NDI – BASED NEUROCONTROLLER FOR UNMANNED COMBAT AERIAL VEHICLES DURING AERIAL REFUELLING

Aarti Panday

A research report submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, in partial fulfilment of the requirements for the degree of Master of Science in Engineering

Johannesburg, 2008

Declaration

I declare that this research report is my own unaided work and all other material has been referenced. It is being submitted for the Degree of Master of Science at the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination at any other University.

_____

Aarti Panday

_____day of _____(year)_____

Abstract

The success of Unmanned Combat Aerial Vehicles (UCAVs) requires further developments in the field of automated aerial refuelling (AAR) and control systems. AAR aircraft models identified thus far do not take the centre of gravity (cg) position movement into account during refuelling. A six-degree-of-freedom aircraft model was combined with a moving cg model for refuelling. The equations of motion for the aircraft in flight refuelling showed the aircraft dynamics to be coupled in the longitudinal and lateral-directional planes when the cg had moved away from the reference point. Applying assumptions specific to the flight conditions, simplified equations of motion were derived. Modal analysis of four cases for the linearised aircraft model during aerial refuelling was conducted. This revealed that the increase in mass was favourable to the stability of the Dutch Roll mode, but the mode did become more oscillatory initially as mass was increased, but as the cg moved forward, the mode became less oscillatory. The opposite was observed with the Phugoid mode. The Short Period Oscillation (SPO) decomposed into two first order modes during refuelling and these remained unchanged during the refuelling process. Three radial basis function (RBF) neural networks (RBFNN) were developed and trained to approximate the inverse plant dynamics and predicted commanded deflections of the elevator, aileron and rudder. Training data required for the network was randomly generated and the desired rates and commanded control surface deflections were computed. The training error was the smallest in the elevator deflection required during refuelling. A basic nonlinear dynamic inversion (NDI) controller without a neural network (NN) was designed for the aircraft. The performance of this controller was not satisfactory. The RBF was combined with the NDI to form a RBFNN-based controller. The longitudinal NDI RBFNN-based controller was less sensitive to modelling errors than the base NDI controller. The lateral NDI RBFNN-based controller's performance was worse than the longitudinal controller, but showed potential as a technique for future consideration. Including the variation of aircraft inertia in the model has been recommended as further work, as well as exploring other neural network topologies in the NDI NN controller.

Acknowledgements

I would like to thank my supervisor, Dr. J.O. Pedro, without whose dedication, this project would not have been successful. Dr. Pedro's passion and enthusiasm in the field of aircraft control has been a great motivation for me personally, and I would like to thank him for all the assistance in this project.

Publications Arising from Study

The following two papers arising from this study were presented at the International Conference on Modelling and Optimisation of Structures, Processes and Systems on 22 – 24 January 2007 in Durban, South Africa:

Pedro, J.O. and Panday, A. (2007), "Trim and Stability Analysis of a UCAV during In-Flight Refuelling", *CD-ROM Proceedings of International Conference on Modelling and Optimisation of Structures, Processes and Systems,* Durban, South Africa, ISBN: 1-86840-643-1

Panday, A. and Pedro, J.O. (2007), "Neural Network Control of a UCAV during In-Flight Refuelling", *CD-ROM Proceedings of International Conference on Modelling and Optimisation of Structures, Processes and Systems*, Durban, South Africa, ISBN: 1-86840-643-1

Table of Contents

List of Figures

List of Tables

List of Acronyms

AAR	automated aerial refuelling
BAC	backstepping adaptive controller
DI	dynamic inversion controller
FLC	fuzzy logic controller
GRBFN	growing radial basis function network
HALE	high altitude long endurance
IAC	indirect adaptive controller
IFCS	intelligent flight control system
ISA	international standard atmosphere
MIMO	multi input multi output
MPC	model predictive controller
MRDANC	model reference direct adaptive neural control
NN	neural network
NNC	neural network controller
PID	proportional-integral-derivative
RBF	radial basis function
RBFNN	radial basis function neural network
SEAD	suppression of enemy air defences
SPO	short period oscillation
UAV	unmanned aerial vehicle
UCAV	unmanned combat aerial vehicle
US	United States
USAF	United States Air Force
VSC	variable structure controller

