steady state error. Fig. 4.31 shows the Actuator 2 at its position constraint, and the integrated scheme is able to orthogonally update the control signal for the other actuators to iteratively reduce the virtual demand error. Using the 2-norm of the virtual demand error at the end of simulation time of 0.25*sec* as an indication of the difference in steady state error, the integrated scheme achieved a far lower 0.2 compared with 4.3 for the conventional scheme due to the usage

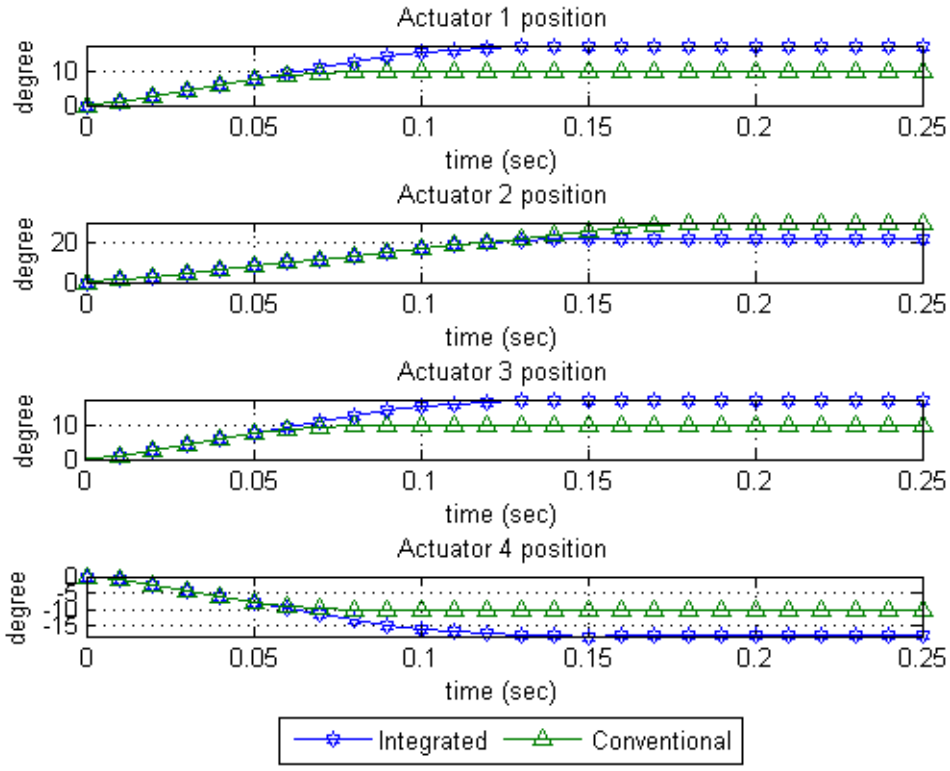

Figure 4.29: Simulation results of a four actuators with three virtual demands actuation system with control input constraints. The magnitude of Actuator 2's position is seen to be larger compared to the other three actuators for the conventional scheme. This slows down its overall virtual demand response as seen in the earlier figure. However, this effect is reduced for the integrated scheme, as the other actuators compensate for the input saturation.

of the available analytical redundancy in the system.

So far, the simulations presented are with static step desired virtual demands. Next, a varying desired virtual demand is studied. Letting the virtual demand to be tracked be a sinusoidal signal of magnitude 10 deg and frequency of $1Hz$, Fig. 4.32 shows both control schemes tracking the virtual demands well when unsaturated.

However, when the virtual demand frequency is increased to 2 Hz, the conventional scheme's performance degrades more significantly in comparison with the integrated scheme as shown in Fig. 4.33. In the figure, the conventional scheme's response peak magnitude decreased by about 28% of the virtual de-

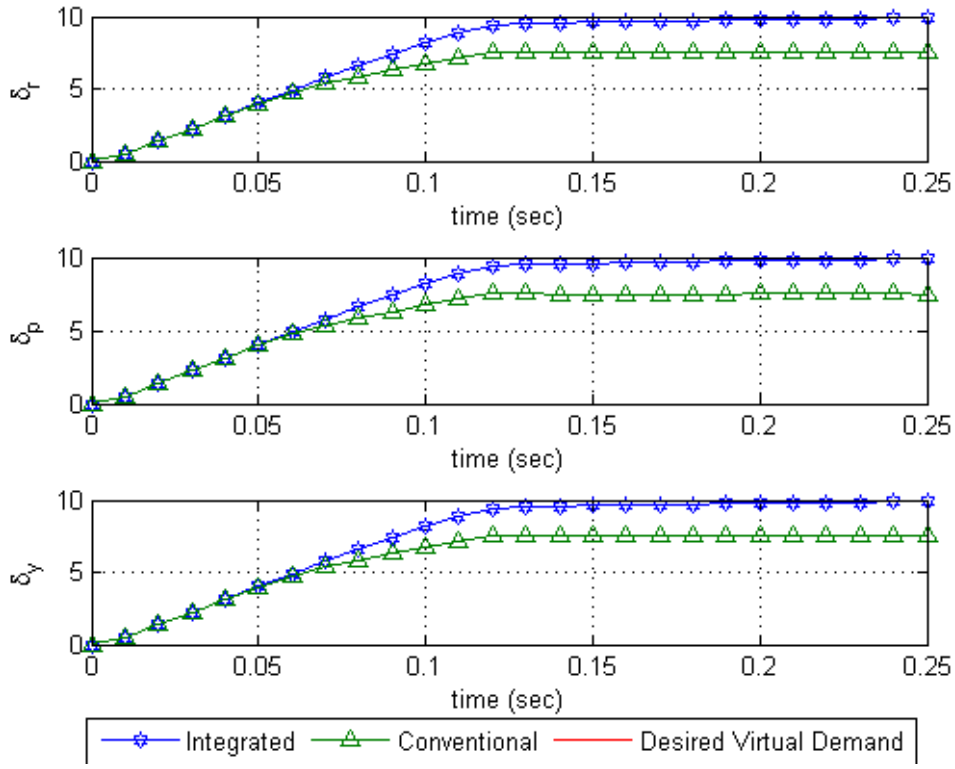

Figure 4.30: Simulation results of a four actuators with three virtual demands actuation system with control input and actuator position constraints. The integrated scheme is able to achieve the desired virtual demand response, unlike the conventional scheme.

mand compared to a drop of 8% for the integrated scheme. The reduction in clipped magnitude may be explained by the reduction in clipping of the input saturation due to the change in control signal update direction in the integrated scheme, which alleviates the saturation effect as graphically explained in Fig. 4.6. This provides an alternative perspective to the improvement in performance of the integrated scheme over the conventional scheme, which has been analysed in the author's previous work [89], which showed the magnitude of error dynamics to be smaller with the integrated scheme.

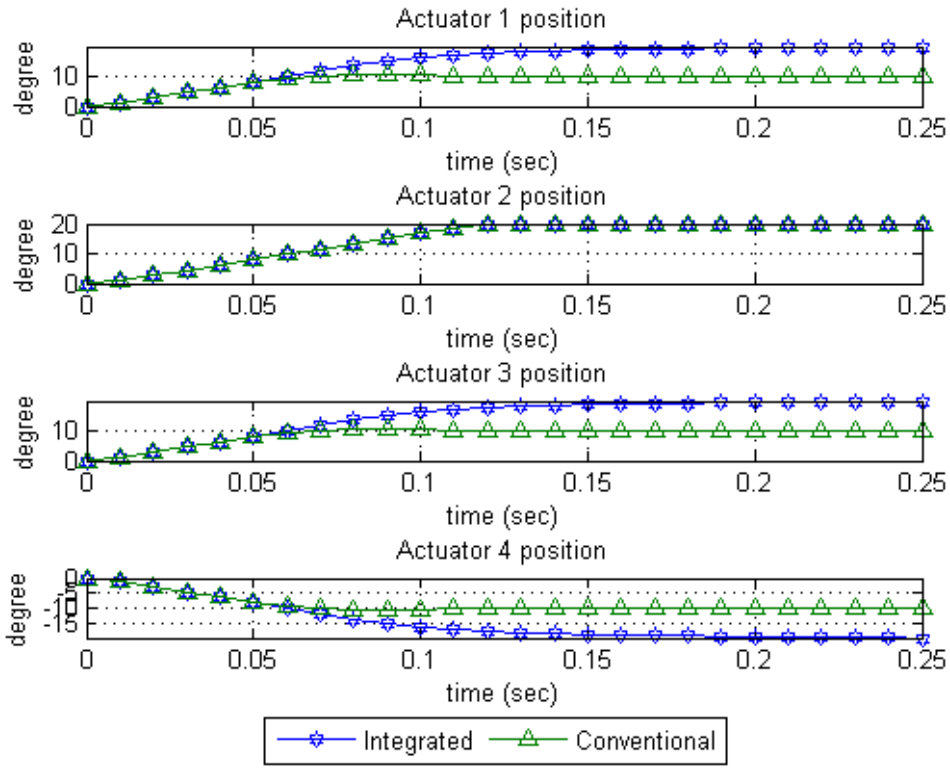

Figure 4.31: Simulation results of a four actuators with three virtual demands actuation system with control input and actuator position constraints. With a 20 deg limit on the actuators' position, the integrated scheme increases the magnitude of the other actuators to achieved the desired virtual demand shown in the earlier figure. In contrast, the conventional scheme has a steady state virtual demand error due to Actuator 2's position saturation shown here.

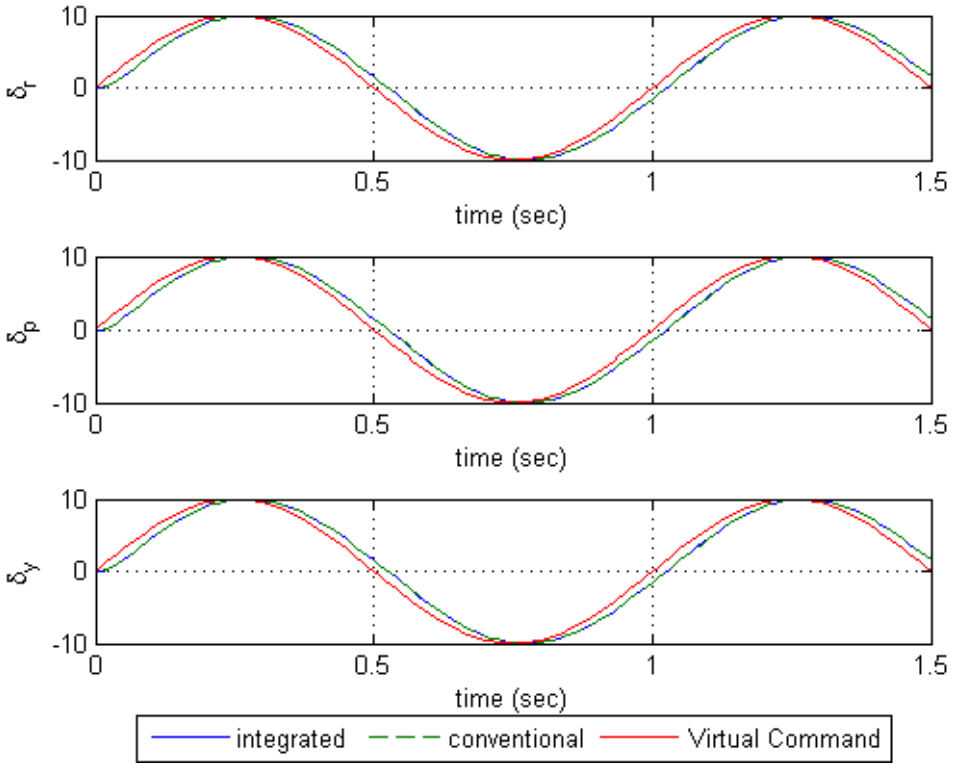

Figure 4.32: Simulation results showing similar responses with a sinusoidal virtual demand of 1Hz frequency. This plot shows the tracking performance of the two control schemes to be similar with less demanding virtual demand reference signal.

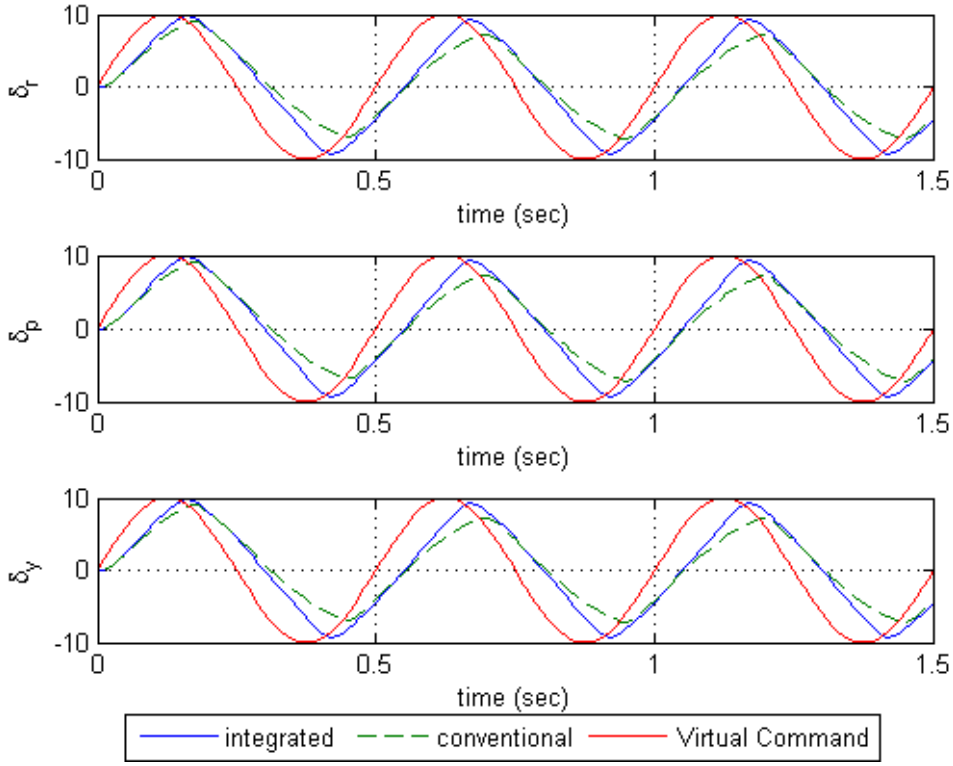

Figure 4.33: Simulation results illustrating the improvement in performance of the integrated control scheme when the virtual demand is changing. With an increased in virtual demand reference signal's frequency, the conventional scheme's performance degrades more significantly when compared to the integrated scheme.

4.5 Conclusion

This chapter presents a theoretical discrete domain analysis of an integrated control allocation and actuator control scheme. The integrated scheme integrates a pseudo-inverse control allocation with the actuator controller. This allows for the integrated scheme to improve on its performance when compared to a conventional control scheme.

The integrated control scheme is shown to be equivalent to a conventional control scheme in the virtual demand space when operating linearly. However, the integrated scheme is shown to update the actuators' control actions orthogonally to the desired virtual demand space. This results in the integrated scheme's potentially faster performance as compared to a conventional control scheme when both are subjected to fault saturation. Geometrical interpretations are shown, and conditions when the two control schemes are similar are given. A drawback of the integrated scheme is that it may require larger actuator demands over the conventional scheme. Two possible solutions to alleviate this drawback are suggested. Simulation examples in the chapter illustrated the theoretical analysis provided.

Chapter 5

Application of an integrated actuator control scheme to improve the Aerosonde UAV's fault tolerance

5.1 Introduction

In the previous chapter, an integrated actuator control scheme has been analysed, and simple initial simulations have shown promising results. In this chapter, the integrated scheme is applied to an Aerosonde UAV, which has one of its actuators stuck. This provides a complete system level analysis, and point to the areas where the application of the integrated scheme has the most impact on.

The analysis first starts with the assumption the flaps on the Aerosonde UAV are not available for control purposes. A simple right aileron fault effect is first presented. Next, the analysis is shifted to a right ruddervator stuck case. The challenge of this problem is the lack of adequate actuator redundancy to fully handle the entire position range of a stuck ruddervator actuator. Nevertheless, increases in the capability of the system to handle a wider stuck range would always be desired. The analysis is then extended to include the case with usage of flaps to evaluate whether a slight increase in control authority improves the fault tolerance of the system. In the simulations, it is evident the integrated scheme improves the ruddervator angle stuck range compared to the conventional scheme. It also improved the results of an active reconfigurable scheme performance in Ref. [7] by 23%.

Furthermore, better guidance tracking performance was obtained using the integrated scheme through the alleviation of the fault degradation by the integrated scheme. The integrated scheme may appeal as an alternative method to improve on actuator fault tolerance for systems when the ability to increase actuator redundancy is constrained. A guidance scheme consisting of a path generator and path following algorithm to follow a path is next included. The integrated

scheme's is shown to improve the guidance performance due to the improvement in flight performance with the stuck actuator faults.

The organization of this chapter is as follows. First, simulations are conducted with an Aerosonde UAV model using first with a classical flight controller experiencing aileron and ruddervator faults. Two sets of simulation results are shown comparing the differences between the usage of the integrated actuator control scheme and conventional actuator control scheme; one without the usage of flaps and the other with the flaps usage to assess the availability of control authority has post fault occurrence. Next, a comparison between the schemes on their fault handling range to the ruddervator fault is presented. To complete the overall system level analysis, a nonlinear guidance scheme is incorporated into the simulation setup to examine the guidance performance on a path following mission to further examine the impact of using the integrated scheme. Lastly, conclusions are presented.

5.2 Demonstration with flight control

In this section, the integrated scheme is applied onto the Aerosonde UAV model detailed in Ref [7] to evaluate its impact on the UAV suffering from a stuck right ruddervator fault. The Aerosonde UAV has two ailerons, two ruddervators and two flaps for actuation purposes as shown in Fig. 5.1. Conventionally, the ruddervators are predominant yaw and pitch moment generators, while the ailerons are used for the roll moment generators. The flaps are usually used during takeoff and landing phases to provide additional lift to the UAV. Due to the symmetry in positioning of the ailerons, a single aileron stuck fault may be accommodated due to the availability of control authority from the remaining healthy aileron. On the other hand, the UAV has limited capability to accommodate a large ruddervator fault due to the lack of remaining control authority to compensate for the fault induced moments and control of the vehicle.

Figs. 5.2 and 5.3 show the simulation setups for the integrated and conventional schemes implementation. A classical longitudinal and lateral autopilot is used to track the airspeed, height and roll reference commands. The airspeed and roll autopilots are designed with an inner rate damping control loop and outer

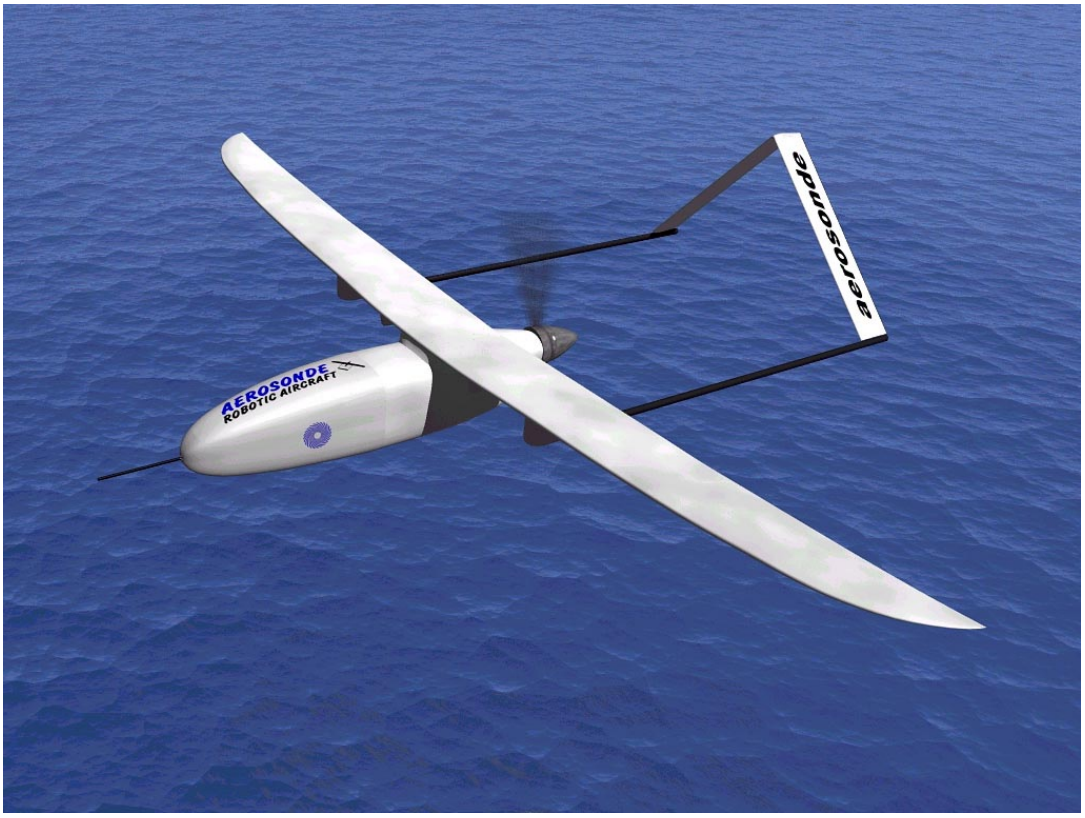

Figure 5.1: Aerosonde UAV model under study; picture from https://commons.wikimedia.org/wiki/File:Aerosonde.jpg.

reference command control loop, while the height control loop uses a Proportional Integral (PI) controller. Turn coordination compensation is achieved by adding the associated kinematic relation rate commands into the pitch and yaw rate loops described in [80]. The sensors assumed available for flight control purposes are the body angles, body angular rates and accelerations, airspeed and height, which have been modelled as a time delay of 0.01 sec. The sensors output is filtered by a first order low pass filter before being used in the autopilots.

In this assessment, two simulation configurations are analysed. First, simulations are conducted without the usage of flaps for control. This configuration may appeal in cases when the flaps are not available or desirable. This is further evaluated in the next simulation configuration, where the flaps are utilized for fault handling, to assess the potential improvement in performance with increased control authority on the difficult ruddervator fault problem.

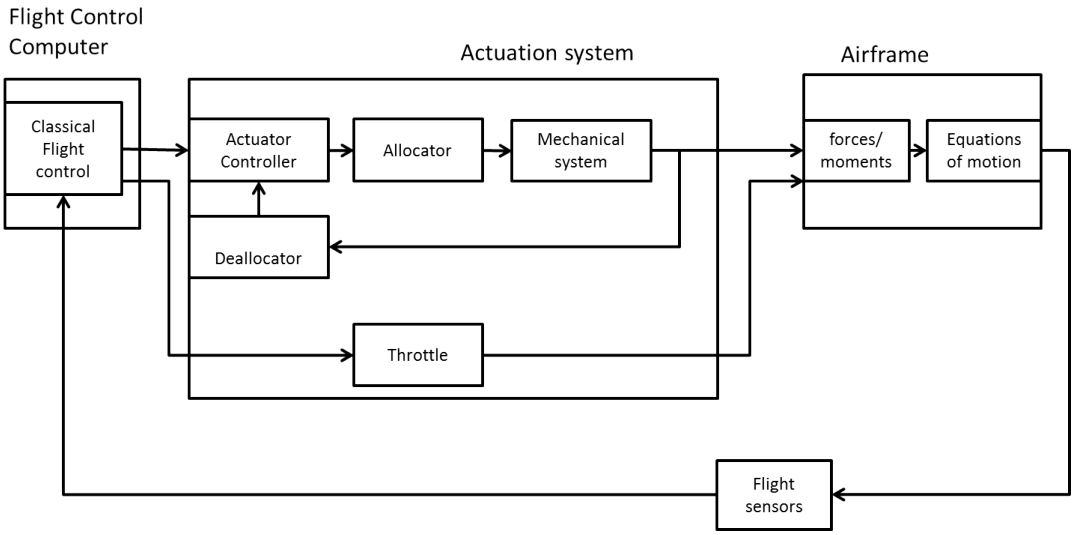

Figure 5.2: Simulation block diagram for the integrated scheme.

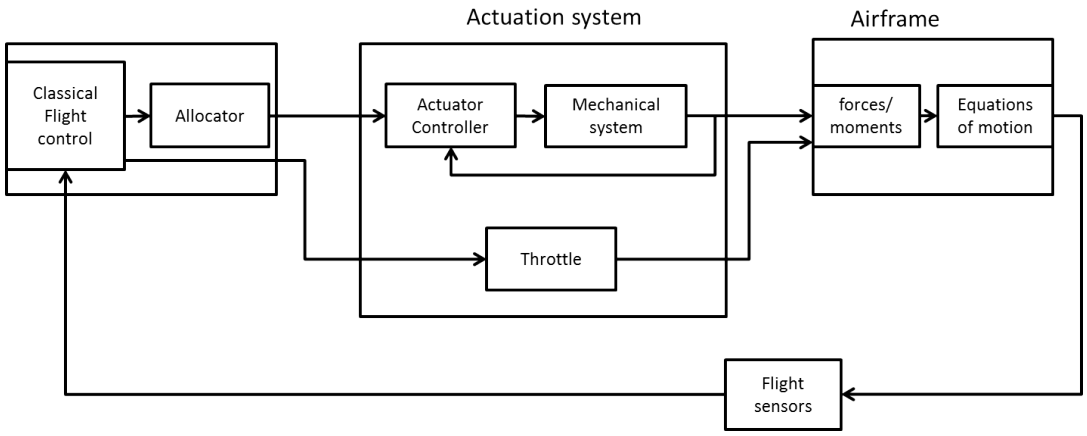

Figure 5.3: Simulation block diagram for the conventional scheme.

5.2.1 Simulations Results

5.2.1.1 Aileron fault

First, a severe right aileron fault of 16° is simulated. Figures 5.4 to 5.6 show the simulation results with the fault injected at 5 seconds into simulation.

As there are two actuators (left and right ailerons) to generate the desired roll moments, the remaining healthy aileron would be able to null the effects of the faulty aileron. Fig. 5.4 shows the outer loop responses. The airspeed and height responses are similar using either of the schemes. This is due to the flight control

tracking and regulation, which minimizes the fault degradation. While the roll response at the largely similar, the integrated scheme regulated the roll response better than the conventional scheme. The roll response perturbation due to fault with the integrated scheme is within 2°, while the conventional scheme's roll response went to −8° due to the fault occurrence at simulation time of 5 seconds. The better regulation is due to the integrated scheme handling the fault at the faster actuator control loop.

Fig. 5.5 plots the angle of attack and sideslip angle incurred by the two schemes. Here,both the angle of attack and sideslip angle plots show the two schemes to be similar, except for the sideslip angle at the point of fault occurrence. The sideslip angle for the integrated scheme at the point of fault injection is slightly smaller than the conventional scheme. This is due to the faster regulation by the integrated scheme.

Fig. 5.6 shows the actuator responses by the two control schemes. The two schemes actuator responses are largely similar, except for sharp left aileron response at the fault occurrence in the integrated scheme which arrests the roll angle variation faster compared to the conventional scheme.

When there is remaining control authority remaining in the system, there is little difference between the two control schemes as the flight control will regulate and minimize any deviation from its nominal response. The integrated scheme would be slightly faster in regulating the fault degradation, as it regulates the fault from the faster actuator control loop compared to the outer (and slower) roll control loop in the conventional scheme.

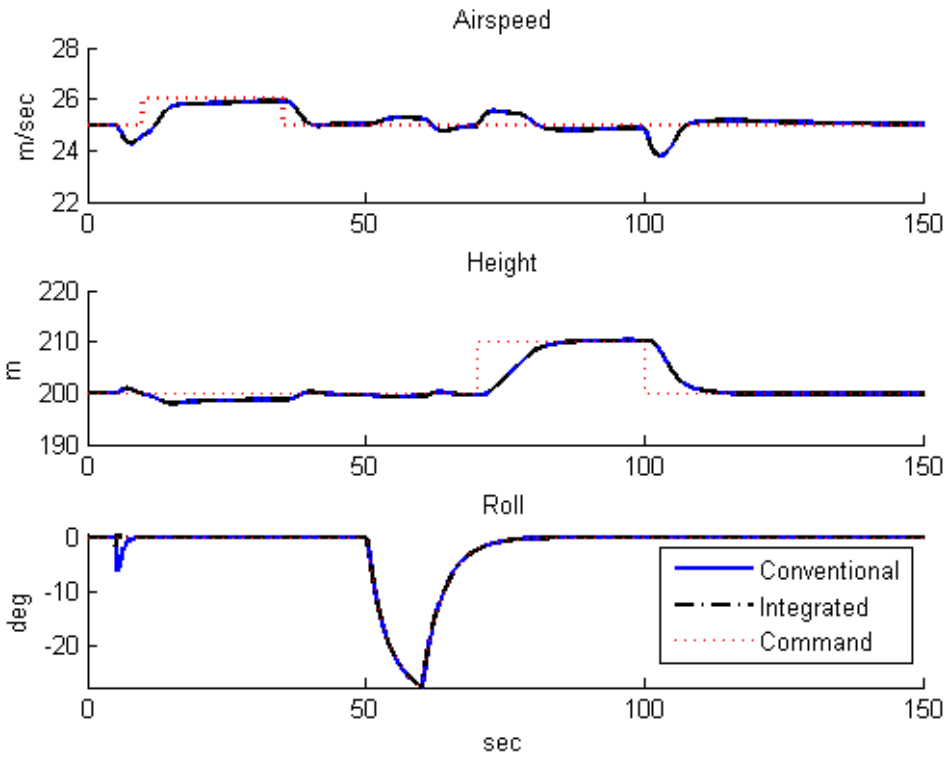

Figure 5.4: Outer loop control responses showing the similar tracking responses between the integrated and conventional schemes with a 16° right aileron fault. They are largely similar due to the healthy left aileron providing the control authority to compensate for the fault. However, the integrated scheme is slightly faster in the roll angle recovery as it handles the fault at the inner actuator control loop.

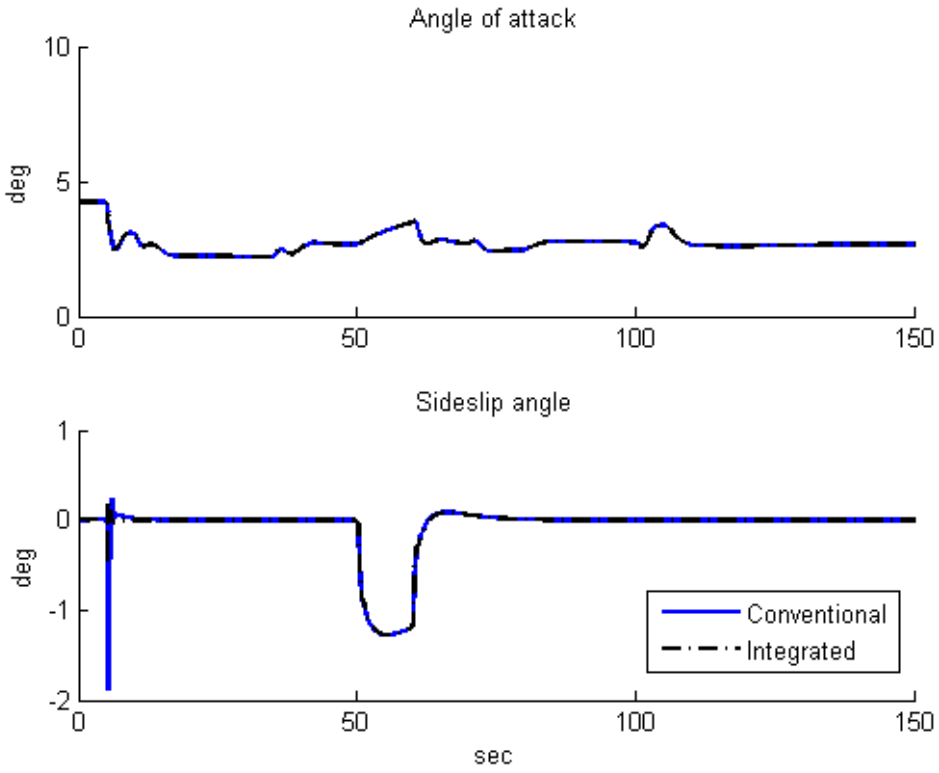

Figure 5.5: Angle of attack and sideslip angle comparison between the conventional and integrated schemes with aileron fault without flap usage.

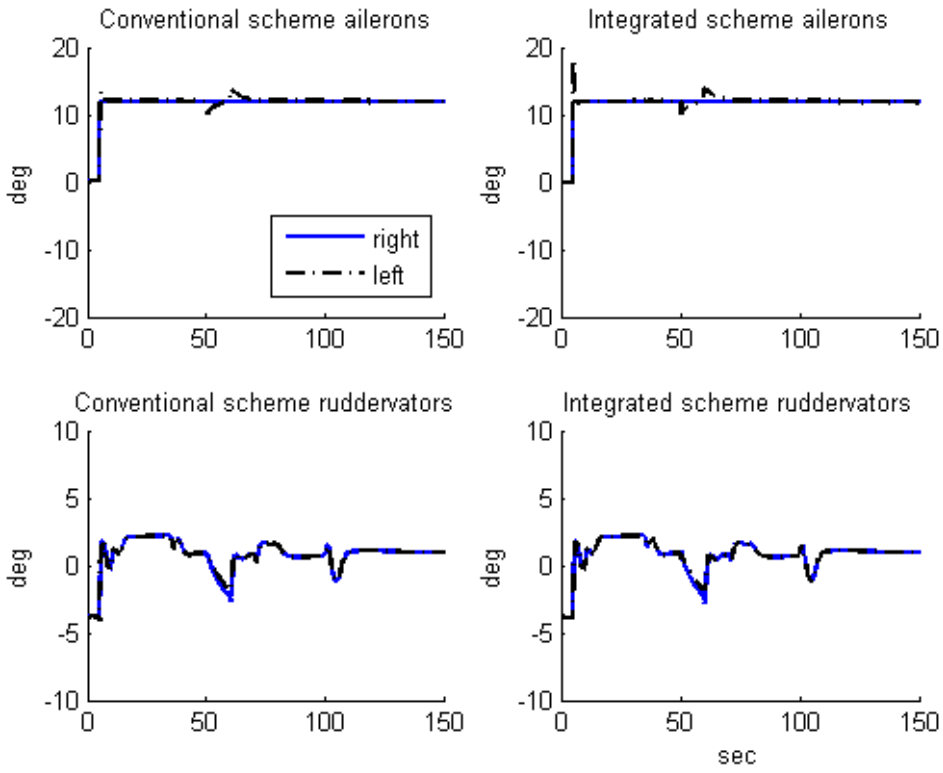

Figure 5.6: Actuator responses showing the different actuator positions adapted by the two control schemes. The plots on the left show the conventional scheme's while the right plots show the integrated scheme's. The two scheme's responses are largely similar, except for the left aileron response at the fault injection time of 5seconds for the integrated scheme to better compensate for the fault.

Next, the flaps is included into the simulation setup to examine the system's behaviour when slightly more control authority and more degrees of freedom is presented. Here, the extreme limit of 18° right aileron fault is injected into the simulation at the 5^{th} second of the simulation. This is the extremal limit of the aileron. With the flaps included in the simulation, there is remaining control authority to handle the fault and possibly meet the desired virtual demands from flight control.

Fig. 5.7 shows the outer loop control responses for the simulation. The plots showed the responses of almost identical responses between the two schemes, except for the slightly smaller roll angle degradation by the integrated scheme compared to the conventional scheme. The reason for this faster response is the same as the previous simulation case: the integrated scheme by handling the fault at the inner control loop, is able to provide a faster response to handle the fault.

The main takeaway from conducting the aileron fault simulations is the similarity in performance between the integrated and conventional schemes when there is actuator redundancy in the system to adequately compensate for the fault. The results are very similar due to the identical outer loop controllers which minimize the outer loop tracking deviations. However, the integrated scheme allows for a slightly faster roll angle recovery at the aileron fault occurrence.

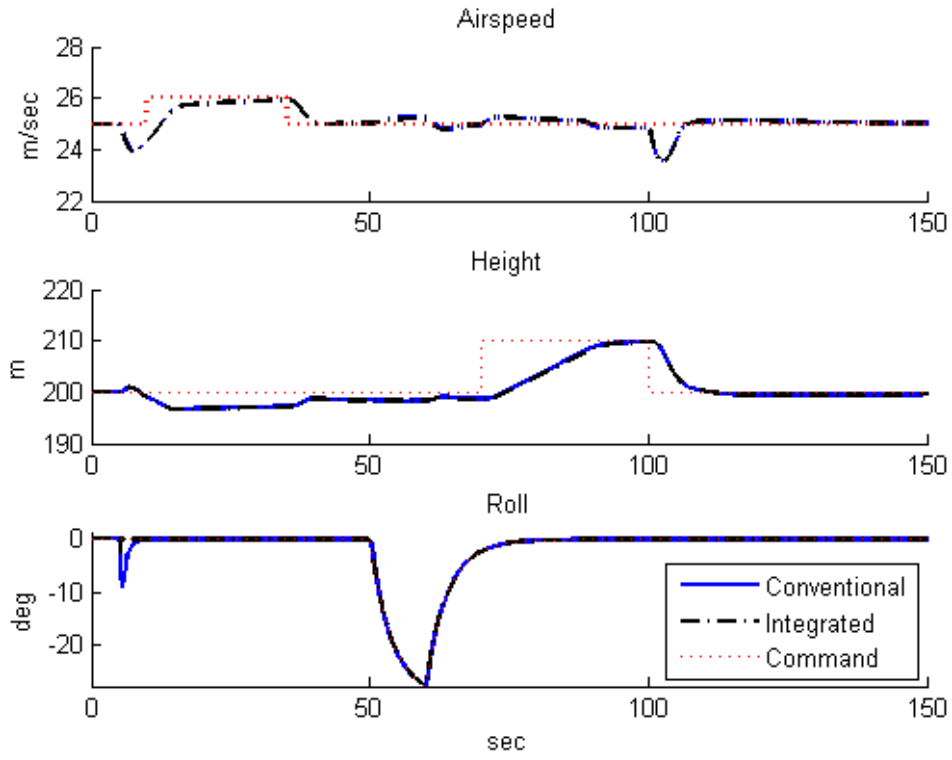

Figure 5.7: Outer loop control responses showing the similar tracking responses between the integrated and conventional schemes. They are largely similar due to the identical flight controllers in place.

5.2.1.2 Ruddervator fault

Next, the problem of handling a ruddervator fault is examined. This is a harder problem for fault accommodation, as normally the two ruddervators are dominant pitch and yaw axes moment generators. The reason is that with a ruddervator fault, there is only one remaining degree of freedom with the remaining healthy ruddervator to control two axes.

First, the simulations are conducted without the usage of flaps. Results are presented in Fig. 5.8 to 5.11, with the right ruddervator stuck fault occurrence of -5 deg at simulation time $5sec$. The reference profile consists of step changes in airspeed, height and roll to assess the capability of the system to follow the reference profile after fault occurrence. The roll command reference is filtered by a first order filter to allow for smooth transitions.

In Fig. 5.8, the conventional and integrated scheme's outer loop control re-

sponses are similar when subjected to the severe fault. This can be expected since the outer loop controllers for both schemes are identical. However, when one plots the actuator position responses in the simulations shown in Fig. 5.9 show different steady state fault allocation outcomes. For the integrated scheme, the remaining healthy left ruddervator steady state value lies near the faulty right ruddervator. This nulls the faulty ruddervator's fault induced yawing moment. This additional pitching moment is countered by the two ailerons, whose steady state values has been increased to provide the pitch up moment. In contrast, the nominal scheme nulls the fault induced pitching moment, while leaving the pitching moment to the ailerons.