List of Symbols

A	system matrix
b	wing span [m]
$\bar{\mathbf{c}}$	mean aerodynamic chord [m]
cg	centre of gravity [-]
C_l, C_m, C_n	nondimensional rolling, pitching and yawing moment coefficients [-]
C_x, C_y, C_z	nondimensional force coefficient [-]
f	nonlinear state distribution function
g	acceleration due to gravity [$m.s^{-2}$]
g	nonlinear control distribution function
h	altitude [m]
I_T	engine spinning rotors moment of inertia [$kg.m^2$]
I_x I_y I_z	roll, pitch, and yaw moments of inertia [$kg.m^2$]
I_{x0} I_{y0} I_{z0}	initial roll, pitch and yaw moments of inertia [$kg.m^2$]
I_{xz} I_{xy} I_{yz}	products of inertia [$kg.m^2$]
I_{xz0} I_{xy0} I_{yz0}	initial products of inertia [$kg.m^2$]
L, M, N	total rolling pitching yawing moment [N.m]
L_p	rolling moment derivative w.r.t. roll rate [N.m.s/deg]
L_r	rolling moment derivative w.r.t. yaw rate [N.m.s/deg]
L_β	rolling moment derivative w.r.t. sideslip angle [N.m/deg]
$L_{\delta a}$	rolling moment derivative w.r.t. aileron deflection [N.m/deg]
$L_{\delta r}$	rolling moment derivative w.r.t. rudder deflection [N.m/deg]
m	aircraft mass [kg]
M_q	pitching moment derivative w.r.t. pitch rate[N.m/deg]
M_α	pitching moment derivative w.r.t. angle of attack [N.m/deg]
$M_{\delta e}$	pitching moment derivative w.r.t. elevator deflection[N.m/deg]
N_p	yawing moment derivative w.r.t. roll rate[N.m.s/deg]
N_r	yawing moment derivative w.r.t. yaw rate [N.m.s/deg]
N_β	yawing moment derivative w.r.t. sideslip angle [N.m/deg]
$N_{\delta a}$	yawing moment derivative w.r.t. aileron deflection [N.m/deg]
$N_{\delta r}$	yawing moment derivative w.r.t. rudder deflection [N.m/deg]
p, q, r	aircraft roll pitch yaw rate [deg/s]
$\mathbf{r_c} = [x_C, y_C, z_C]^T$	position displacement vector w.r.t the origin of Oxyz [m]

$\mathbf{r_T} = [x_T, y_T, z_T]^T$	engine position vector w.r.t. the origin of Oxyz
S	wing reference area [m^2]
S_x, S_y, S_z	product of mass and displacement of centre of gravity from reference [kg.m]
T	thrust force [N]
T	period [s]
t_s	settling time [s]
u v w	aircraft speed in X Y Z direction [m/s]
u	control vector
V	total speed [m/s]
w	disturbance vector
x	state vector
$\mathbf{x_o}$	system eigenvector
x_E, y_E, z_E	aircraft displacement in X Y Z direction relative to Earth-fixed axes[m]
y	output vector
X_{aero} Y_{aero} Z_{aero}	aerodynamic component of force X Y Z [N]
X_{total} Y_{total} Z_{total}	total force in X Y Z direction [N]

Greek Symbols

α	angle of attack [deg]
β	sideslip angle [deg]
δ_a	aileron deflection [deg]
δ_e	elevator deflection [deg]
δ_r	rudder deflection [deg]
δ_T	throttle deflection [-]
δ_{Tmax}	maximum throttle deflection [-]
$\mathbf{\Delta f_c}$	vector incremental forces and moments
γ	flight path angle [deg]
λ	system eigenvalues
ω_d	damped frequency [rad/s]
ω_T	angular speed [rad/s]
ϕ, θ, ψ	Euler angles [deg]
ρ	air density [$kg.m^{-3}$]
ρ_0	air density at sea level [kg/m^3]

τ	time constant [s]
φ_{TY}, φ_{TZ}	angular orientation of the engine w.r.t. the origin of Oxyz [deg]
ζ	damping ratio [-]

1 Introduction

1.1 Background and Motivation

1.1.1 Unmanned Combat Aerial Vehicles

Attempts at unmanned flight were made before the Wright brothers designed their successful piloted vehicle. Since the Wright brothers' success in 1903, manned flight has dominated the aviation scene. Alongside manned airplane development, unmanned airplane development continued, but was not given the same detailed attention as manned aircraft design. It took many years for the merit in an air force having a successful unmanned aerial vehicle (UAV) to be realised. (Clark, 2000)

In recent times, the Pioneer UAV (Figure 1.1) was used in tactical intelligence operations in Iraq in 1991. The Pioneer was successfully used for gunfire spotting. Since the first conflict in Iraq, the Pioneer has been used in operations in Haiti, Somalia, the Balkans, and Afghanistan (Sanders, 2003; Bone and Bolkom, 2003; Pardesi, 2005)

Figure 1.1: Pioneer UAV

In Afghanistan, the Predator UAV (Figure 1.2) started performing armed reconnaissance missions. Prior to this, it was used in Kosovo in 1995 for surveillance. In 2003, it was equipped with Hellfire air-to-ground missiles to destroy mobile radar units and to attack high-value targets in Iraq. Armed strike and close air support were new capabilities for the UAV. It has worked together with F/A-18 Hornets' to locate targets. The Predator UAV was also put to use during a rescue mission of a U.S. Army prisoner-of-war. (Pardesi, 2005; Bone and Bolkcom, 2003)

Figure 1.2: Predator UAV

Starting off as an experimental system, the Global Hawk UAV (Figure 1.3) made its debut in the skies over Afghanistan in 2001. It is one of the largest and most expensive UAVs produced to date. It is a high-altitude, long-endurance (HALE) platform, and became the first UAV to make a trans-Pacific flight from California to Australia. Flying at 65 000 ft, it provided more than 15 000 images in operations over Afghanistan. In Iraq, the Global Hawk's capabilities were highlighted when it was able to identify targets during severe sandstorms (Pardesi 2005; Bone and Bolkcom, 2003).