Looking at the virtual demand errors, which can be calculated by computing the difference between the desired virtual demand from the autopilot and the generated moment by the control schemes in Fig. 5.10, it can be seen the integrated scheme has smaller virtual demand errors in the yaw and pitch moment planes compared to the conventional scheme. This improvement shows up in Fig. 5.11, which shows the angle of attack and sideslip responses of the UAV. It can be seen the integrated scheme managed to regulate the fault induced sideslip to near zero post fault, compared to the large induced sideslip angle by the conventional scheme. However, the angle of attack is larger than the nominal scheme's, due in part to the larger post fault trim control surface deflections. However, this reduces the aircraft state vector movement away from the nominal operating point due to the fault.

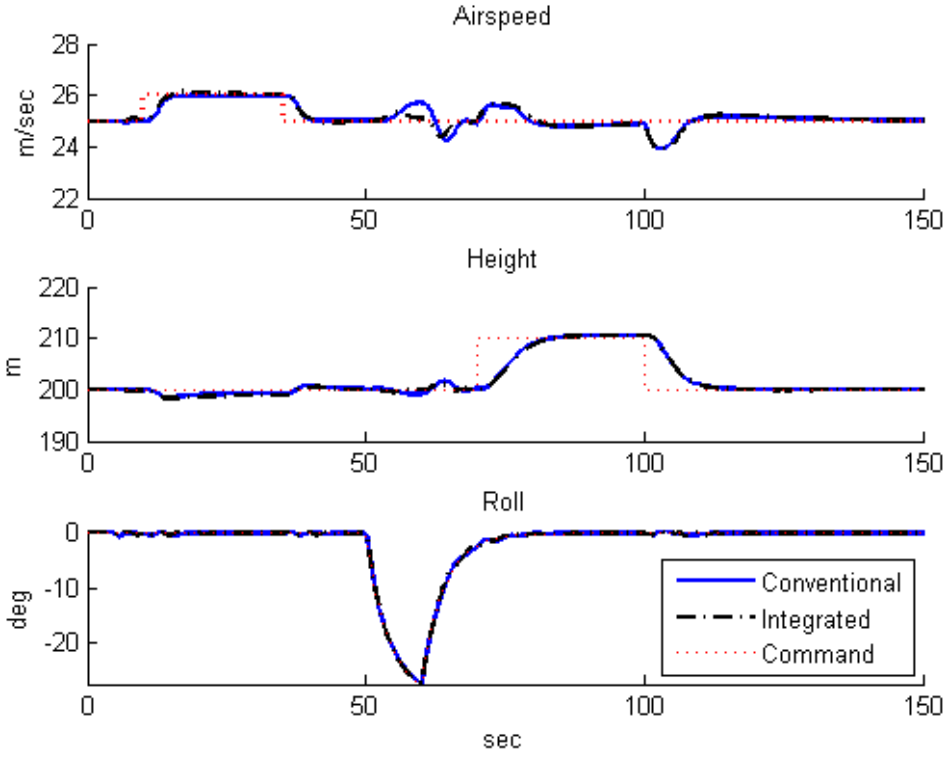

Figure 5.8: Outer loop control responses showing the similar tracking responses between the integrated and conventional schemes. They are largely similar due to the identical flight controllers in place.

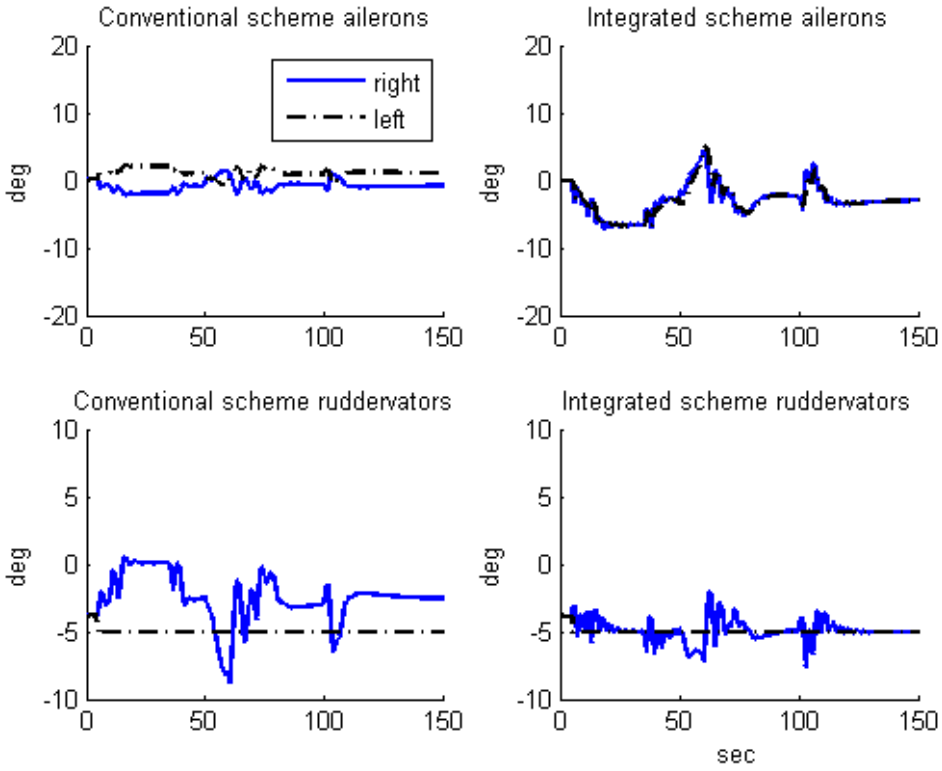

Figure 5.9: Actuator responses showing the different actuator positions adapted by the two control schemes. The plots on the left show the conventional scheme's while the right plots show the integrated scheme's whose post-fault trims are different.

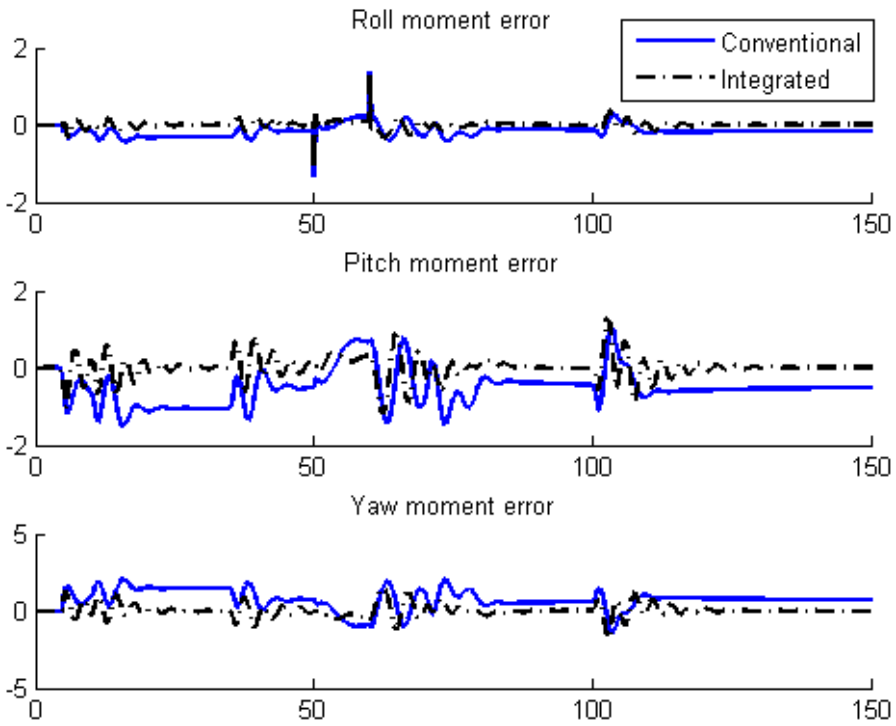

Figure 5.10: Virtual demand error in the roll, pitch and yaw planes. While the roll virtual demand error between the two schemes is similar, the integrated scheme is able to reduce the yaw and pitch moment virtual demands compared to the conventional scheme.

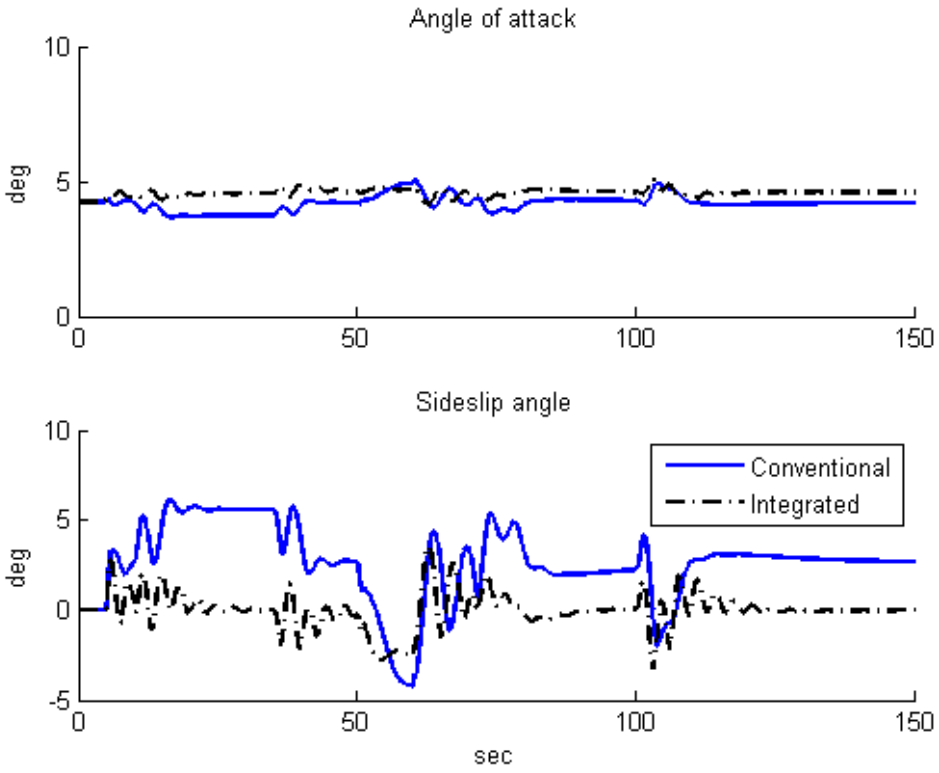

Figure 5.11: Angle of attack and sideslip angle comparison between the conventional and integrated schemes.

5.2.1.3 Ruddervator fault

Simulation results are presented in Figs. 5.12 to 5.15, with the right ruddervator stuck fault occurrence of -10 deg at simulation time 5 sec for the operaing point of $200m$ height and $25msec^{-1}$ speed. The reference profile consists of step changes in airspeed and height in the longitudinal plane, and a filtered roll command to assess the capability of the system to follow the reference profile after fault occurrence. Besides the usage of flaps in fault accommodation, the fault magnitude used for comparison here is doubled.

As shown in Fig. 5.12, the nominal and integrated scheme's outer loop control responses are similar when subjected to the severe fault. This can be expected since the outer loop controllers for both schemes are identical. However, in Fig. 5.13, the actuator position responses in the simulations show different steady state fault trim outcomes. For the integrated scheme, the remaining healthy left ruddervator steady state value lies near the faulty right ruddervator. This nulls the faulty ruddervator's fault induced yawing moment. This additional pitching moment is countered by the two ailerons, whose steady state values have been increased to provide the pitch up moment. The flaps are used in part to meet the manoeuvring demands. In contrast, the nominal scheme's healthy ruddervator trimmed the fault induced pitching moment, while leaving the yawing and roll moment demands to the ailerons and flaps.

Looking at the virtual demand errors, which can be calculated by computing the difference between the desired virtual demand from the autopilot and the generated moment by the control schemes in Fig. 5.14, it can be seen the integrated scheme has smaller virtual demand errors in the yaw and pitch moment planes compared to the conventional scheme. This improvement shows up in Fig. 5.15, which shows the angle of attack and sideslip responses of the UAV. It can be seen the integrated scheme managed to regulate the fault induced sideslip to near zero post fault, compared to the large induced sideslip angle by the conventional scheme. However, the angle of attack is larger than the nominal scheme's, due in part to the larger post fault trim control surface deflections. However, this reduces the aircraft state vector movement away from the nominal operating point due to the fault. The significance of this difference is elaborated in the next section when a guidance scheme to track a path is included into the simulation setup.

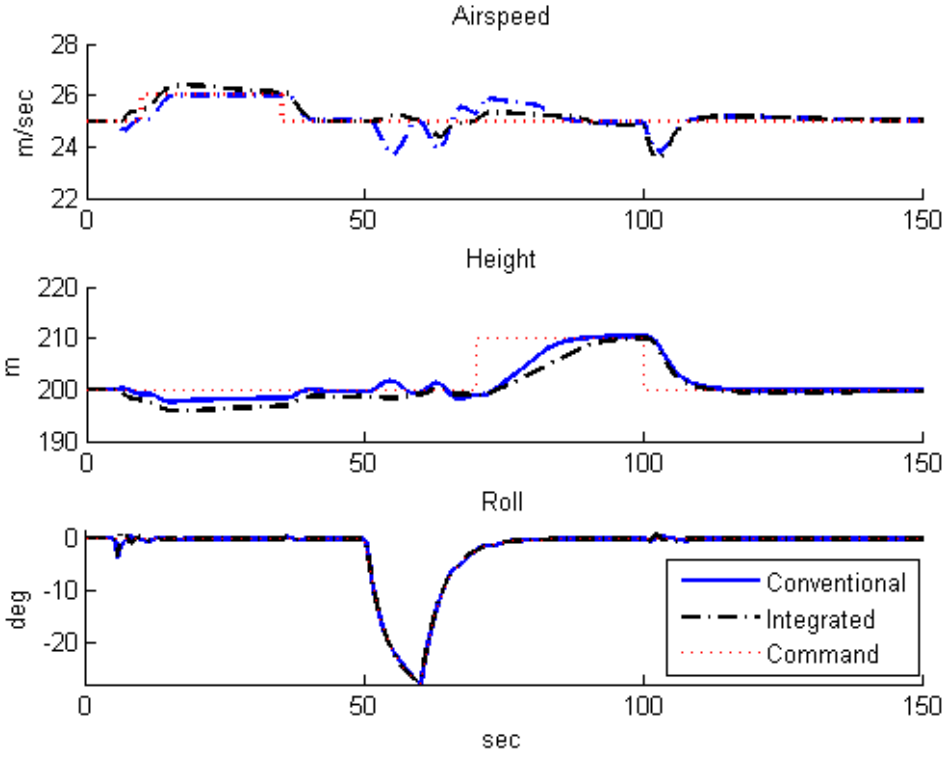

Figure 5.12: Outer loop control responses showing the similar tracking responses between the integrated and conventional schemes. They are largely similar due to the identical flight controllers in place.

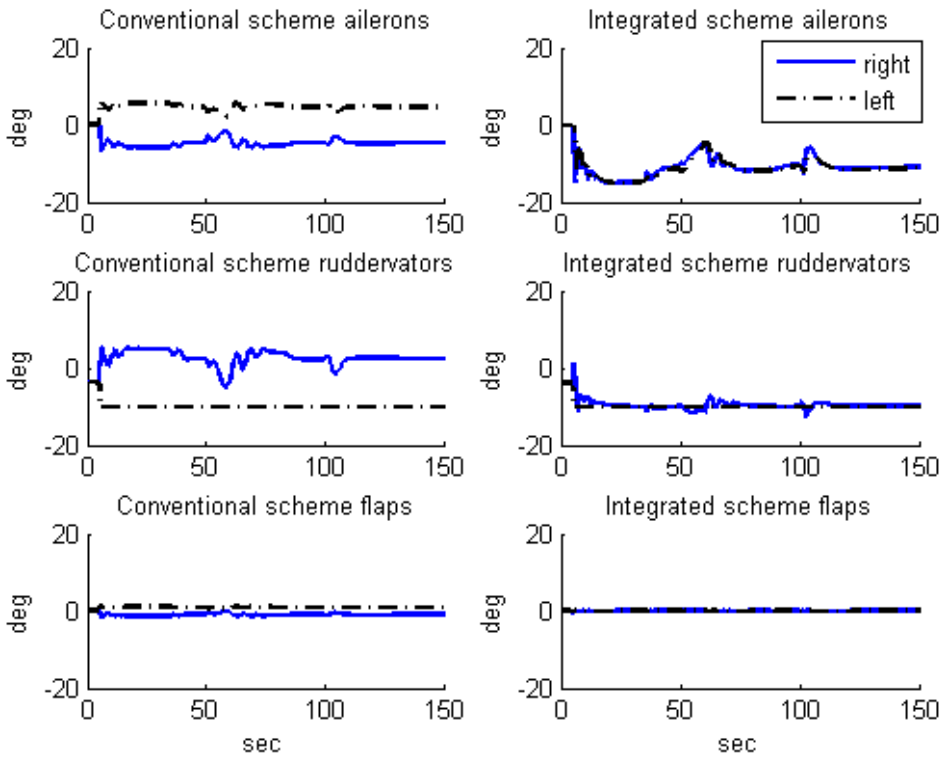

Figure 5.13: Actuator responses showing the different actuator positions adapted by the two control schemes. The plots on the left show the conventional scheme's while the right plots show the integrated scheme's whose post-fault trims are different.

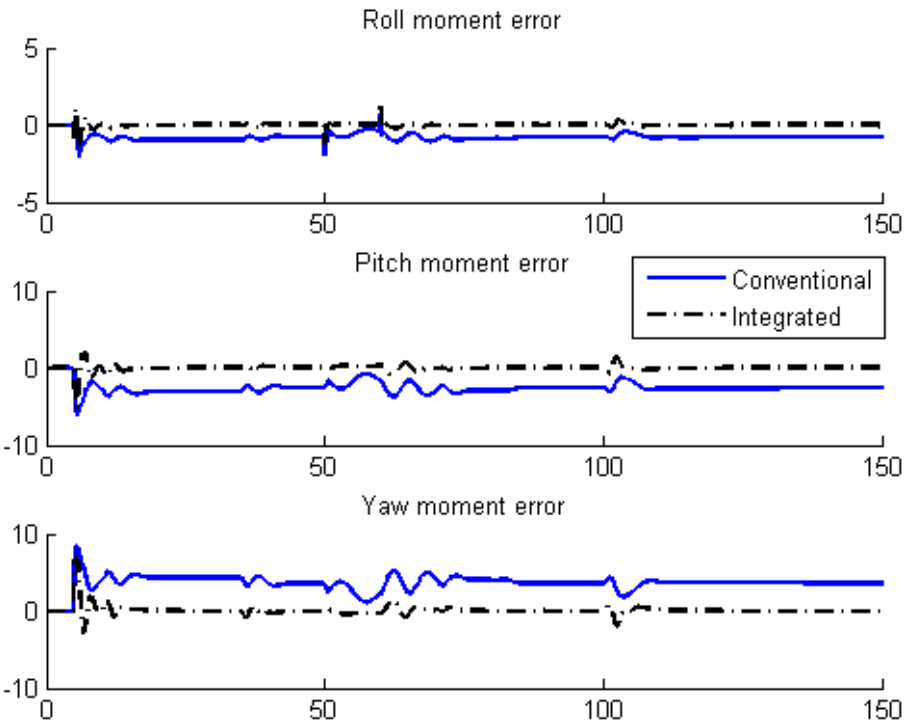

Figure 5.14: Virtual demand error in the roll, pitch and yaw planes. While the roll virtual demand error between the two schemes is similar, the integrated scheme is able to reduce the yaw and pitch moment virtual demands compared to the conventional scheme.

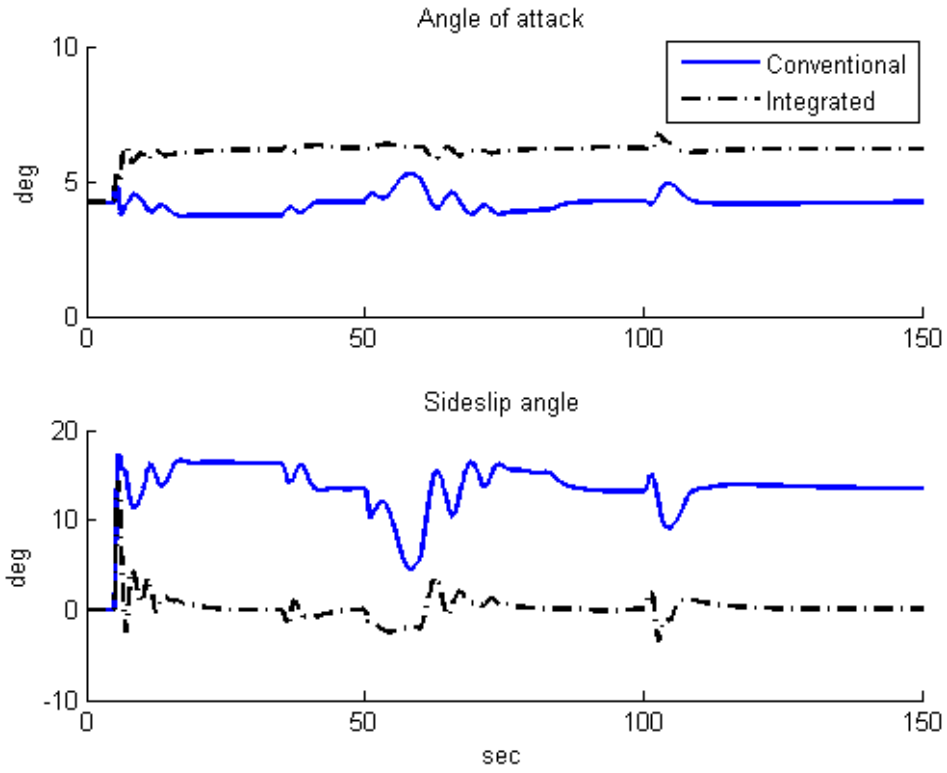

Figure 5.15: Angle of attack and sideslip angle comparison between the conventional and integrated schemes.

5.2.2 Comparisons on fault tolerant range

One important criteria to assess the control schemes is on their ability to improve on the actuator stuck fault range. The ability of the schemes to handle stuck ruddervator fault is now compared with the usage of flaps in the simulations.

First, the conventional scheme is shown in Figs.5.16 and 5.17 to able to handle a stuck ruddervator fault range between [-11,1] deg inclusive. This result also matches the range of an active reconfiguration controller's ability range in Ref [7] of [-9,3] deg.

For the integrated scheme, Figs.5.18 and 5.19 showed the integrated scheme being able to improve on the stuck fault range to [-12,3] deg, which is an 23% improvement in the results over the conventional and active reconfiguration control schemes. This may appeal as an attractive solution for consideration when stuck actuator fault tolerance in the system requires enhancement.

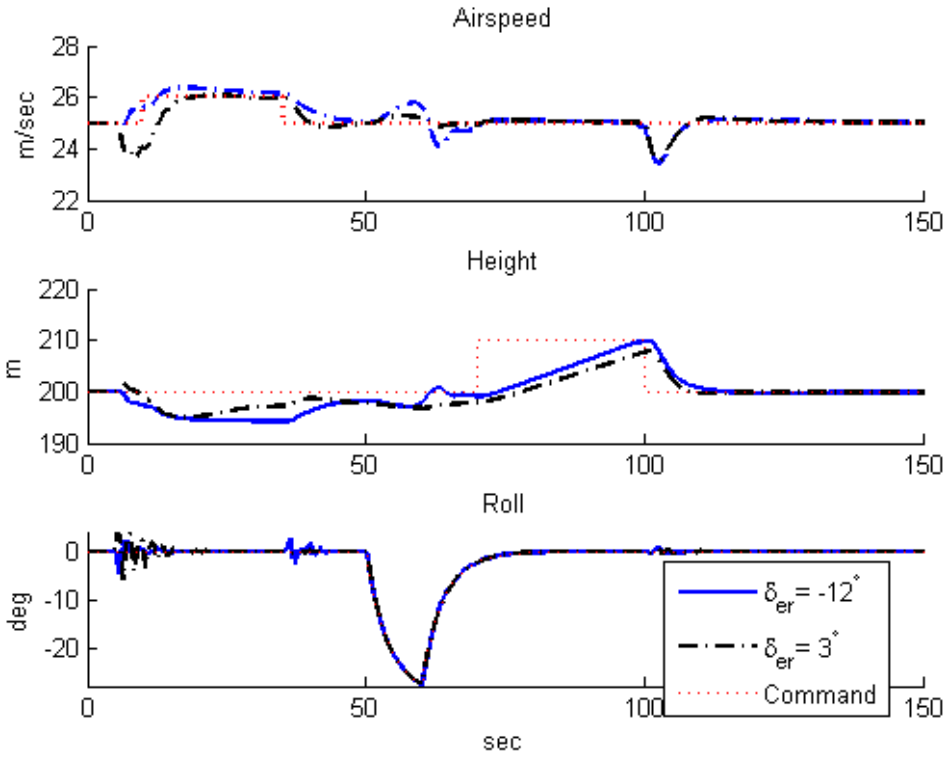

Figure 5.16: Conventional scheme's outer loop performance

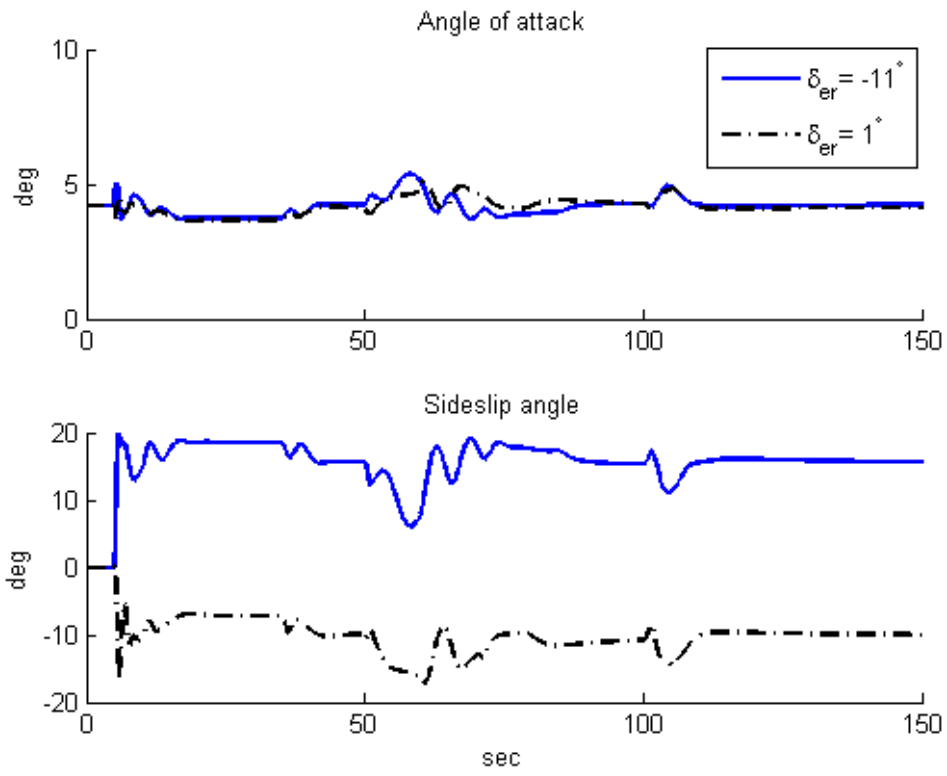

Figure 5.17: Angle of attack and sideslip angle at the extreme end of the conventional scheme.

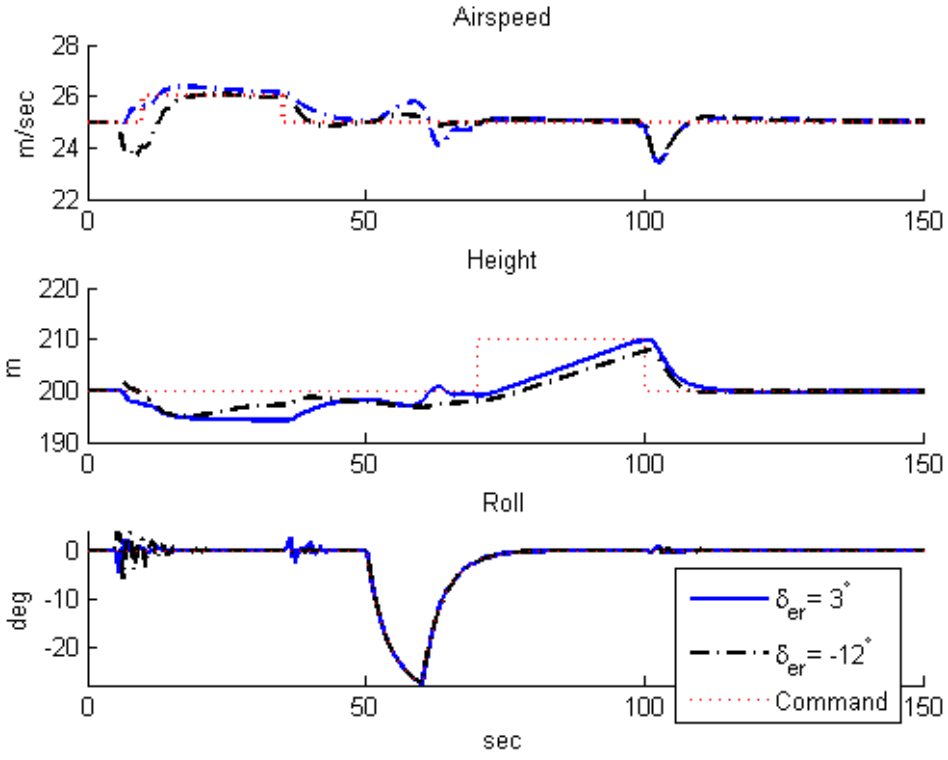

Figure 5.18: Integrated scheme's outer loop performance

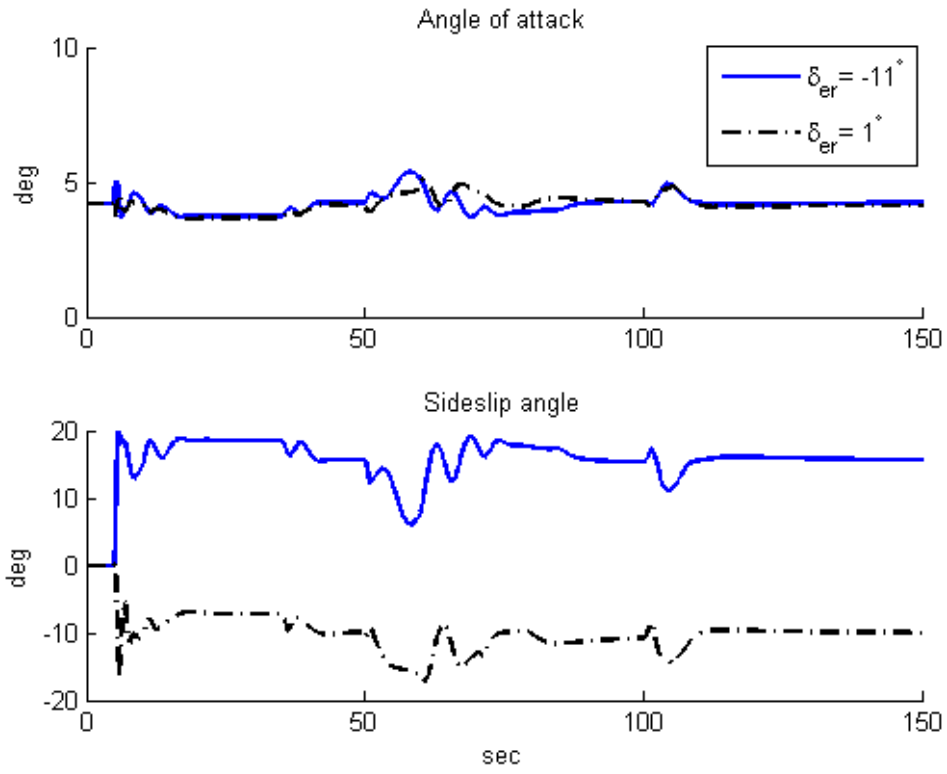

Figure 5.19: Angle of attack and sideslip angle at the extreme end of the conventional scheme.

5.3 Demonstration with guidance included

The previous simulation results showed the flight performance improvement by the integrated scheme with a classical flight controller when stuck actuator fault occur when compared with conventional scheme. In this section, a guidance scheme is included into the simulation setup to assess the impact of the integrated scheme into mission performance. The mission here is to track a pre-planned path as closely as possible, that is, to reduce the cross track error between the UAV and the path.

The UAV guidance scheme consists of a path generator and path follower. Note that as development of guidance schemes is beyond the scope of this study, one of well-known and most studied algorithm is selected for each path planning and path following. In the path generator, waypoints are used to describe the path. A waypoint consists of position coordinates, waypoint type (arc or line), radius and direction of turn of arc. Dubins curves [90] are used to construct the planned path which is optimal in path length. The path following algorithm used here is a nonlinear lateral path following algorithm from Ref. [91], which experimentally demonstrated it can track circular paths closely. With reference to Fig. 5.20, the path following algorithm generates the lateral acceleration command Ay_c as

$$Ay_c = 2\frac{V^2}{L_1}\sin\eta \tag{5.1}$$

where V is the inertial speed of the UAV, and η is the angle between the velocity vector and virtual target. The virtual target is set at a fixed L_1 distance ahead on the path from the UAV. In Ref [91], it has been shown to be similar to a Proportional Derivative Controller (PD) for small perturbations. L_1 parameter can be chosen to balance between stability and performance.

The lateral acceleration command can be translated to flight control roll command ϕ_c by assuming level flight as follows.

$$\phi_c = \arctan\left(\frac{Ay_c}{G}\right) \tag{5.2}$$

where G is the gravitational acceleration.

In the simulation, a right ruddervator stuck at -10 deg fault is introduced at simulation time of 2 sec. A constant wind of $10m/sec$ in the northern direction

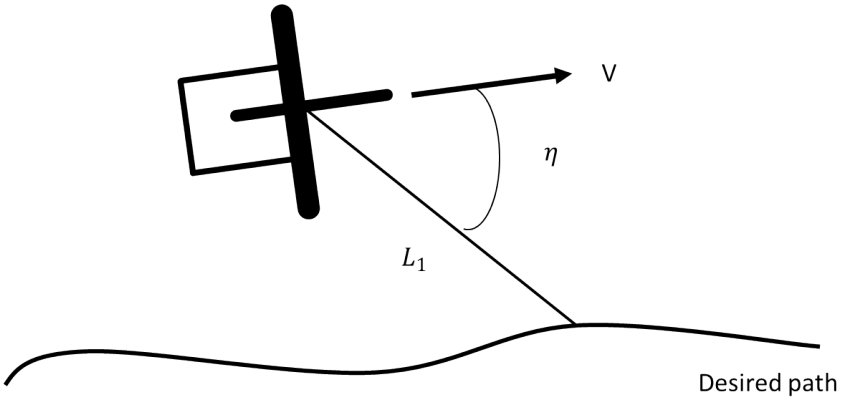

Figure 5.20: Diagram for the guidance law

is injected. This large injection of wind at 40% of the UAV's airspeed will test the ability of the guidance and control scheme further in coping with practical disturbance. The path consists of two sections of straight lines at each end of the path, with a detour arc of 200m radius, and small 100m radius between the lines and the detour semi-circle to smooth the path as shown. This simple path is similar to the path profile in Ref. [9]. In Ref. [9], the circular arc represents a pop-up No-Fly-Zone (NFZ) which the UAV has to avoid. Thus, small tracking error performance is important to prevent potential incursions into the NFZ. The lookahead distance L_1 is chosen to be 100m, which is the maximum allowed from stability considerations with the pre-planned path's minimum turn radius.

First, the guidance performance of the two schemes is compared. Fig. 5.21 shows the tracks taken by the integrated and conventional schemes. The integrated scheme's track is closer to the path compared to the conventional scheme's. This is illustrated in Fig. 5.22, which shows the magnitude of the cross track errors incurred by the two schemes.

Similar to earlier simulation results without the guidance scheme, the conventional scheme incurred a larger sideslip angle shown in Fig. 5.23 compared to the integrated scheme. This results in poorer guidance tracking by the conventional scheme, as typical guidance schemes may not measure or estimate the sideslip angle for compensation. Overall, the application of a nonlinear path following guidance scheme in the control architecture quantified the improvements to mission performance by the integrated scheme.

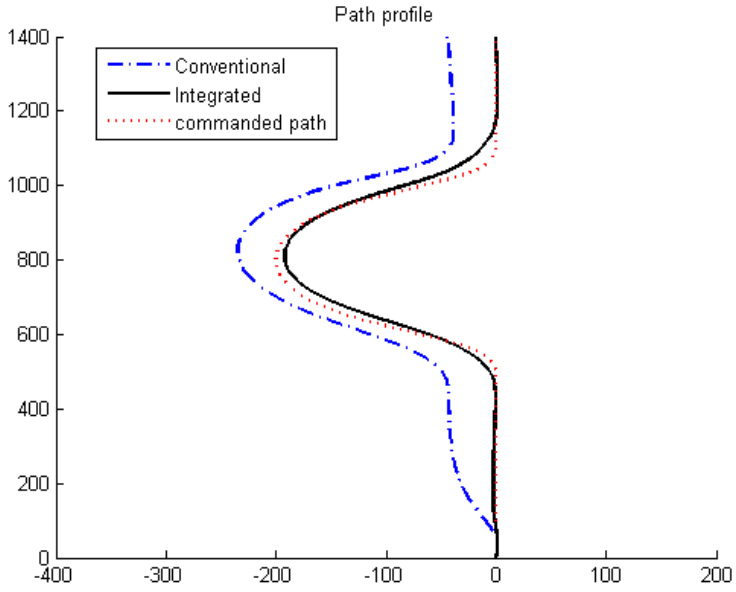

Figure 5.21: Lateral plot of the paths taken by the two schemes with the pre-planned path

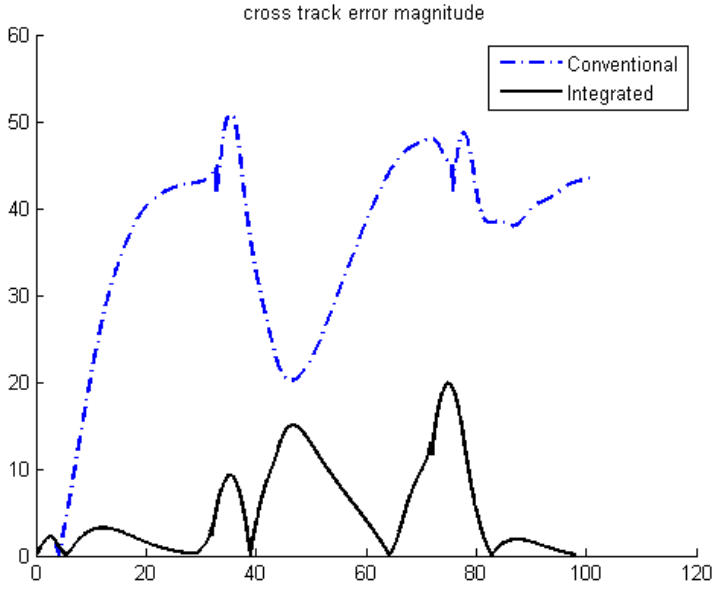

Figure 5.22: Cross track error performance of the two schemes.