Figure 1.3: Global Hawk UAV

The cases cited above only prove that UAVs represent a transformational technology that has indeed changed the way wars have been fought in recent times. The effectiveness of UAVs in these modern conflicts has highlighted the immense advantage of having an unmanned platform as part of a military arsenal. Unmanned flight has come a long way since early developments from World War I through to the cutting edge systems available today UAV systems are now being developed in countries all over the world. Many air

forces throughout the world have actively taken steps in fully evaluating a UAV/UCAV concept (Wilson, 2005)

The evolution of unmanned platforms is apparent since roles other than intelligence, surveillance and reconnaissance were now being considered. One mission identified for UCAVs is the suppression of enemy air defences (SEAD) mission. This is considered to be the most dangerous mission for a manned fighter. The X-45, the first unmanned system specifically designed for combat operations, made aviation history on the 18th of April 2004 when it dropped a weapon from its internal weapons bay, hitting a ground target, while flying at 35 000 ft at a speed of Mach 0.67. Its missions include electronic attack, SEAD, intelligence, surveillance, reconnaissance, and deep strike. The next stage in this program would be multi-vehicle co-ordinated operations. The X-45 is an example of a *from-the-ground-up* UCAV, where the aircraft has been designed specifically for the purpose of unmanned combat. Other UCAV options include the converted concept, where traditional fighters such as the F-16 or F-117 are converted to UCAVs for the purpose of fulfilling the SEAD mission or for demonstrating enabling technologies. (Clark, 2000; Pardesi, 2005)

1.1.2 Automated Aerial Refuelling

Recently, significant efforts have been devoted to increase the flight endurance and payload of UAVs as is evident with the performance of the Global Hawk and Predator. Extended endurance has been achieved due to progress made in propulsion and aerodynamics without the need for a pilot in the system. One of the major limitations of UAVs currently, is the lack of an in-flight-refuelling capability as is present in manned aircraft. An air force having unmanned vehicles that can refuel in the air would have the advantage of a greatly extended endurance and the ability to respond to a threat thousands of miles away. Without a man in the cockpit and with an aerial refuelling capability, the endurance of the aircraft would only be limited by routine maintenance. The in-flight-refuelling capability will take away the extra time needed for the UAV to land, refuel and redeploy, and would greatly reduce the logistical trail involved in operation. This capability will allow the UCAV to take off with a larger payload thereby allowing new operational capabilities to be explored. The extended endurance offered by the aerial refuelling capability would give a battle ready UCAV endurance in weeks, months, and

possibly even years (Withrow, 2004; Fravolini et. al., 2003; Pardesi, 2005; Thompson, 1998; Clark, 2000; Vendra et. al,. 2007; Ollero and Merino, 2004; Jin et. al., 2006).

From a control point of view, manned in-flight-refuelling is known to be a difficult task for a pilot, and becomes greatly difficult when performed autonomously. Many years of testing is needed before in-flight-refuelling becomes a reality. NASA Dryden Flight Research Centre has initiated an automated aerial refuelling research project based on flight testing using a hose and drogue system. Flight data will be used in the development of a refuelling system. Also, flight tests will assist in setting up simulation models which will be used in the refuelling system development. The manoeuvres identified for flight test were specifically chosen to isolate the change in drogue position as a function of flight condition, hose and drogue type, tanker type and weight, receiver type and tanker-receiver manoeuvring. The next stage in this effort is to use the validated hose and drogue model in control law design. (Withrow, 2004; Hansen et. al., 2004).

1.1.3 Control

A study conducted in Australia identified that autonomous control is a core technology that will ensure success in future UAV/UCAV development. In a separate study, command and control were identified as technologies that will have to be further developed and are crucial to the success of future UCAVs. Airfoils, engines and weapons systems have been sufficiently developed and now the focus of technology development must change to these crucial areas. Shortcomings of command and control systems will lead to the failure of achieving a SEAD capability within a UCAV platform. The focus of research must now change so as to assist in building new, efficient and robust technologies that are required to make UCAVs operational as fast as possible (Clark, 2000; Wong and Bill, 1998).

Since the performance and success of UAVs depend on control; advanced and reliable control systems are needed. In a conventional aircraft controller, states are measured by sensors and are compared to desired states after conditioning. The control input required for the aircraft is based on the error between the measured and desired. The controller responds by supplying the aircraft with actuator control inputs to minimise the error. While this sounds like a relatively simple process, when implemented, tuning the

controller poses a challenge. Selection of the correct gains required by the controller affects the system response and stability. With degradation in controller performance, gains need to be re-tuned. Also, the gains are dependent on the operating conditions of the aircraft, should these change, new gains are required. Re-tuning controller gains is not practical and is time consuming (Soares et. al., 2006).

In addition, traditional controller design usually involves complex and extensive mathematical analysis, which implies high cost and cannot guarantee a good performance level in the whole flight envelope. Assuming linear and time invariant dynamics requires gains to be scheduled as functions of the nominal flight condition in a conventional flight controller. It could happen that in certain flight conditions, the performance of such systems can deteriorate due to unmodelled effects and nonlinearities present in the flight dynamics. Gain scheduling has been used to deal with nonlinear elements present in aircraft dynamics, provided the gain scheduling parameters vary slowly compared with the dynamic response of the aircraft. This process has worked well but with increased capabilities and performance requirements for modern aircraft, traditional controllers often do not offer acceptable performance (Reiner et. al., 1996; Gili and Battipede, 2001; Lane and Stengel, 1988).

Neural network and fuzzy logic control schemes have been put forward as a valid means to overcome most of the typical limitations of classical control techniques and to solve complex control problems. Adaptive control offers an alternative to completely re-designing a control system for these changes (Gili and Battipede, 2001; Dufrene, 2004; Soares et. al., 2006).

Nonlinear dynamic inversion control laws can also be used to overcome limitations by conventional controllers. The advantage provided by the NDI control law is the ability to directly command specific state variables. This controller has a more accurate representation of forces and moments due to large state and control perturbations. These controllers, however, are susceptible to errors arising from the inversion of the plant dynamics. Research has shown that neural networks can be used to overcome this issue in dynamic inversion controllers. The neural network cancels inversion errors by learning the approximate inverse plant dynamics (McFarland 2000; Lane and Stengel, 1988).

The NDI-NN scheme proved to be effective over a wide range of applications and is summarised (Gili and Battipede, 2001):

- systems operating in regimes characterized by highly nonlinear aerodynamics;
- systems displaying multi-time scale behaviour and thus rapidly varying nonlinear dynamics;
- systems characterized by a high degree of uncertainty; and
- systems demanding the maintenance of a certain level of handling qualities even after failures in actuation channels.

1.2 Literature Review

1.2.1 Modelling and Simulation of AAR

Aerial refuelling can be done in one of two ways – either with a flying boom or with a probe and drogue. In flying boom method, a stiff tube is manually operated from the tanker aircraft and the problem here is that the UCAV must be able to maintain a relative displacement with the tanker. The boom is docked by human operator onboard the tanker. In the probe and drogue option, the tanker lets out a flexible hose with the drogue at the end. The UAV has to insert the probe into the drogue for refuelling to take place. Again here, the problem is relative displacement and orientation (Pollini et. al., 2003).

Extensive simulation work is underway to ensure that AAR capability would be integrated on the X-45 UCAV. Wind tunnel tests have already been performed using a generic UAV with vehicles used in the USAF and Navy. Tests will soon be conducted with the X-45 in the wind tunnel (Blake, 2003). Side by side, flight demonstrations have been successfully conducted in evaluating the feasibility of certain technologies in UAV AAR applications. Data from flight tests will help to determine safe refuelling speeds, tanker interference and the use of new advanced methods for precise positioning. The data will also be used in developing a control algorithm specifically in AAR. (Withrow, 2004)

Pollini et. al. (2003) developed a simulation set-up of a virtual benchmark for AAR of UAVs using the probe and drogue method. The group presents a vision-based algorithm, which is capable of estimating the relative displacement and orientation of the UAV. In simulating the AAR, aerodynamic interference between the receiver and tanker was

modelled, as well as the flexibility of the refuelling hose, and the vision system equipment was taken into account. Aircraft modelling was based on simple point mass approximations with wake effects due to receiver and tanker incorporated. The AAR process was modelled as described due to the focus area being the vision system.

Two of the most significant factors that would affect the UAV dynamics during aerial refuelling are (Dogan et. al., 2005):

- time varying mass and inertia properties
- wind effect due to tanker's trailing vortices

Dogan et. al. (2005) set up a refuelling model that focused on the effect of the trailing wake vortex of the tanker on the receiver aircraft. The equations of motion in a uniform wind field were derived. But, the vortex induced wind field acting on the receiver aircraft is non-uniform. Thus, the non-uniform induced wind components and gradients were approximated by using equivalent uniform wind and gradients. This approximation allows for a computationally efficient implementation of the modelling of the close proximity effects of the receiver and tanker aircraft. The equations of motion of the receiver aircraft were derived in terms of the position and orientation of the receiver relative to the tanker aircraft. The velocity vector of the receiver was the sum of the velocity of the aircraft relative to the surrounding air and the velocity of the air relative to an inertial reference frame. The modelling of the AAR did not take the aircraft mass change during refuelling into account.

Ochi and Kominami (2005) conducted simulation and control studies for the case of automatic aerial refuelling using two methods: line-of-sight angle control, and proportional navigational control. The modelling of the AAR was based on linear aircraft modelling, but nonlinearity was introduced into the simulation via the kinematic and geometric relations between the receiver and tanker. Wind turbulence was modelled; however, effects of changing mass were ignored.