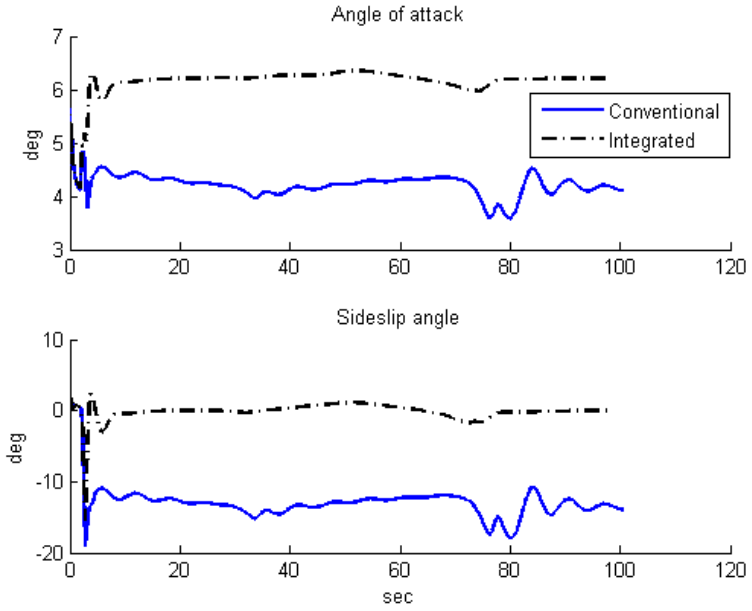

Figure 5.23: Comparison of angle of attack and sideslip angles of the two schemes with guidance loop

5.4 Conclusion

This chapter presented a system level analysis of the usage of the integrated actuator control scheme on an Aerosonde UAV model with the incorporation of flight control and guidance control loops into the simulation to assess the overall system effects of using the integrated actuator control scheme.

A classical flight control loop is first included into the simulation to assess the flight performance differences between the conventional and integrated schemes. A severe right aileron fault is first simulated. While the integrated scheme has a smaller unintended roll due to the fault occurrence, the two schemes are largely similar in performance for both simulations with or without flaps usage. This is due to the availability of control authority redundancy offered by the remaining healthy left aileron. The integrated scheme may offer little improvement when adequate control redundancy is available for fault handling.

A right ruddervator fault is then simulated. Unlike the aileron case, the two ruddervators are used as dominant yaw and pitch control effectors nominally. With the loss of control to one ruddervator, the challenge is to fulfil the desired control demands by flight control using the remaining healthy actuator and the two ailerons. In this case, the integrated scheme reduces the allocation error compared to the conventional scheme. This in turn reduces the steady state sideslip angle magnitude dramatically.

The significance of the better allocation performance on fault tolerance through the usage of the integrated scheme is the ability to improve on the fault tolerant stuck range of the UAV to a ruddervator fault. The integrated scheme is able to improve on the stuck fault tolerant range of the over the conventional scheme, as well as results in Ref. [7] by 23%.

Furthermore, the impact of the flight control tracking performance by the integrated scheme on the overall system performance during missions is assessed by including a path following guidance scheme to the simulation setup. Due to the conventional scheme incurring a larger stuck trim sideslip angle with a stuck ruddervator fault compared to the integrated scheme. This translates to better cross track error performance by a nonlinear guidance scheme to follow a path as its mission. The guidance improvements come about due to the better

flight regulation performance by the integrated scheme which alleviates the fault degradation effects.

Overall, the work in this chapter presents the areas which the integrated scheme may have improvements over the conventional one, and where little differences may be seen. The integrated scheme shines in situations when there is little degree of freedom in the control authority remaining.

Chapter 6

A guidance reconfiguration strategy

6.1 Introduction

Actuator faults reduce the amount of control authority remaining in the UAV system. This leads to the degradation in the UAV manoeuvring capability, which may impact its ability to successfully meet its mission demands. Many schemes have focused on fault tolerant control to manage fault degradation.

Fault tolerant guidance may be deployed with fault occurrence. Fault tolerant guidance would be necessary especially when the available system's control resources are limited, and when fault tolerant control alone is insufficient [29]. From the system perspective, the guidance scheme needs to be compatible with the degraded system to achieve mission success. When faults occur, guidance schemes may generate excessive demands in which the degraded system may be unable to fulfil. This may degrade the performance of the overall system, or even resulting in system instability. Managing the guidance scheme's demand also potentially leads to improving the chances of mission success in the degraded system.

A typical guidance scheme may consist of a path manager to provide the path, and a path following algorithm to synthesize the flight control command required for tracking the path. In recent research on fault tolerant guidance, Ref. [9] proposed a method to reconfigure the path's radius given the actuator fault in the path manager. This is applied to the No-Fly-Zone mission whom mission success depends on the UAV maintaining within the safety margin from the planned path.

However, this may not be sufficient when handling extremal actuator faults, or may pose severe constraints to other aspects of path planning. The aim of this research is to propose a method which adapts the guidance algorithms according to the severity of the aileron actuator fault. The usefulness of this research is to provide an easily implementable method to change the guidance scheme's lookahead distance parameter for accommodating the degraded system's capabilities.

By reconfiguring both the path and the path following algorithm parameters, extremal faults may be accommodated more easily as fault accommodation is shared by both the path planning and path following schemes. This results in smaller changes to either of the schemes due to the fault, which results in closer performance to pre-fault conditions.

The organization of this chapter is as follows. First, the guidance scheme is described. This is followed by a description of the effects of the aileron fault on the system. The proposed guidance reconfiguration scheme is next presented. This is followed by presentation of the simulation results, which is followed by the conclusions of the chapter.

6.2 Guidance Scheme

The mission being considered here is a path following mission where the UAV is required to track a desired path. This task is chosen as many high level missions may be described as an UAV following a desired path. Typically, the guidance scheme used to implement a path following mission may compose of a path manager describing the desired path, and a path following algorithm to synthesize the desired commands for flight control execution. Dubin's paths [90] has been used to construct the path as it is optimal in path distance yet conceptually simple. The path is planned using straight lines and arcs.

A UAV nonlinear path following guidance algorithm has been recently developed and has successfully shown good experimental results is used here as the path following algorithm [91]. In an evaluation of UAV path following algorithms, the chosen scheme is shown to require the least control effort compared with other state of the art algorithms [92] [93]. This algorithm is chosen here as the remaining healthy control effort in fault occurrence may be severely constrained.

As shown in Fig. 6.1, the nonlinear guidance law to generate the lateral acceleration Ay_c is

$$Ay_c = \frac{2V^2}{L_1} \sin(\eta) \tag{6.1}$$

where V is the inertial speed and η is the angle between the inertial velocity and the virtual target. The virtual target is set to be a lookahead distance L_1 between

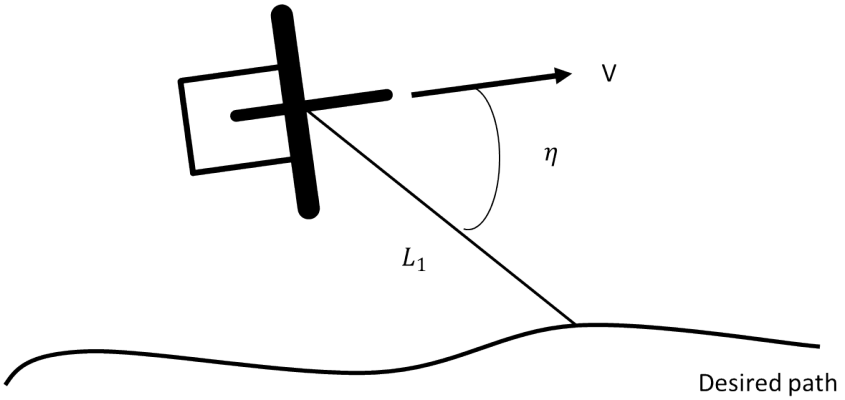

Figure 6.1: Path following guidance law geometry

the UAV and on the desired path. In the next section, the actuator fault effects are described followed by the implications to guidance path following capabilities.

6.3 Aileron actuator fault

This section describes the effects of a stuck aileron fault. Starting with the basic torque equation:

$$\tau = \mathbf{I}\dot{\omega} + \omega \times \mathbf{I}\omega \tag{6.2}$$

with $\omega = \begin{bmatrix} p & q & r \end{bmatrix}^T$; p, q and r being the roll, pitch and yaw rates of the UAV respectively. The inertia matix $\mathbf{I}$ is

$$\mathbf{I} = \begin{bmatrix} I_{xx} & 0 & I_{xz} \\ 0 & I_{yy} & 0 \\ I_{xz} & 0 & I_{zz} \end{bmatrix} \tag{6.3}$$

$\tau = \begin{bmatrix} L & M & N \end{bmatrix}^T$ is the torque moment acting on the UAV, with L, M and N being the roll, pitch and yaw moments respectively. The roll moment L for a typical small UAV may be modelled as

$$L = \bar{q}Sb_{ref}\left(C_{l_\beta}\beta + C_{l_{\bar{p}}}p + C_{lr}r + \mathbf{C_{l}}_{\delta\mathbf{act}}\theta_{\mathbf{act}}\right) \tag{6.4}$$

where $\bar{q}$ is the dynamic pressure, S and b_{ref} are the surface area and reference length respectively. C_{l_*} refers to the relevant aerodynamic coefficients of the UAV. $\mathbf{C_{l}}_{\delta\mathbf{act}} = \begin{bmatrix} C_{l_{\delta act_1}} & \dots & C_{l_{\delta act_n}} \end{bmatrix}$, and $\delta_{\mathbf{act}} = \begin{bmatrix} \delta_{act_1} & \dots & \delta_{act_n} \end{bmatrix}^T$, with the system having n actuators. In the roll axis, the linearised roll rate about trim can be expressed as

$$\dot{p} = -V\theta_p p + V^2\theta_{\mathbf{act}}\delta_{\mathbf{act}} \tag{6.5}$$

where

$$\theta_p = \frac{\rho I_{zz} S {b_{ref}}^2 Cl_p}{4\left(I_{xx}I_{zz} - I_{xz}^2\right)} \tag{6.6}$$

$$\theta_{\delta_{\mathbf{act}}} = \frac{\rho I_{zz} S b_{ref}}{4\left(I_{xx}I_{zz} - I_{xz}^2\right)}\mathbf{Cl}_{\delta_{\mathbf{act}}} \tag{6.7}$$

Integrating with respect to time Eqn. 6.5 results in

$$p = \frac{V}{\theta_p}\theta_{\mathbf{act}}\delta_{\mathbf{act}}\left(1 - e^{-V\theta_p t}\right) \tag{6.8}$$

6.3.1 Roll performance degradation due stuck aileron fault

Due to the location of the different control effectors on the UAV, the ailerons on the UAV are the dominant control effectors for roll control. Assuming the other control actuators are dominantly used to effect the moment axes and their contributions to the roll channel is small, Eqn. 6.8 can be reduced to

$$p = \frac{V}{\theta_p}\theta_{da}\delta_{da}\left(1 - e^{-V\theta_p t}\right) \tag{6.9}$$

where

$$\theta_{\delta_{act}} = \frac{\rho I_{zz} S b_{ref}}{4\left(I_{xx}I_{zz} - I_{xz}^2\right)} C_{l_{\delta_a}} \tag{6.10}$$

with $\delta_a = \delta_{al} - \delta_{ar}$, where δ_{al} and δ_{ar} are the left and right aileron positions respectively. Let the ailerons' position range be $[-u_{max}, u_{max}]$. When an actuator fault occurs, the roll rate performance would be reduced more in one direction than the other when the faulty stuck actuator is not at the trim or null position. The fault also reduces the healthy control authority, and hence the maximum roll rate achievable in the opposite direction to the fault induced moments. This is due to both the fault effect, and the need for the remaining healthy actuators to compensate for the fault to achieve trim. Let

$$\theta_{\delta_{a\min}} = \min\left(|\delta_{fstuck} - \delta_{ar_{\min}}|, |\delta_{fstuck} - \delta_{ar_{\max}}|\right) \tag{6.11}$$

be the minimum aileron authority between the left and right roll directions remaining post fault to influence the roll moment. Using this and integrating Eqn. 6.9, a conservative roll capability of the degraded system may be expressed as

$$\phi_{cap} = V\frac{\theta_{\delta_a}}{\theta_p}\left(1 - \frac{1}{V\theta_p}\right)\Delta_{\delta_{a_{\min}}} t \tag{6.12}$$

Using Eqn. 6.12 as a conservative estimate of the system's roll rate capability, the proposed reconfiguration scheme is next described.

6.4 Proposed method

6.4.1 Reconfiguration of path following algorithm

In many path following schemes, a virtual target is used to aim for by the algorithm in order to synthesize the guidance acceleration commands to flight control. With the usage of a virtual target planted L_1 distance between the UAV and itself along the desired path, the actual acceleration profile is changed/ smoothed, from the step acceleration demand to a gradual one. Fig. 6.2 shows the change in profile through the usage of the virtual target.

For a Bank-To-Turn UAV, the flight control roll command may be generated from the guidance lateral acceleration command by assuming level flight

$$\phi_{cmd} = \arctan\left(\frac{Ay_c}{G}\right) \tag{6.13}$$

where G is the gravitational constant. Substituting Eqn. 6.1 into Eqn. 6.13,

$$\phi_{cmd} = \arctan\left(\frac{2V^2}{L_{1_{nom}}G}\sin(\eta)\right) \tag{6.14}$$

where $L_{1_{nom}}$ is the nominal virtual target distance designed. In the nominal design of guidance, the ratio of $\frac{L_{1_{nom}}}{R_{\min}}$ may be determined to satisfy the design requirements. This ratio K may be expressed as

$$K = \frac{L_{1_{nom}}}{R_{\min}} \tag{6.15}$$

From Eqn 6.14, the maximum required roll command $\phi_{required}$ to track the path

$$\phi_{required} = \arctan\left(\frac{2V^2}{KR_{\min}G}\right) \tag{6.16}$$

where $R_{\min}$ is the path's minimum radius.

As the fault reduces the roll rate achievable by the system, the time required by the system to reach the required roll angle may increase. This can be calculated by

$$T_{\min} = 2\frac{\phi_{required}}{\phi_{cap}} \tag{6.17}$$

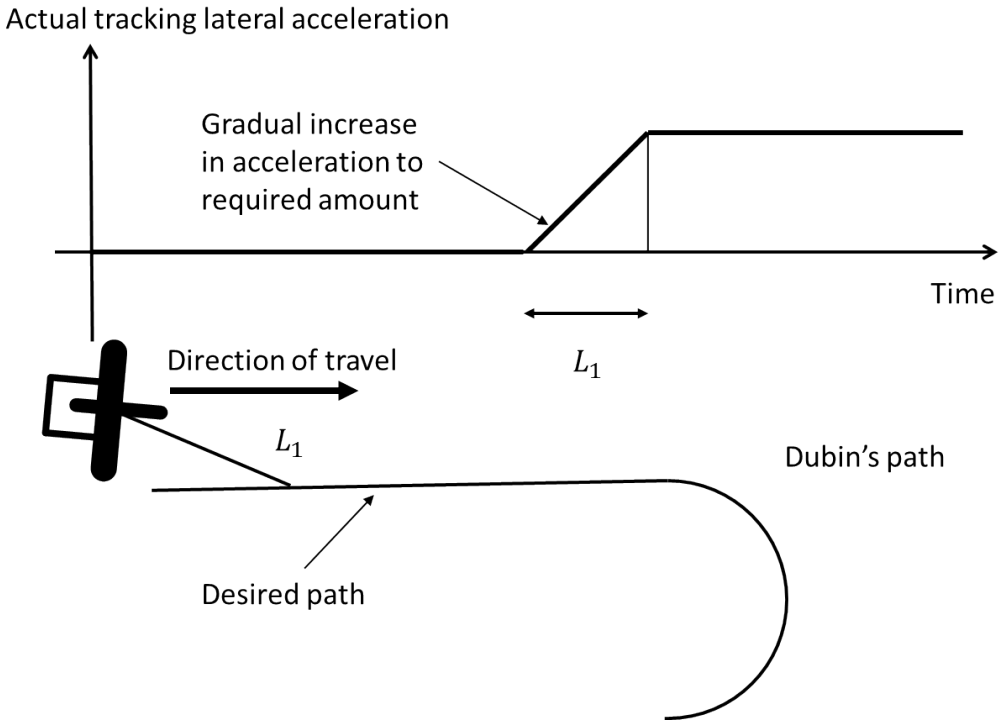

Figure 6.2: Acceleration profile required to track the desired Dubin's path, and the profile with a virtual target

The factor of 2 in Eqn. 6.17 is to account for the possibility of doublet manoeuvres, where the UAV is initially banked at one roll angle and turning to the opposite direction of roll.

To be compatible with the reduced capabilities of the system, the proposed method increases the minimum lookahead distance L_1 according to the relationship

$$L_{1_{\min}} = VT_{\min} \tag{6.18}$$

Therefore, upon fault occurrence and having fault actuator position, the guidance lookahead distance L_1 would be changed to

$$L_{1_{new}} = \max\left(L_{1_{nom}}, L_{1_{\min}}\right) \tag{6.19}$$

This means when the actuator fault is relatively minor, and the system is assessed to have sufficient remaining control authority to complete the manoeuvre, no change to the path following algorithm's lookahead distance is proposed. This is in contrast to the method in Ref [75], in which reconfiguration occurs regardless of the size of the fault.

The proposed method assumes the flight control system continues to maintain the stability of the degraded system. In the case when the fault becomes too severe, any remaining control authority should be prioritized for flight stabilization before guidance.

6.4.2 Impact of path following ability of the guidance scheme with change

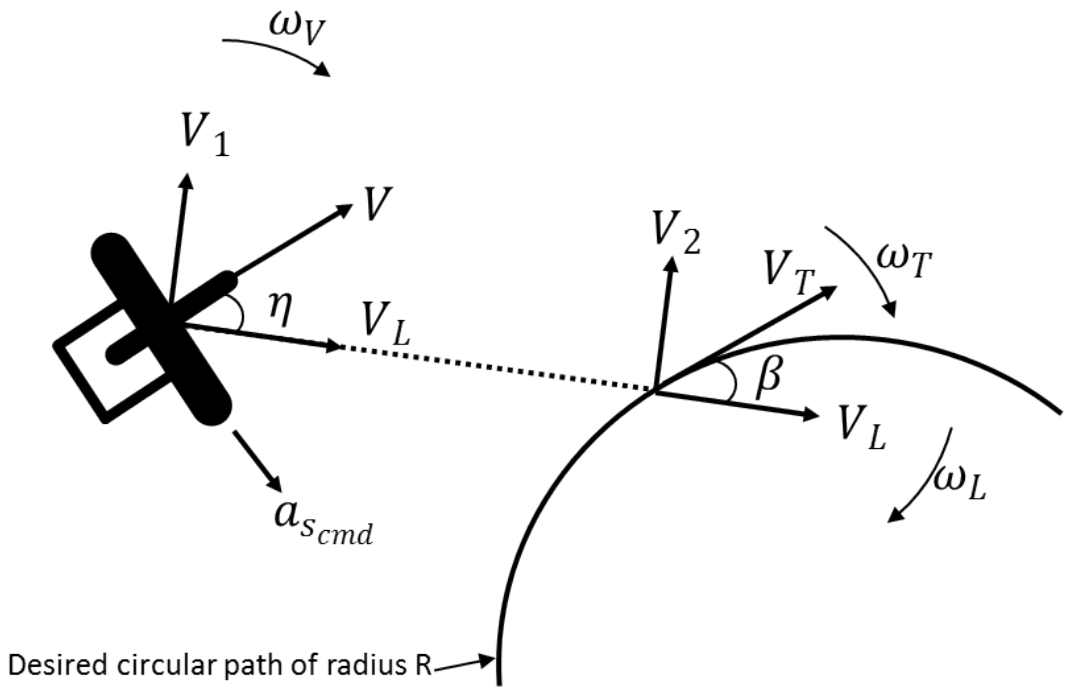

Figure 6.3: Path following algorithm's geometry

The stability of the guidance scheme is closely related to the lookahead distance L_1 as the parameter manages the sensitivity of the algorithm. Insights to the impact of the proposed L_1 change in the proposed scheme is analysed here using some of the results from Ref. [91].