Fravolini et. al. (2003) focused on the control and design of modelling tools for AAR using nonlinear tanker (B747) and receiver (F-4) aircraft models. The research focused largely on the accurate modelling of the GPS and vision system, as such, standard rigid body, fixed mass aircraft modelling was considered with turbulence being included.

Considering safety and accidents, the study of Zhu and Meguid (2007) examined the dynamics of an aerial hose-and-drogue refuelling system focusing on cable tension, tow point disturbance and vortex wake effects due to the presence of the cable. A finite element method was used in this analysis, where the aerial refuelling hose-and-drogue system was idealized as a generalized aerial cable towed system with a prescribed motion at the tow point. As such the pertinent parameters identified in this study were the drag, weight and mechanical properties of the cable; the drag and weight of the towed body; the prescribed motion at the tow point; the disturbance at the towed body due to the coupling between the drogue and the probe, and the vortex wake behind the tanker aircraft.

A large effort has been channelled to specifically research issues related to finding the relative distance between the tanker and the UAV. A simulation study of the UAV aerial refuelling problem has been set up by Vendra et. al. (2007) and aspects of a machine vision based automated refuelling algorithm has been researched. A feature extraction algorithm which is used to detect and correctly identify features like corners on the tanker airframe was investigated. This was used to find the relative distance and orientation between the tanker and the UAV aircraft, assuming the position of the detected features in the tanker reference frame is constant and known. Dell'Aquila et. al. (2007) investigated a machine vision position sensing system for the problem of docking for UAV aerial refueling. The results indicate that it is possible to use image processing algorithms in real-time with off-the-shelf hardware to obtain accurate relative tanker-UAV position and orientation estimations. Point matching, as part of a machine vision based approach to aerial refueling, was investigated by Mammarella et. al., (2008). Algorithms were compared and implemented in a simulation environment specifically developed for the study of machine vision methods.

1.2.2 NDI and NN Control Techniques in the Context of Aircraft Applications

Hunt et. al. (1992) conducted a survey focusing on the use of neural networks in the fields of modelling, identification and control of nonlinear systems. It was identified that neural networks can be used in the field of control for the following reasons:

- Nonlinear systems – A neural network has the ability to approximate arbitrary nonlinear mapping.

- Parallel distributed processing – The highly parallel structure of a neural network allows parallel implementation. Thus, a neural network based scheme can be considered more fault tolerant than conventional schemes.
- Learning and adaptation – A neural network is trained using past data from the system. The network can be adapted on-line.
- Multivariable systems – A neural network is able to process many inputs and many outputs – very good application for multivariable systems.

A comparative study within the field of neural control for aircraft used seven control techniques with a scheduled dynamic inversion controller as the baseline (Steinberg, 2001). The study, amongst other objectives, sought to track a linear desired performance model for different types of single and multiaxis manoeuvres. The aircraft used in the simulation was the F-18. The controllers compared are:

- dynamic inversion (DI) controller,
- indirect adaptive controller (IAC),
- backstepping adaptive controller (BAC),
- neural network controller (NNC),
- variable structure controller (VSC),
- model predictive controller (MPC), and a
- fuzzy logic controller (FLC).

Results were divided into small, medium and large manoeuvres (eg. ϕ = 5º, 60 º and 180º). For medium manoeuvres, the controllers had comparable performance except the VSC and FLC which give worse results. The FLC converged the slowest to zero steady state error. In the large manoeuvres scenario, the FLC performs much better and the VSC still records the highest errors. The controller giving the best results was the MPC. The FLC, MPC, BAC and VSC controllers had increased actuator usage over the DI scheme. It was found that the DI controller was fairly robust and that it was difficult to develop another scheme that could exceed its performance overall. In the case of simulating flight failures, the adaptive approaches performed better than the conventional DI controller (Steinberg, 2001).

Li et. al. (2001) proposed a neurocontroller design for nonlinear fighter aircraft high angle of attack manoeuvre using a fully tuned radial basis function network. The study uses an on-line control method with a feedback-error-learning strategy where all the parameters

of the network will be updated, ensuring that the system dynamics are captured quickly. A growing radial basis function network (GRBFN) was used to approximate the inverse dynamics of the F-16 aircraft. A baseline control was established in the form of a conventional proportional controller based on a linearised model which gave good performance for the linear model. However, when implemented in the six-degree-of-freedom model, the tracking performance became poor. Using a RBF network with a fixed amount of neurons, the results were found to be inferior to that of the GRBFN. The GRBFN had fewer neurons than the fixed RBF network. Results showed that the neural control outputs change when commands are executed – this demonstrated the learning process of the network.

A model reference direct adaptive neural control system for the F-8 aircraft was studied by Suresh et. al. (2006). A neural network with a linear filter was trained offline using backpropagation through a time learning algorithm to approximate the unknown control law. The offline trained system acts as a starting point for the online adaptation for different flight scenarios. The model reference direct adaptive neural control (MRDANC) system was compared to a dynamic inversion controller and a radial basis function controller. The MRDANC was able to improve the system damping and the tracking ability of the aircraft. An uncertainty of 70% was added to the plant model and control surface losses were also modelled by a 70% uncertainty in the control matrix. The online adaptation was observed in the results, as the controller adapted to the new situations. The MRDANC controller had the ability to track a commanded signal under model error and control surface loss. The dynamic inversion and radial basis function network worked well up to an uncertainty of 30% in the model. When stretched to a 40% model error, these controllers had oscillatory output, but settled eventually. After comparing the mean square errors, the proposed controller performs better.

Kaneshige et. al. (2000) devised a generic flight control system using neural techniques. The controller eliminates the use of extensive gain scheduling and explicit identification. The neural network approach brought together direct adaptive control and dynamic inversion. The controller uses pre-trained and on-line learning neural networks and reference models to specify the desired handling qualities. The pre-trained networks give the aerodynamic and control characteristic estimates that are required for the inversion. The on-line learning networks are used to compensate for errors and adapt to changes in the aircraft dynamics. This then results in consistent handling qualities throughout the

flight envelope for various aircraft types. The controller was tested on commercial transport types, high performance military aircraft and hypersonic aircraft. Simulations showed that the generic flight controller was effective for all three cases. The flight control system performance was comparable to that of the actual control systems of each aircraft.

Fuzzy logic has emerged as a technique for controlling a plant through the feedback control loop. A fuzzy controller is less sensitive to variations in the plant than a conventional controller. The use of fuzzy logic gives an almost generic flight control system that may eliminate the use of gain scheduling. A conventional controller refers to a PID controller that uses gain scheduling.

Nikolos et. al. (2003) investigated the roll control of a UAV using fuzzy logic. A simple kinematics model was developed to simulate the NEARCHOS UAV's roll motion. This model was validated against actual flight data. A Mamdani type fuzzy controller was developed and implemented. The control system is capable of handling multiple inputs and but the output is a single state variable. The fuzzy logic system attempted to follow a number of predefined desired trajectories. The controller was able to follow the paths with only a small deviation. The good results in this study are a motivation to investigate a longitudinal behaviour of the aircraft using a similar controller.

An adaptive predictor-corrector control strategy was developed as part of the NASA Intelligent Flight Control System (IFCS) Program for the F-15 aircraft (Battipede et. al., 2003). This strategy was based on a reference model direct inverse scheme comprising, two adaptive neural networks used to identify the forward and inverse F-15 model. Controller performance was obtained at areas where the state variables vary within a wide portion of the flight envelope. The controller performance does not diminish when the aircraft changes flight condition – this was due to the compensation of the neural networks. The computation was found to be intense and a large computer processor was required to get results.

Doitsidis et. al. (2004) applied fuzzy logic to the autonomous navigation of small UAVs. A two-module fuzzy Mamdani controller was developed and was made up of the altitude module and the latitude-longitude module with an error calculating block for controller parameter tuning and flight adjustment purposes. The controller design was modular and can be applied to any air vehicle. Simulations showed that the controller performed

adequately well. But it was discovered that once the UAV reached its desired altitude, oscillations in the z-axis appeared. This was attributed to the controller design being based on a human pilot model and not on flight performance observations. Tuning will provide better results.

Won et. al. (1999) have used fuzzy logic to design a controller for a supermanoeuvrable version of the F-18 aircraft in a high angle of attack manoeuvre. The design incorporated PID control with sliding control under the fuzzy logic framework incorporating gain scheduling with multilayered fuzzy rules. They included the nonlinear aerodynamic characteristics of the airplane as well as thrust vector controls in their analysis. The angle of attack response shows that the fuzzy control results are comparable to those of the design goals and was effective for going between low, intermediate and high dynamic pressure regions. It was also shown that at low dynamic pressure, thrust vectoring was more effective than elevator control. It was also found that fuzzy implementation of such a system took a much shorter time than if a more conventional approach were to be used. It was concluded that a fuzzy controller would be suitable for control of a supermanoeuvrable aircraft at high angle of attack. The adaptability of the fuzzy controller to sudden changes in operating conditions was demonstrated.

Thampi et. al. (2002) proposed using multiple adaptive controllers in aircraft control to avoid issues of gain scheduling throughout the flight envelope. This study uses multiple models to identify the system over the whole flight envelope, where switching between models occurs according to the change in aircraft dynamics. The research showed that multiple model based controllers do well in tracking the desired signal, since the aircraft dynamics is known at each point.

The ducted fan UAV is an emerging technology under the American Defence Advanced Research Projects Agency program. An unaugmented ducted fan has nonlinear dynamic characteristics, and is unstable and susceptible to disturbances. A nonlinear dynamic inversion controller has been proposed to overcome the floor flying qualities of the UAV (Spaulding et. al., 2005). The controller approach used was found to be adequate in that the complex vehicle dynamics were reduced to that of a simple integrator, which was able to cover the entire flight envelope without the need for gain scheduling.