The rates of change of angles η and β can be described in terms of inertial angular rotation rates ω_L, ω_V, and ω_T from Fig. 6.3 as

$$\dot{\eta} = \omega_L - \omega_V \tag{6.20}$$

$$\dot{\beta} = \omega_L - \omega_T \tag{6.21}$$

Resolving the UAV and virtual target's velocities parallel and orthogonal to the L_1 direction with reference to the geometry in Fig. 6.3, the system differential equations can be obtained as

$$\dot{\eta} = \frac{V}{KR_{min}} (\sin\eta + \cos\eta \tan\beta) \tag{6.22}$$

$$\dot{\beta} = \frac{V}{KR_{min}} (\sin\eta - \cos\eta \tan\beta) - \frac{V\cos\eta}{R\cos\beta} \tag{6.23}$$

The states are well-behaved and satisfy Lipschitz conditions if

$$|\eta(t)| < 90\,\text{deg} \tag{6.24}$$

$$|\beta(t)| < 90\,\text{deg} \tag{6.25}$$

A unique stationary point or perfect following of the desired arc path is achieved with

$$\begin{aligned} \eta_0 &= \sin^{-1}\left(\frac{K}{2}\right) \\ &= -\beta_0 \end{aligned} \tag{6.26}$$

From Eqn 6.26, a feasible solution exists only when K's range is limited to

$$0 < K \leq 2 \tag{6.27}$$

A physical interpretation of the upper bound of K arises from the virtual target needing to track a circle of $R_{\min}$, in which case the maximum distance possible between a vehicle on the desired path and the circle is $2R_{\min}$.

Another way of assessing the impact of change in K is to consider the linearised dynamics of Eqn. 6.22 and Eqn. 6.23 about the stationary point. The linearised dynamics of the guidance law may be approximated as a second order system with damping ratio and bandwidth of

$$w_n = \frac{\sqrt{2}V}{KR_{min}} \tag{6.28}$$

$$\zeta = \frac{1}{\sqrt{2}} \sqrt{1 - \left(\frac{K}{2}\right)^2} \tag{6.29}$$

where w_n and ζ are the bandwidth and damping ratio of the system, and R is the radius of the path. From the two equations, changing L_1 value can be seen to influence both of them. In Eqn 6.28, the bandwidth of the linearised system is inversely related to K. To visualize the relationship between K and the damping ratio ζ, Fig. 6.4 shows the relationship between the two variables. From the figure and author's experience, $K \leq 1$ for reasonable good tracking performance.

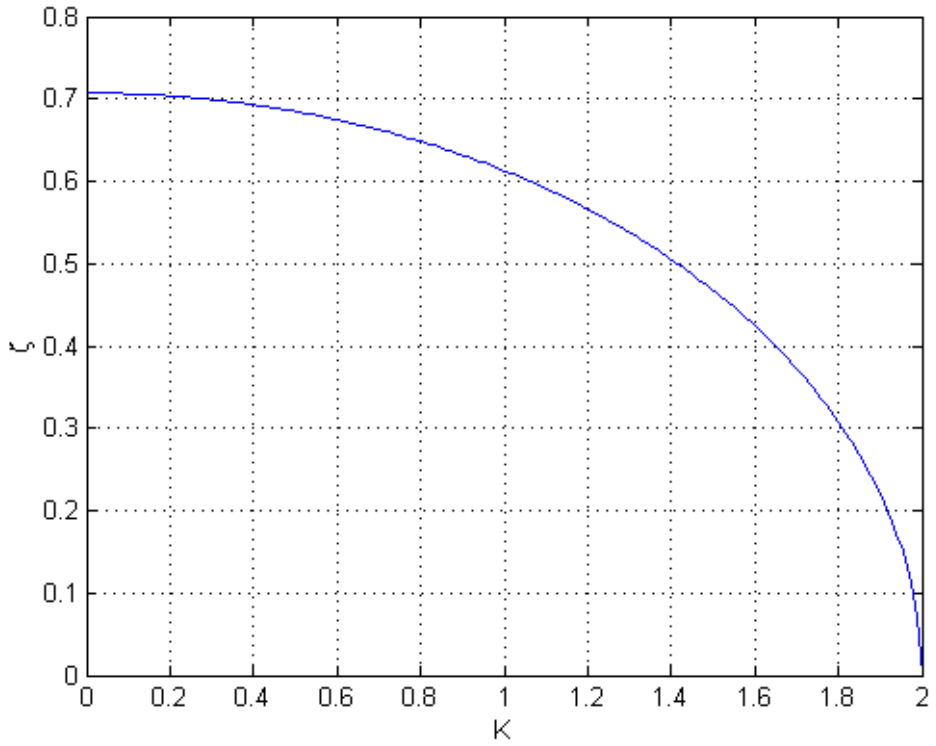

Figure 6.4: Plot showing relationship between K and damping ratio ζ.

6.4.3 Changing path radius to mitigate fault

A problem with this approach is that when the fault is large or stuck towards the extremal positions leaving little healthy actuation authority, the lookahead distance L_1 may be dramatically increased using above method. This results in poor tracking performance which may be unacceptable. This effect may be alleviated if the path is allowed to be rerouted using a larger minimum turning radius. The following provides a strategy for this. Combining Eqns. 6.17 and 6.16,

$$\tan\left(\frac{L_1\phi_{cap}}{2V}\right) = \frac{V^2}{RG} \tag{6.30}$$

Using Eqn. 6.15 and approximating tan(x) by x, the minimum radius for replanning the path with a chosen K ratio may be found as

$$R_{\min_{re}} \approx \sqrt{\frac{2V^3}{KG\phi_{cap}}} \tag{6.31}$$

The corresponding L_1 is

$$L_1 = KR_{\min_{re}} \tag{6.32}$$

Using Eqns. 6.31 and 6.32, the proposed method provides the changes to $R_{\min}$ andL_1 required to maintain the damping ratio of the linearised dynamics of the guidance scheme with fault occurrence. This indirectly helps to maintain the guidance performance of the degraded system.

In summary, the proposed method suggests the following

- Minor fault. The system has sufficient remaining capability to complete the mission. No change to the guidance parameters is proposed.
- Severe fault. The system does not have sufficient capability to meet the desired path. L_1 is increased to accommodate the fault using Eqn. 6.18.
- Very severe fault. The system has little authority remaining. L_1 change is too large relative to the original minimum radius, or when reconfigured would bring about unacceptable tracking performance. One way to alleviate this is to consider the path being rerouted with a larger and less restrictive minimum radius. The proposed scheme suggests a method to proportionally increase the lookahead distance and the minimum radius to preserve the linearised dynamics damping ratio of the nominal design. Eqns. 6.31 and 6.32 are used to determine the new $R_{\min}$ and L_1 respectively.

6.5 Simulation results

In this section, the proposed guidance reconfiguration scheme is analysed using the Aerosonde UAV model detailed in Ref [7]. As described in the previous chapter, the UAV guidance scheme consists of a path generator, path follower and guidance reconfiguration module as shown in Fig. 6.5. In the path generator, waypoints are used to describe the path. A waypoint consists of position coordinates, waypoint type (arc or line), radius and direction of turn of arc. Dubins curves [90] are used to construct the planned path which is optimal in path length. The path following algorithm used here is a nonlinear lateral path following algorithm from

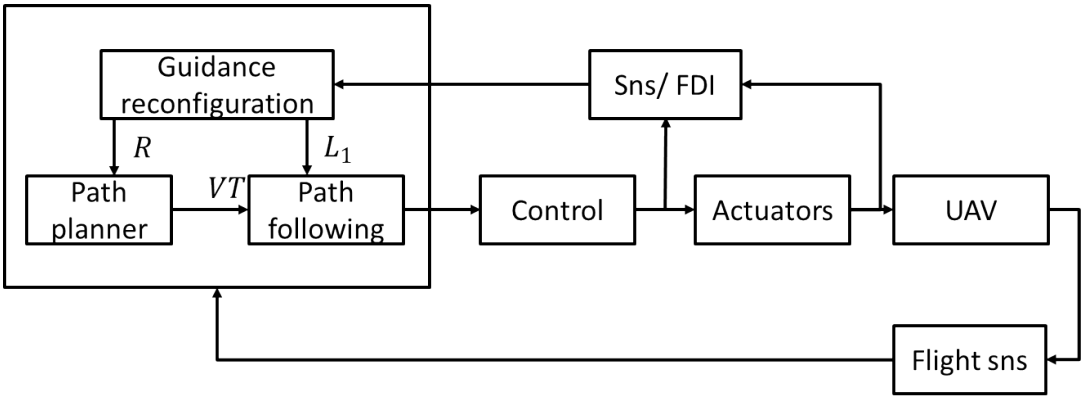

Figure 6.5: Simulation block diagram for the reconfiguration guidance scheme.

Ref. [91], which experimentally demonstrated it can track circular paths closely. Actuators faults are either detected using sensors or through a fault detection and isolation system. When faults occur, the information is passed to the guidance reconfiguration module, which assesses the severity of fault and reconfigures the guidance parameters according to the proposed method. The control and other modules are the same as the simulation setup described in the previous chapter.

First, the simulation results for a right aileron actuator with 14 deg fault at simulation time of 10 sec are shown in Figures. 6.6. Here, the path taken by the UAV with and without guidance reconfiguration is compared. With reconfiguration in accordance to the proposed method, it is seen the UAV tracks the desired path better than the one without reconfiguration. This is confirmed in Fig. 6.7, where the cross track guidance error is plotted. With the proposed reconfiguration method, the maximum guidance error is reduced significantly. The reason for this is shown in Fig. 6.8, where the remaining healthy left aileron is saturated at its limit during some periods of time. This leads to the inability of the UAV without reconfiguration to track the guidance roll command well, as shown in Fig. 6.9.

Next, the simulation is repeated with a smaller right aileron actuator fault of 7 deg at simulation time of 10 sec. Due to the smaller fault, the calculated L_{1new} is smaller than the nominal L_1, resulting in no changes to the guidance lookahead distance. Thus, the virtual target distance is maintained at the nominal L_1 value of $40m$ as the actuator fault is deemed to be minor by the proposed scheme. Figures 6.10 to 6.13 compare the simulation with fault against the nominal fault free case. Fig. 6.10 shows generally good tracking of the roll command despite

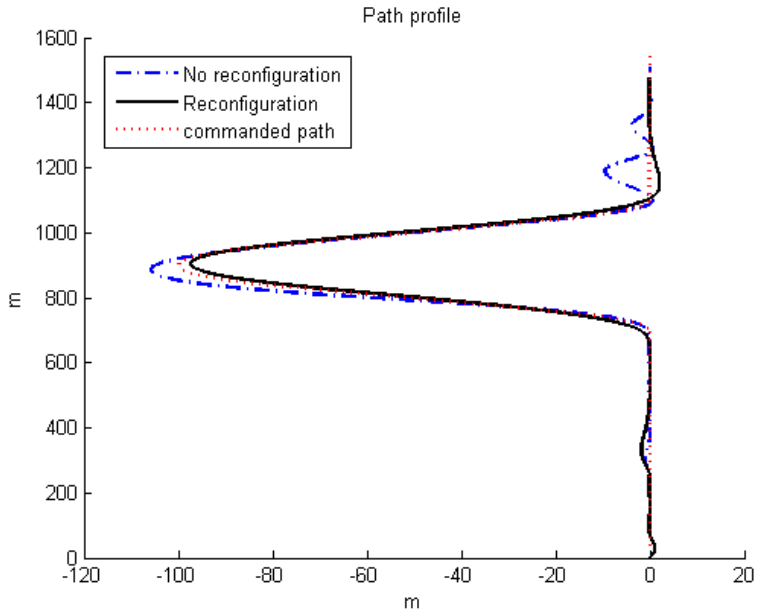

Figure 6.6: UAV paths of simulations with right aileron fault of 14 deg

fault occurrence, which is similar to the nominal fault free case presented for comparison. This is due to the remaining healthy left aileron being able to fulfil the demands by control and guidance, as shown in Fig. 6.11. Next, in Fig. 6.12, it can be seen both the nominal fault free and the fault simulations tracked the desired path closely. The magnitude of the cross track error performance is further examined in Fig. 6.13. As one may expect, the simulation with fault has a slighter larger cross track error compared to the fault free case due to the fault degradation, which may still be acceptable for mission requirements.

In the next simulation, the change in radius to improve the guidance performance post fault is examined. Here, the right aileron is stuck at fault position of 16 deg out of the total possible 18 deg position limit. This leaves a marginal 2 deg of the healthy aileron above the trim value to manage the guidance requirements. In this case, only changing the lookahead distance to accommodate the fault results in a L_1 being $147m$, which is larger than the minimum radius of the planned path. This naturally results in poor tracking performance as shown in Fig. 6.14. Changing both the mimimum radius and the lookahead distance in this case results in the path radius having a minimum radius of $203m$ and the lookahead distance is $81m$ according to the proposed method. The same figure shows the UAV being able to track the planned path well.

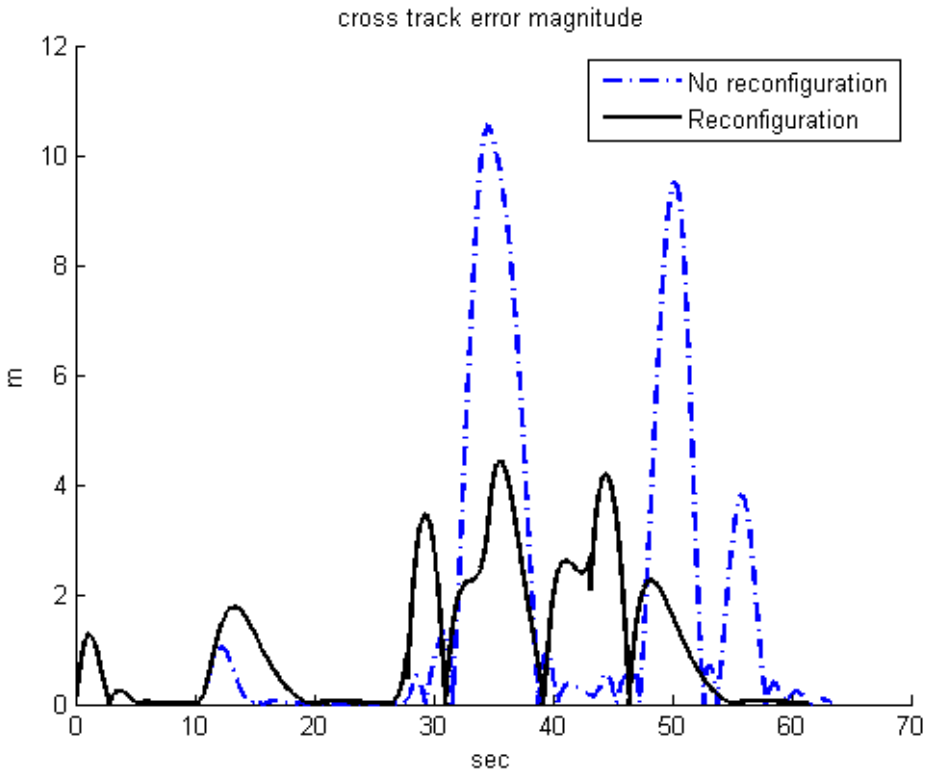

Figure 6.7: Guidance cross track error of simulations with right aileron fault of 14 deg

For the last comparison simulation, the proposed method is compared with another recent reconfiguration method in Ref. [9] for No-Fly-Zone (NFZ) application. Sometimes, the UAV may be presented with pop up areas which the UAV would be required to detour from its original intended path. The path's turning radius and the ability to reasonably track the path is important to prevent excursions into the NFZ. To recap, in Ref. [9], the proportion of the remaining aileron authority in terms of left and right control authority is used to reduce the roll angle limit and increase the roll time constant used. The information is next used to determine the minimum radius required to handle the fault degradation.

A right aileron fault of 12 deg is simulated with a NFZ of 80m with safety margin of 20m. The proposed method increases the lookahead distance L_1 slightly from the original 40m to 49m to accommodate the fault, while maintaining the original path. In contrast, Ref. [9]'s method requires the increase of the minimum radius of the path to 458m. Fig. 6.15 presents the paths taken by the two methods

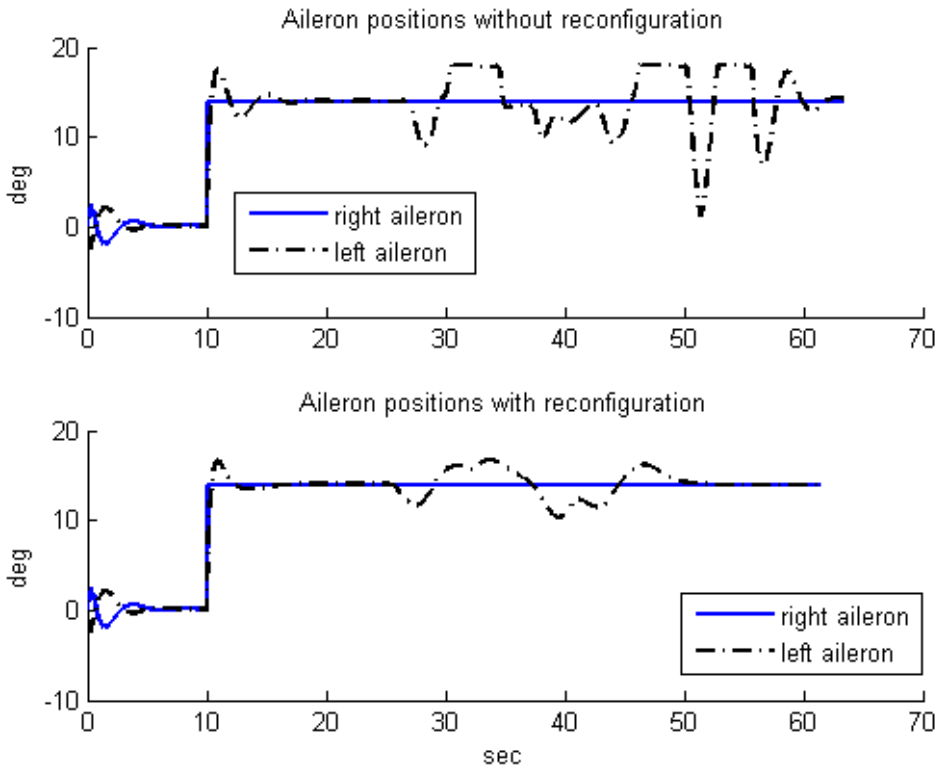

Figure 6.8: Actuator plots of simulations with right aileron fault of 14 deg

relative to the NFZ. From the figure, the proposed scheme is able to track the original path relatively well without a change in radius with the fault, while a huge increase in path radius is suggested for the other method.

If one is to increase the fault position towards its extremity, the size of the minimum radius will increase significantly using Ref [9] method. This may pose difficulties for path planners with the dramatic increase in path radius. On the other hand, the proposed method reduces the potential change in minimum radius by first using an increase in lookahead distance to alleviate the fault degradation. This additional degree of freedom in handling the fault allows the designer to trade-off between guidance tracking and path planning, which potentially enables coping with mission constraints more effectively.

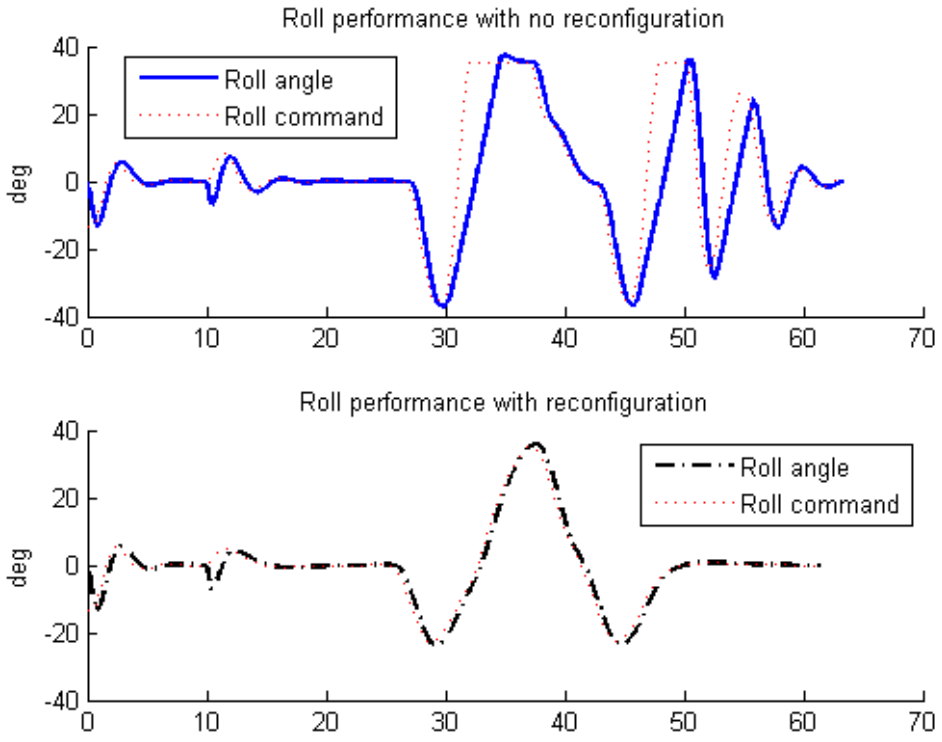

Figure 6.9: Roll tracking performance of simulations with right aileron fault of 14 deg

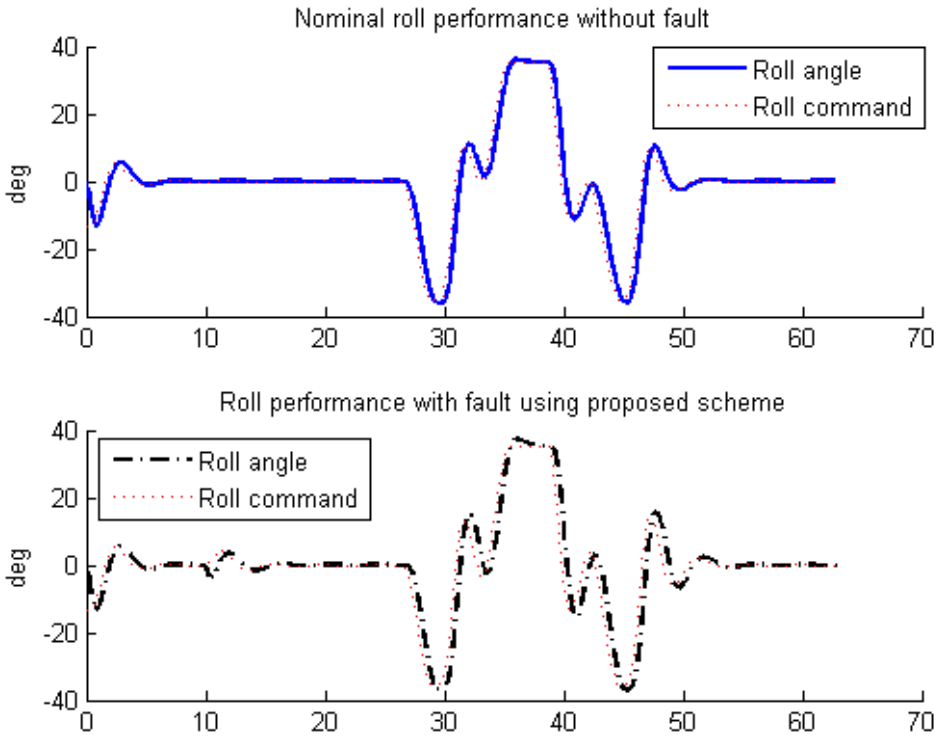

Figure 6.10: Roll tracking performance of simulations with right aileron fault of 7 deg and fault free case

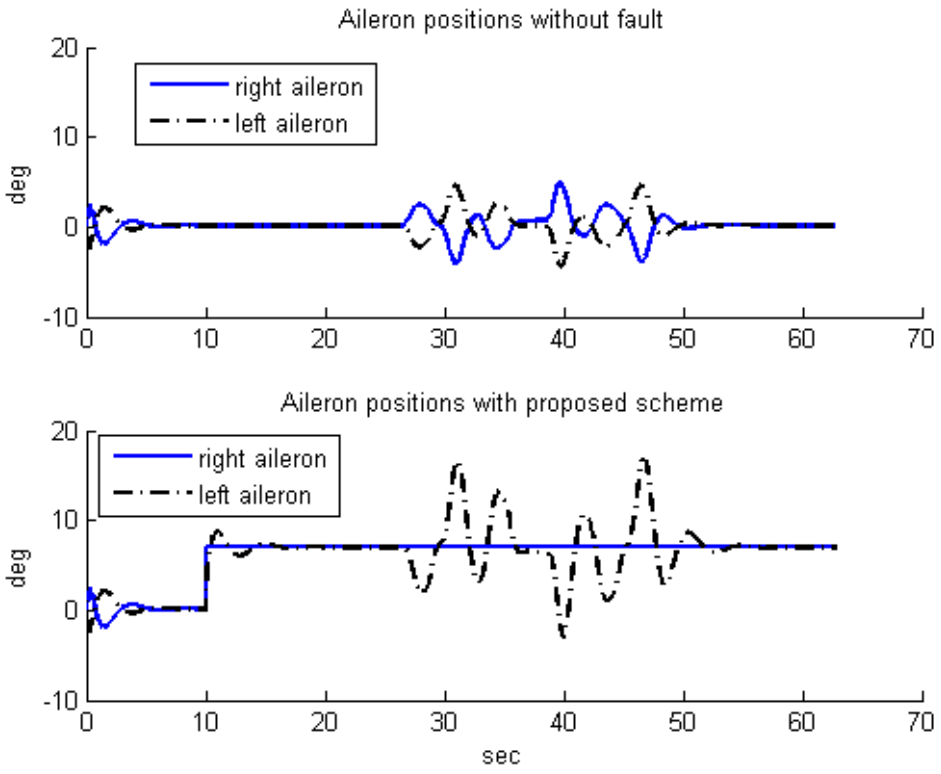

Figure 6.11: Aileron plots of simulations with right aileron fault of 7 deg and fault free case

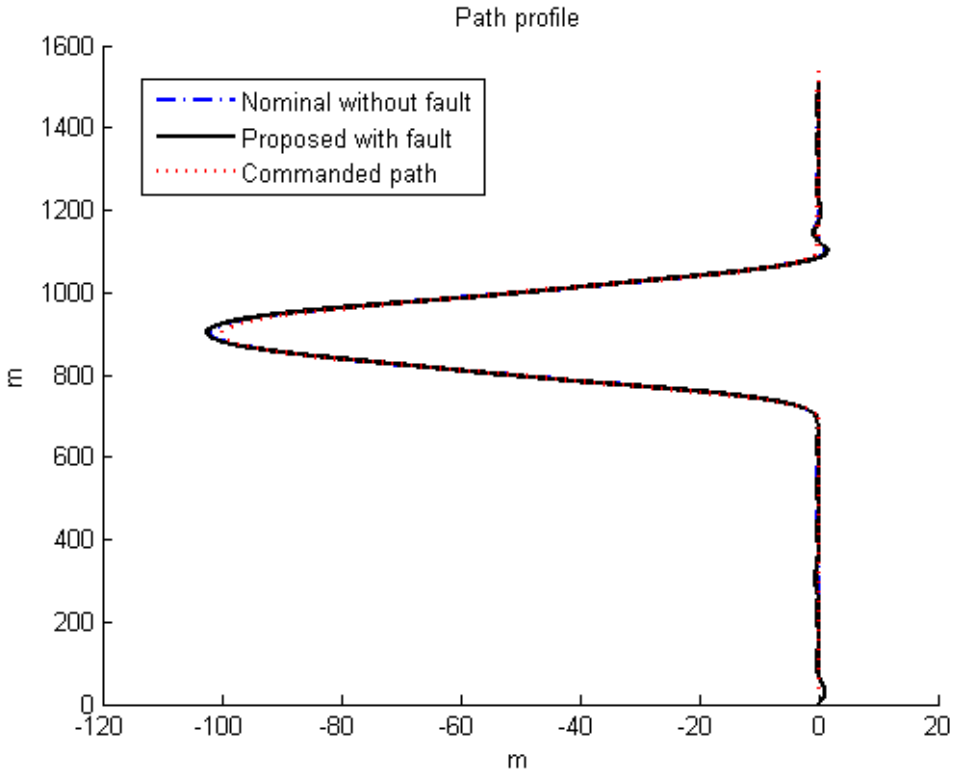

Figure 6.12: UAV paths of simulations with right aileron fault of 7 deg and fault free case

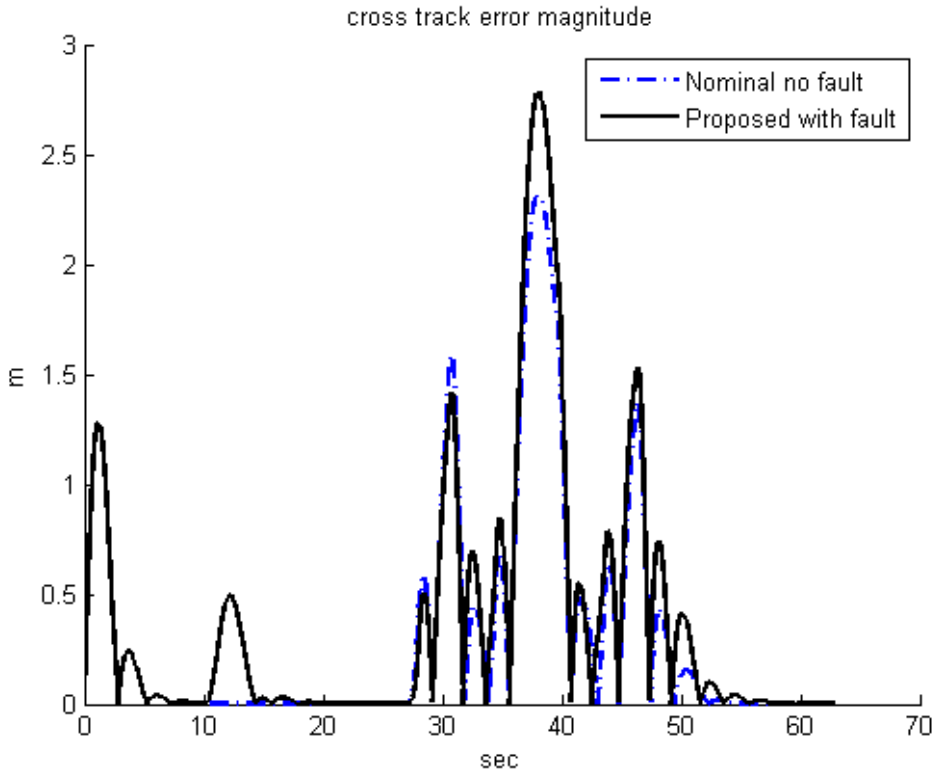

Figure 6.13: Guidance cross track error of simulations with right aileron fault of 7 deg and fault free case

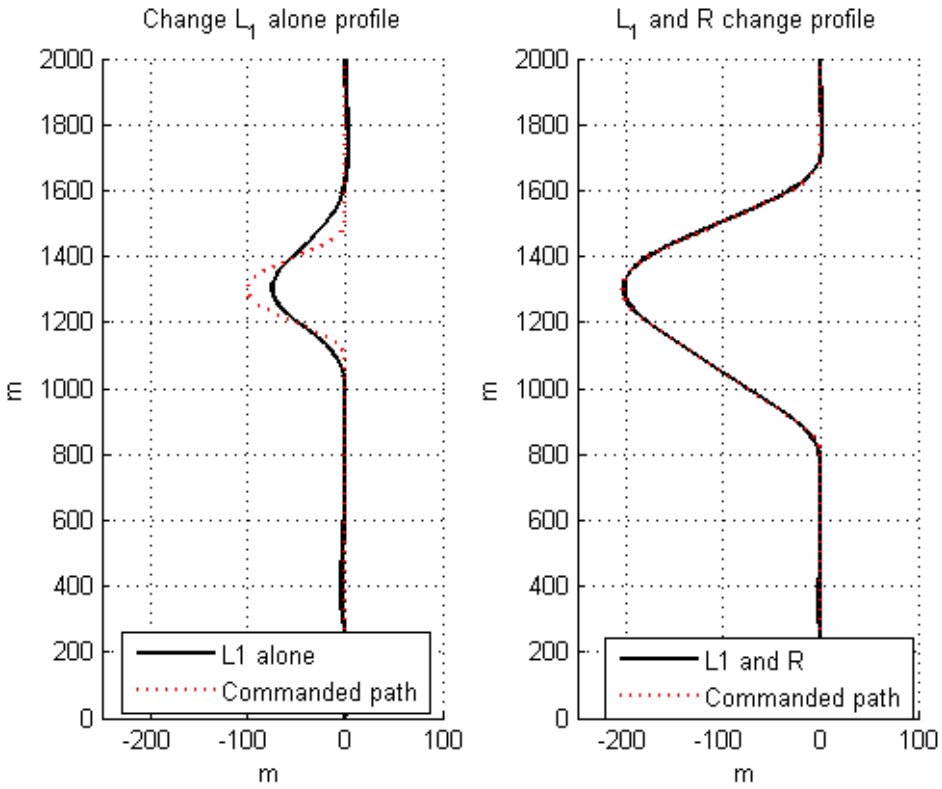

Figure 6.14: UAV paths profile of simulations with very severe right aileron fault of 16 deg with change in lookahead distance alone at left hand side, and together with change in path's radius on right hand side

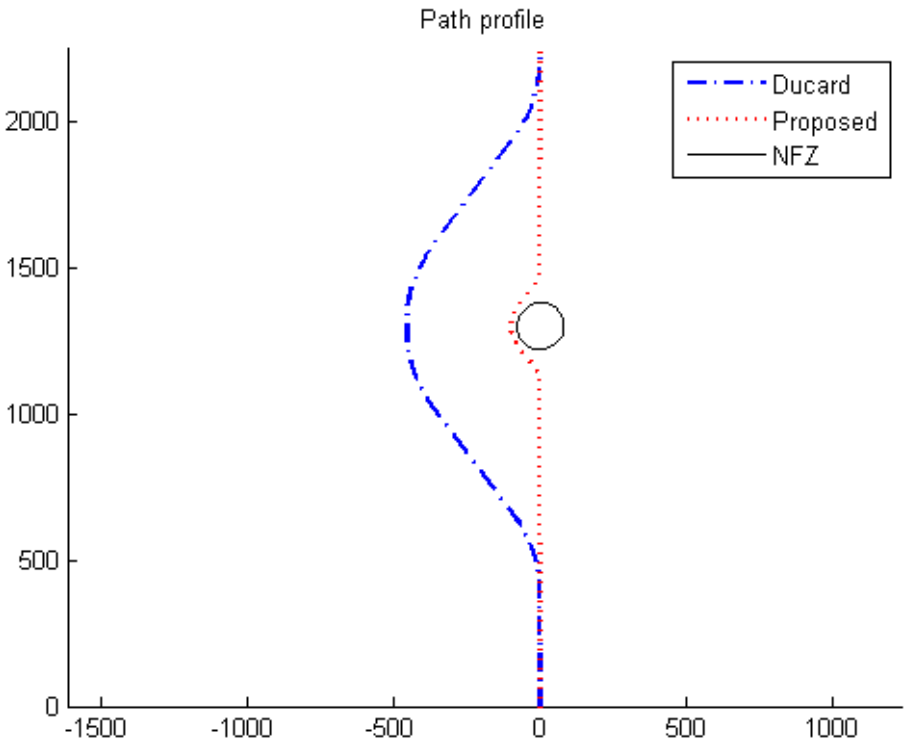

Figure 6.15: UAV path profiles taken by the two methods for an 12 deg aileron fault

6.6 Conclusions

Actuator faults reduce the amount of control authority remaining in the UAV system. This leads to the degradation in the UAV manoeuvring capability, which may impact its ability of follow particular path profile required to successfully complete its mission. Many schemes have focused on fault tolerant control to manage the fault degradation. However, fault tolerant guidance may be deployed to further enhance the mission success rate, or when fault tolerant control may not be sufficient alone.

This research proposes a guidance reconfiguration method to adapt the guidance scheme to stuck aileron actuator fault. By accounting for the fault degradation, the guidance virtual target distance may be changed to be compatible with the remaining healthy actuator capabilities. Stability criteria are suggested to keep the guidance scheme feasible. The impact of the guidance dynamics by the guidance adaptation is also described in changes to the damping ratio and bandwidth. One advantage of this method is that it does not change the guidance virtual target distance for minor actuator faults when the system's remaining capabilities are adequate to avoid unnecessary performance degradation.

Simulations using an Aerosonde UAV model shows improved guidance path tracking performance using the proposed method with large actuator faults. Also, simulation results comparing between the nominal fault free case and with minor actuator fault show the system is able to adequately track the path without causing the remaining healthy actuators to saturate. By adapting both the path following lookahead distance and the path's radius to accommodate the fault degradation, the proposed method's additional degree of freedom in the choice allows for a smaller turning radius. Minor faults or less severe faults may be accommodated by increasing the lookahead distance. This potentially provides more flexibility in the system design. In a No-Fly-Zone (NFZ) flight path simulation application, the proposed method is shown to be less restrictive in the path's change required to the same fault compared to another guidance reconfiguration scheme in Ref. [9]. Overall, this fault tolerant guidance may be a possible approach to improve the mission success rate with low computational requirements.

Chapter 7

Conclusions

This research journey started with the author's interest in getting UAVs to be more reliable from a system's perspective. Starting from the causal factors of UAV losses, fault tolerant control is identified as a means to manage the severe faults before they degrade the system sufficiently to cause system failure. The aim of this research is to improve the ability of UAVs to handle actuator faults. In particular, focus is placed on the mitigation of extremal cases via integrated or combined approaches to improve the performance of the degraded system. The crux of fault tolerant control in handling the fault is the availability of redundancy to accommodate it. In the absence of or limited redundancy, strategies may need to be developed to enable safe recovery.

In many areas, the integration and miniaturization of many components to form systems within systems have brought about the new capabilities and improvements. This idea is explored with an integrated actuator control scheme combining control allocation in the actuation system. Initial research has resulted in theoretical proof of the superiority of the proposed actuator scheme being shown for a four fin tail controlled UAV when one and two actuators are subjected to input fault. The superiority comes from the automatic utilization of the analytical redundancy by the controller in this new structure.

Besides superiority in performance, the new actuator control scheme utilizes fewer controllers when compared to conventional schemes. This may lead to cheaper implementation realizations. Also, the proposed scheme is analytically shown to be equivalent in nominal operating conditions during initial research, which may require less design effort when one would like to switch from existing conventional scheme to the proposed scheme.

Using discrete time domain analysis, geometrical insights into the proposed scheme is obtained. The geometrical insight shows the improved performance of the proposed scheme is due to its orthogonal update direction with respect to the actuator solution space. However, a potential shortcoming of the proposed scheme for flying platforms is that it may cause larger trim values. Two possible alleviation methods are proposed to reduce the larger trim values. Simulations

are conducted which verified the analysis done. Overall, a deep understanding of the proposed scheme is achieved.

The integrated actuator control scheme is applied on a typical small Aerosonde UAV model for evaluation at system level. The integrated scheme is able to improve on the stuck fault tolerant range over the conventional scheme and results in Ref. [7] by 23%. The simulation results also showed the conventional scheme incurring a larger stuck trim sideslip angle with a stuck ruddervator fault compared to the proposed scheme. This translates to better cross track error performance by a nonlinear guidance scheme to follow a path as its mission. The guidance performance improvements come about due to better flight regulation performance by the proposed scheme which alleviates the fault degradation effects.

To improve mission safety and performance with actuator faults, a guidance reconfiguration method is proposed for a common path following mission. By adapting both the path planner and path following algorithms' parameters, larger faults may be better accommodated. Also, increased flexibility is afforded by the method compared to other schemes. Stability criteria are suggested, and simulations with the Aerosonde UAV model demonstrated the proposed scheme's properties for fault accommodation.