1.2.3 Current Study

Literature consulted (Dogan et. al., 2005; Ochi and Kominani, 2005; Pollini et. al., 2003; Fravolini et. al., 2003) have all considered the aircraft centre of gravity as being a fixed parameter. Much effort has been channelled to specifically research issues related to finding the relative distance between a tanker and a UAV (Vendra et. al., 2007; Dell'Aquila et. al., 2007; Mammarella et. al., 2008). Insufficient attention has been given to simulation and fundamental modelling of the receiver vehicle during refuelling. The current study looks into the control of a UCAV for the case of in-flight-refuelling specifically taking the cg movement into account.

Studies conducted by Won et. al. (1999) highlighted shortcomings of the fuzzy logic controller when applied to a supermanoeuvrable aircraft. These were overcome using a PID control algorithm, but then required gain scheduling. A conventional controller used in aircraft applications assumes that the dynamics are linear and time invariant about a nominal flight condition. Controller gains are scheduled to be functions of these flight conditions. But, due to the effects of unmodelled dynamics in the system, the performance of such control systems become non-optimal. The use of dynamic inversion eliminates the need for a gain scheduled controller. A dynamic inversion control law allows state variables to be commanded directly. Dynamic inversion control as applied by Spaulding et. al. (2005), and Steinberg (2001) has been successful. But, dynamic inversion is vulnerable to modelling and inversion errors. Using a neural network based control design would assist in overcoming these errors in addition to addressing unmodelled dynamics and parametric uncertainty.

Kaneshige et. al. (2000), Li et. al. (2001) and Battipede et. al. (2003) make use of online neural networks in order for the controller to be able to adapt to changes in the aircraft dynamics and to be applicable throughout the entire flight envelope. This study focuses on a specific case in the aircraft flight envelope, namely, the automated aerial refuelling case. It is unnecessary at this stage to consider using an online network as the neural network in this study is trained offline to simply invert the plant dynamics. When considering a very wide operating envelope, the online network has been shown to be effective, but as highlighted in the study of Battipede et. al. (2003), intense computation is required for this option.

A GRBFN was proposed by Li et. al. (2001). This was done in order to capture the aircraft dynamics over a wide operating range. Again, since this study looks at a specific case in a possibly large operating envelope, a simple RBF network will be sufficient to use in this study.

1.3 Thesis Objectives

The main objective of the study is to investigate the suitability of using a nonlinear dynamic inversion controller with a neural network for a UCAV during aerial refuelling.

The following sub-objectives for the research have been defined:

1. To combine a nonlinear 6 DOF flight model for a converted UCAV with a cg travel model for the automated aerial refuelling flight condition
2. To numerically simulate the behaviour of the aircraft based on the developed model
3. To validate the model using existing data
4. To combine a radial basis function neural network for plant inversion with a nonlinear dynamic inversion controller

1.4 Thesis Outline

The aircraft mathematical model is discussed in Chapter 2. General equations of motion for an aircraft are presented. Thereafter assumptions used in this study are discussed and the equations are written for the specific case of flight refuelling. In Chapter 3, the mathematical considerations behind the control system used in this research are explored. The nonlinear dynamic inversion control formulation is given. The radial basis function neural network is explained. Chapter 4 discusses how the models from Chapters 2 and 3 were synthesised and implemented in the Matlab/Simulink environment. In Chapter 5, the model used is validated against existing data. The results are presented and discussed in Chapter 6. This chapter is further organised into a discussion of the inherent aircraft behaviour during refuelling, and a discussion of the controlled aircraft behaviour. Finally, Chapter 7 concludes the study and further work is recommended.

2 Aircraft Mathematical Model

2.1 Aircraft Six-Degree-of-Freedom Nonlinear Equations of Motion

The aircraft mathematical model with the inclusion of the effects of change in cg position is adopted from Pedro (1992). It is consistent with the equations in Etkin and Reid (1994) and Stevens and Lewis (1992) when these effects are neglected. The twelve aircraft equations of motion are broken down into four sets, as given below (Pedro, 1992):

Force Equations

$$m(\dot{u}+qw-rv)-S_x(q^2+r^2)-S_y(\dot{r}-pq)+S_z(\dot{q}+pr) = -mg\sin\theta+T\cos\varphi_{TZ}\cos\varphi_{TY}-\frac{1}{2}\rho V^2 SC_x \quad \text{... (2.1)}$$

$$m(\dot{v}+ru-pw)+S_x(\dot{r}+qp)-S_y(p^2+r^2)-S_z(\dot{p}-qr) = mg\cos\theta\sin\phi+T\sin\varphi_{TZ}+\frac{1}{2}\rho V^2 SC_y \quad \text{... (2.2)}$$

$$m(\dot{w}+pv-qu)-S_x(\dot{q}-pr)+S_y(\dot{p}+qr)-S_z(q^2+p^2) = mg\cos\theta\cos\phi-T\cos\varphi_{TZ}\sin\varphi_{TY}+\frac{1}{2}\rho V^2 SC_z \quad \text{... (2.3)}$$