Overall, this research provided new insights into the handling of actuator faults in UAVs severely constrained in control authority. The research may be particularly relevant for small UAVs which could be limited by system constraints to provide for the redundancy needed for handling the fault. In today's context of moving towards UAV operations in non-segregated airspace, a UAV's ability to conduct safe operations reliably is crucial. This implies the ability to handle safety critical component failures becoming important. In some UAVs, the system's ability to provide the redundancy may be constrained by the platform. This may especially be so for control surface actuators, where the placing or availability of the actuators is limited. New schemes which alleviate the performance degradation due to actuator faults are proposed here. The impact of this work is to expand the possibilities a designer may have for improving the fault tolerance capabilities of a platform should other alternatives not be feasible.

Chapter 8

Future Work

With each new understanding and discovery come new exciting possibilities. This work is no exception. The following aims to detail some of the possibilities to enhance the work presented thus far.

In this work, an integrated actuator control scheme has been developed. Analysis and simulations using a tail-fin controlled UAV verified the smaller error dynamics, and studies into the geometrical interpretations enhanced performance are obtained. In the case when control authority is available, the proposed scheme may reduce the performance degradation. In the case when redundancy is not available, more research may be conducted to understand and devise possibilities to conduct safe emergency recovery for such dire situations.

Although the proposed scheme is shown to improve on the stuck fault tolerance of the system, the overall remaining capability of the system may still be overwhelmed by excessive guidance demands. One direction for future work is to research on fault tolerant guidance schemes which can accommodate the limited actuator capabilities post fault.

This has led to initial work on reconfigurable guidance to accommodate the actuator fault degradation effects. The proposed reconfiguration method utilizes both guidance modules to provide the flexibility and ability to accommodate the fault. Although the proposed work has shown to improve the overall stuck fault handling capability of the degraded system post fault, the extremal stuck fault cases still pose a formidable problem. Future research may explore tighter integration between guidance and flight control modules for possibilities to extract and enhance the fault handling capabilities.

A potential obstacle to the usage of integrated schemes in practical flight applications is the increased complexity in the verification and validation (V&V) process needed before flight. This is due to the increased complexity of the integrated module compared to the often smaller standalone modules. Research into this aspect may bring about increased knowledge and confidence in practitioners deploying such approaches in future.

References

[1] R. Schaefer. Unmanned aerial vehicle reliability study. *Office of the Secretary of Defense, Washington, DC*, 2003.

[2] K. Dalamagkidis, KP Valavanis, and LA Piegl. On unmanned aircraft systems issues, challenges and operational restrictions preventing integration into the national airspace system. *Progress in Aerospace Sciences*, 44(7-8):503–519, 2008.

[3] S.A. Cambone and OFFICE OF THE SECRETARY OF DEFENSE WASHINGTON DC. *Unmanned aircraft systems roadmap 2005-2030*. Defense Technical Information Center, 2005.

[4] Y. Zhang and J. Jiang. Bibliographical review on reconfigurable fault-tolerant control systems. *Annual Reviews in Control*, 32(2):229–252, 2008.

[5] J.D. BosÌŇkovicÌĄ and R.K. Mehra. A decentralized fault-tolerant control system for accommodation of failures in higher-order flight control actuators. *Control Systems Technology, IEEE Transactions on*, 18(5):1103–1115, 2010.

[6] P. Goupil. Airbus state of the art and practices on fdi and ftc in flight control system. *Control Engineering Practice*, 2011.

[7] F. Bateman, H. Noura, and M. Ouladsine. Fault diagnosis and fault-tolerant control strategy for the aerosonde uav. *Aerospace and Electronic Systems, IEEE Transactions on*, 47(3):2119–2137, 2011.

[8] F. Bateman, H. Noura, and M. Ouladsine. Active fault diagnosis and major actuator failure accommodation: Application to a uav.

[9] G. Ducard. *Fault-tolerant flight control and guidance systems: practical methods for small unmanned aerial vehicles*. Springer Verlag, 2009.

[10] R. Isermann. *Fault-diagnosis systems: an introduction from fault detection to fault tolerance*. Springer Verlag, 2006.

[11] M. Blanke. *Diagnosis and fault-tolerant control*. Springer Verlag, 2003.

[12] S.X. Ding. *Model-based fault diagnosis techniques: design schemes, algorithms, and tools*. Springer Verlag, 2008.

[13] Jie Chen and Ron J Patton. *Robust model-based fault diagnosis for dynamic systems*. Kluwer academic publishers, 1999.

[14] Mogens Blanke, Roozbeh Izadi-Zamanabadi, Søren A Bøgh, and Charlotte P Lunau. Fault-tolerant control systems- a holistic view. *Control Engineering Practice*, 5(5):693–702, 1997.

[15] M. Pecht. *Prognostics and health management of electronics*. Wiley Online Library, 2008.

[16] O. Benedettini, TS Baines, HW Lightfoot, and RM Greenough. State-of-the-art in integrated vehicle health management. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 223(2):157–170, 2009.

[17] WH Prosser, TL Brown, SE Woodard, GA Fleming, and EG Cooper. Sensor technology for integrated vehicle health management of aerospace vehicles. *Strain*, 12000(6000):0, 2002.

[18] K. Keller, D. Wiegand, K. Swearingen, C. Reisig, S. Black, A. Gillis, and M. Vandernoot. An architecture to implement integrated vehicle health management systems. In *AUTOTESTCON Proceedings, 2001. IEEE Systems Readiness Technology Conference*, pages 2–15. IEEE, 2001.

[19] D. Huston, B. Esser, G. Spencer, D. Burns, and E. Kahn. Hierarchical actuator systems. In *Smart Structures and Materials*, pages 311–319. International Society for Optics and Photonics, 2005.

[20] Thomas C Corke, C Lon Enloe, and Stephen P Wilkinson. Dielectric barrier discharge plasma actuators for flow control. *Annual Review of Fluid Mechanics*, 42:505–529, 2010.

[21] George C Zimbru, Woo Ho Lee, and Dan O Popa. Design and fabrication of microflap actuators for steering of micro air vehicles. In *SPIE Defense, Security, and Sensing*, pages 73180R–73180R. International Society for Optics and Photonics, 2009.

[22] Mehul P Patel, Terry T Ng, Srikanth Vasudevan, Thomas C Corke, and Chuan He. Plasma actuators for hingeless aerodynamic control of an unmanned air vehicle. *Journal of Aircraft*, 44(4):1264–1274, 2007.

[23] Onur Bilgen, Kevin B Kochersberger, Daniel J Inman, and Osgar J Ohanian. Novel, bidirectional, variable-camber airfoil via macro-fiber composite actuators. *Journal of Aircraft*, 47(1):303–314, 2010.

[24] PK Menon, GD Sweriduk, EJ Ohlmeyer, and DS Malyevac. Integrated guidance and control of moving mass actuated kinetic warheads. Technical report, DTIC Document, 2002.

[25] PK Menon, GD Sweriduk, and EJ Ohlmeyer. Optimal fixed-interval integrated guidance-control laws for hit-to-kill missiles. Technical report, DTIC Document, 2003.

[26] J.D. Schierman, D.G. Ward, J.R. Hull, N. Gandhi, M.W. Oppenheimer, and D.B. Doman. Integrated adaptive guidance and control for re-entry vehicles with flight-test results. *Journal of Guidance Control and Dynamics*, 27:975–988, 2004.

[27] B.S. Kim, A.J. Calise, and R.J. Sattigeri. Adaptive, integrated guidance and control design for line-of-sight based formation flight. 2006.

[28] Takeshi Yamasaki, Hiroyuki Takano, and Yoriaki Baba. Robust path-following for uav using pure pursuit guidance. *Aerial Vehicles*, pages 671–690, 2009.

[29] A. Zolghadri. Advanced model-based fdir techniques for aerospace systems: Today challenges and opportunities. *Progress in Aerospace Sciences*, 2012.

[30] S.H. Kim, Y.I. Lee, and M.J. Tahk. New structure for an aerodynamic fin control system for tail fin-controlled stt missiles. *Journal of Aerospace Engineering*, 24(4), 2011.

[31] J.J. Gertler. Survey of model-based failure detection and isolation in complex plants. *Control Systems Magazine, IEEE*, 8(6):3–11, 1988.

[32] I. Hwang, S. Kim, Y. Kim, and C.E. Seah. A survey of fault detection, isolation, and reconfiguration methods. *Control Systems Technology, IEEE Transactions on*, 18(3):636–653, 2010.

[33] J. Jiang. Fault-tolerant control systemsâĂŤan introductory overview. *Acta Automatica Sinica*, 31(1):161–174, 2005.

[34] P.M. Frank. Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy: A survey and some new results. *Automatica*, 26(3):459–474, 1990.

[35] M. Verhaegen, S. Kanev, R. Hallouzi, C. Jones, J. Maciejowski, and H. Smail. Fault tolerant flight control-a survey. *Fault Tolerant Flight Control*, pages 47–89, 2010.

[36] I. Lopez and N. Sarigul-Klijn. A review of uncertainty in flight vehicle structural damage monitoring, diagnosis and control: Challenges and opportunities. *Progress in Aerospace Sciences*, 46(7):247–273, 2010.

[37] T. Kurtoglu, S.B. Johnson, E. Barszcz, J.R. Johnson, and P.I. Robinson. Integrating system health management into the early design of aerospace systems using functional fault analysis. In *Prognostics and Health Management, 2008. PHM 2008. International Conference on*, pages 1–11. IEEE, 2008.

[38] D.S. Bernstein and A.N. Michel. A chronological bibliography on saturating actuators. *International Journal of robust and nonlinear control*, 5(5):375–380, 1995.

[39] R. Patton. Robustness issues in fault-tolerant control. In *Fault Diagnosis and Control System Reconfiguration, IEE Colloquium on*, pages 1–1. IET, 1993.

[40] Fang Liao, Jian Liang Wang, and Guang-Hong Yang. Reliable robust flight tracking control: an lmi approach. *Control Systems Technology, IEEE Transactions on*, 10(1):76–89, 2002.

[41] BA White. Flight control of a vstol aircraft using polynomial eigenstructure assignment. In *Control'96, UKACC International Conference on (Conf. Publ. No. 427)*, volume 2, pages 758–763. IET, 1996.

[42] BA White. Robust polynomial eigenstructure assignment using dynamic feedback controllers. *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, 211(1):35–51, 1997.

[43] T. Lombaerts, D. Joosten, H. Smaili, and J. Breeman. Assessment criteria as specifications for reconfiguring flight control. *Fault Tolerant Flight Control*, pages 223–243, 2010.

[44] Venkat Venkatasubramanian, Raghunathan Rengaswamy, Kewen Yin, and Surya N. Kavuri. A review of process fault detection and diagnosis: Part i: Quantitative model-based methods. *Computers & Chemical Engineering*, 27(3):293 – 311, 2003.

[45] Venkat Venkatasubramanian, Raghunathan Rengaswamy, and Surya N Kavuri. A review of process fault detection and diagnosis: Part ii: Qualitative models and search strategies. *Computers & Chemical Engineering*, 27(3):313 – 326, 2003.

[46] V. Venkatasubramanian, R. Rengaswamy, S.N. Kavuri, and K. Yin. A review of process fault detection and diagnosis: Part iii: Process history based methods. *Computers & chemical engineering*, 27(3):327–346, 2003.

[47] A. Gelb. *Applied optimal estimation*. MIT press, 1999.

[48] C. Edwards, H. Alwi, and C.P. Tan. Sliding mode methods for fault detection and fault tolerant control with application to aerospace systems. *International Journal of Applied Mathematics and Computer Science*, 22(1):109–124, 2012.

[49] P. Eide and P. Maybeck. An mmae failure detection system for the f-16. *Aerospace and Electronic Systems, IEEE Transactions on*, 32(3):1125–1136, 1996.

[50] Y. Zhang and J. Jiang. Integrated active fault-tolerant control using imm approach. *Aerospace and Electronic Systems, IEEE Transactions on*, 37(4):1221–1235, 2001.

[51] J.D. Boskovic and R.K. Mehra. A multiple model-based reconfigurable flight control system design. In *Decision and Control, 1998. Proceedings of the 37th IEEE Conference on*, volume 4, pages 4503–4508. IEEE, 1998.

[52] E. Chow and A. Willsky. Analytical redundancy and the design of robust failure detection systems. *Automatic Control, IEEE Transactions on*, 29(7):603–614, 1984.

[53] R.J. Patton and J. Chen. Review of parity space approaches to fault diagnosis for aerospace systems. *Journal of Guidance Control Dynamics*, 17:278–285, 1994.

[54] J. Gertler. Fault detection and isolation using parity relations. *Control Engineering Practice*, 5(5):653–661, 1997.

[55] M.R. Napolitano, Y. Song, and B. Seanor. On-line parameter estimation for restructurable flight control systems. *Aircraft Design*, 4(1):19–50, 2001.

[56] C. Kamali, AA Pashilkar, and JR Raol. Evaluation of recursive least squares algorithm for parameter estimation in aircraft real time applications. *Aerospace Science and Technology*, 15(3):165–174, 2011.

[57] G. Chowdhary and R. Jategaonkar. Aerodynamic parameter estimation from flight data applying extended and unscented kalman filter. In *AIAA Atmospheric Flight Mechanics Conference*. Citeseer, 2006.

[58] X. Yu and J. Jiang. Hybrid fault-tolerant flight control system design against partial actuator failures. *Control Systems Technology, IEEE Transactions on*, 20(4):871 –886, july 2012.

[59] S. Kim, J. Choi, and Y. Kim. Fault detection and diagnosis of aircraft actuators using fuzzy-tuning imm filter. *Aerospace and Electronic Systems, IEEE Transactions on*, 44(3):940–952, 2008.

[60] Kumpati S Narendra, Jevendran Balakrishnan, and M Kemal Ciliz. Adaptation and learning using multiple models, switching, and tuning. *Control Systems, IEEE*, 15(3):37–51, 1995.

[61] Jovan D Boskovic and Raman K Mehra. Stable multiple model adaptive flight control for accommodation of a large class of control effector failures. In *American Control Conference, 1999. Proceedings of the 1999*, volume 3, pages 1920–1924. IEEE, 1999.

[62] TJJ Lombaerts, QP Chu, JA Mulder, and DA Joosten. Modular flight control reconfiguration design and simulation. *Control Engineering Practice*, 2011.

[63] M. Athans, D. Castanon, K. Dunn, C. Greene, W. Lee, N. Sandell Jr, and A. Willsky. The stochastic control of the f-8c aircraft using a multiple model adaptive control (mmac) method–part i: Equilibrium flight. *Automatic Control, IEEE Transactions on*, 22(5):768–780, 1977.

[64] K.S. Narendra and J. Balakrishnan. Adaptive control using multiple models. *Automatic Control, IEEE Transactions on*, 42(2):171–187, 1997.

[65] F. Sharifi, M. Mirzaei, B.W. Gordon, and Y. Zhang. Fault tolerant control of a quadrotor uav using sliding mode control. In *Control and Fault-Tolerant Systems (SysTol), 2010 Conference on*, pages 239–244. IEEE, 2010.

[66] S. Bouabdallah and R. Siegwart. Full control of a quadrotor. In *Intelligent robots and systems, 2007. IROS 2007. IEEE/RSJ international conference on*, pages 153–158. Ieee, 2007.

[67] Hafid Smaili, Jan Breeman, Thomas Lombaerts, and Diederick Joosten. Recover: a benchmark for integrated fault tolerant flight control evaluation. In *Fault Tolerant Flight Control*, pages 171–221. Springer, 2010.

[68] M.W. Oppenheimer, D.B. Doman, and M.A. Bolender. Control allocation for over-actuated systems. In *Control and Automation, 2006. MED'06. 14th Mediterranean Conference on*, pages 1–6. IEEE, 2006.

[69] M. Bodson. Evaluation of optimization methods for control allocation. *Journal of Guidance, Control, and Dynamics*, 25(4):703–711, 2002.

[70] A.B. Page and M.L. Steinberg. A closed-loop comparison of control allocation methods. In *Proceedings of the 2000 Guidance, Navigation and Control Conference*, 2000.

[71] M. Steinberg. Historical overview of research in reconfigurable flight control. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 219(4):263–275, 2005.

[72] J.F. Buffington. Modular control law design for the innovative control effectors (ice) tailless fighter aircraft configuration 101-3, 1999.

[73] O. Härkegård and S.T. Glad. Resolving actuator redundancyâĂŤoptimal control vs. control allocation. *Automatica*, 41(1):137–144, 2005.

[74] Tor A Johansen and Thor I Fossen. Control allocation-a survey. *Automatica*, 49(5):1087–1103, 2013.

[75] Guillaume Ducard, Hans P Geering, and Emil Dumitrescu. Efficient control allocation for fault tolerant embedded systems on small autonomous aircrafts. In *Industrial Embedded Systems, 2006. IES'06. International Symposium on*, pages 1–10. IEEE, 2006.

[76] L. Zaccarian. Dynamic allocation for input redundant control systems. *Automatica*, 45(6):1431–1438, 2009.

[77] Qing Xu, Hao Yang, Bin Jiang, Donghua Zhou, and Youmin Zhang. Adaptive fault-tolerant control design for uavs formation flight under actuator faults. In *Unmanned Aircraft Systems (ICUAS), 2013 International Conference on*, pages 1097–1105. IEEE, 2013.

[78] SM Azizi and K Khorasani. Cooperative actuator fault accommodation in formation flight of unmanned vehicles using relative measurements. *International Journal of Control*, 84(5):876–894, 2011.

[79] A Chamseddine, Youmin Zhang, and CA Rabbath. Trajectory planning and re-planning for fault tolerant formation flight control of quadrotor unmanned aerial vehicles. In *American Control Conference (ACC), 2012*, pages 3291–3296. IEEE, 2012.

[80] J.H. Blakelock. *Automatic control of aircraft and missiles*. Wiley-Interscience, 1991.

[81] O. Harkegard. Dynamic control allocation using constrained quadratic programming. *Journal of Guidance Control and Dynamics*, 27(6):1028–1034, 2004.

[82] J.A.M. Petersen and M. Bodson. Constrained quadratic programming techniques for control allocation. *Control Systems Technology, IEEE Transactions on*, 14(1):91–98, 2006.

[83] W. Durham. Constrained control allocation. *Journal of Guidance, control, and Dynamics*, 16(4):717–725, 1993.

[84] H. Alwi and C. Edwards. Fault tolerant sliding modes control allocation with control surface priority weighting. In *Control Applications (CCA), 2010 IEEE International Conference on*, pages 1057–1062. IEEE, 2010.

[85] Y. Zhang, V.S. Suresh, B. Jiang, and D. Theilliol. Reconfigurable control allocation against aircraft control effector failures. In *Control Applications, 2007. CCA 2007. IEEE International Conference on*, pages 1197–1202. IEEE, 2007.

[86] Chang How Lo, Hyo-Sang Shin, Antonios Tsourdos, and Seung-Hwan Kim. Improving the performance of an actuator control scheme during saturation. In *Advances in Aerospace Guidance, Navigation and Control*, pages 15–27. Springer, 2013.

[87] James Hardy Wilkinson. *The algebraic eigenvalue problem*, volume 87.

[88] R.C. Dorf and R.H. Bishop. *Modern control systems*. Prentice Hall, 2001.

[89] C.H. Lo, H.S. Shin, A. Tsourdos, and S.H. Kim. Improving the performance of an actuator control scheme during saturation. In *Euro GNC 2013 âĂŞ 2nd CEAS Specialist Conference on Guidance, Navigation & Control, Delft*, pages 204–216, 2013.

[90] Lester E Dubins. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *American Journal of mathematics*, pages 497–516, 1957.

[91] Sanghyuk Park, John Deyst, and Jonathan P How. Performance and lyapunov stability of a nonlinear path following guidance method. *Journal of Guidance, Control, and Dynamics*, 30(6):1718–1728, 2007.

[92] PB Sujit, Srikanth Saripalli, and JB Sousa. An evaluation of uav path following algorithms. In *Control Conference (ECC), 2013 European*, pages 3332–3337. IEEE, 2013.

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