Moment Equations

$$I_x\dot{p}-(I_y-I_z)qr-I_{xy}(q-pr)-I_{xz}(\dot{r}+pq)-I_{yz}(q^2-r^2)+S_y(\dot{w}-uq+pv) - S_z(\dot{v}-pw+ur) = mg(y_c\cos\theta\cos\phi-z_c\cos\theta\sin\phi) - T(y_T\cos\varphi_{TZ}\sin\varphi_{TY}+z_T\sin\varphi_{TZ})+I_T\omega_T(r\sin\varphi_{TZ}+q\cos\varphi_{TY}\sin\varphi_{TY}) + \frac{1}{2}\rho V^2 SbC_l \quad \text{... (2.4)}$$

$$I_y\dot{q}-(I_z-I_x)rp-I_{xy}(\dot{p}+qr)-I_{yz}(\dot{r}-qp)-I_{xz}(r^2-p^2) - S_x(\dot{w}+vp-uq)+S_z(\dot{u}-vr+qw) = -mg(z_c\sin\theta+x_c\cos\theta\cos\phi) + T(z_T\cos\varphi_{TZ}\cos\varphi_{TY}+x_T\cos\varphi_{TZ}\sin\varphi_{TY}) - I_T\omega_T(r\cos\varphi_{TZ}\cos\varphi_{TY}+p\cos\varphi_{TY})+\frac{1}{2}\rho V^2 S\bar{c}C_m \quad \text{... (2.5)}$$

$$
\begin{aligned}
&I_z\dot{r} - \left(I_x - I_y\right)pq - I_{zy}\left(\dot{q} + rp\right) - I_{zx}\left(\dot{p} - rq\right) - I_{xy}\left(p^2 - q^2\right) + \\
&S_x\left(\dot{v} - wp + ru\right) - S_y\left(\dot{u} + rv + wq\right) = mg\left(x_c\cos\theta\sin\phi + y_c\sin\theta\right) \\
&+ T\left(x_T\sin\varphi_{TZ} - y_T\cos\varphi_{TZ}\cos\varphi_{TY}\right) + I_T\omega_T\left(q\cos\varphi_{TZ}\cos\varphi_{TY} - p\sin\varphi_{Tz}\right) \quad \text{... (2.6)} \\
&+ \frac{1}{2}\rho V^2 SbC_n
\end{aligned}
$$

Kinematic Equations

$$\dot{\phi} = p + \left(q\sin\phi\tan\theta + r\cos\phi\tan\theta\right) \quad \text{... (2.7)}$$

$$\dot{\theta} = q\cos\phi - r\sin\phi \quad \text{... (2.8)}$$

$$\dot{\psi} = \left(q\sin\phi + r\cos\phi\right)\sec\theta \quad \text{... (2.9)}$$

Navigation Equations

$$
\begin{aligned}
\dot{x}_E &= u\cos\theta\cos\psi \\
&+ v\left(\sin\phi\sin\theta\cos\psi - \cos\phi\sin\phi\right) \quad \text{... (2.10)} \\
&+ w\left(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi\right)
\end{aligned}
$$

$$
\begin{aligned}
\dot{y}_E &= u\cos\theta\sin\psi \\
&+ v\left(\sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi\right) \quad \text{... (2.11)} \\
&+ w\left(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi\right)
\end{aligned}
$$

$$\dot{z}_E = -u\sin\theta + v\left(\sin\phi\cos\theta\right) + w\left(\cos\phi\cos\theta\right) \quad \text{... (2.12)}$$

where:

$$I_x = I_{x_0} + m\left(y_c^2 + z_c^2\right) \quad \text{... (2.13)}$$

$$I_y = I_{y_0} + m\left(x_c^2 + z_c^2\right) \quad \text{... (2.14)}$$

$$I_z = I_{z_0} + m\left(x_c^2 + y_c^2\right) \quad \text{... (2.15)}$$

$$I_{xy} = I_{xy_0} + mx_c y_c \quad \text{... (2.16)}$$

$$I_{xz} = I_{xz_0} + mx_c z_c \quad \text{... (2.17)}$$

$$I_{yz} = I_{yz_0} + my_c z_c \quad \text{... (2.18)}$$

$$S_x = mx_c \quad \text{... (2.19)}$$

$$S_y = my_c \quad \text{... (2.20)}$$

$$S_z = mz_c \quad \text{... (2.21)}$$

$$x_c = x_{cg_{ref}} - x_{cg} \quad \text{... (2.22)}$$

$$y_c = y_{cg_{ref}} - y_{cg} \quad \text{... (2.23)}$$

$$z_c = z_{cg_{ref}} - z_{cg} \quad \text{... (2.24)}$$

$$\boldsymbol{r_c} = [x_c, y_c, z_c]^T \quad \text{... (2.25)}$$

$I_{x_0}, I_{y_0}, I_{z_0}, I_{xy_0}, I_{xz_0}$ and I_{yz_0} are the aircraft moments and products of inertia before refuelling occurs.

The cg position vector is $\mathbf{r_c}$ and is illustrated in Figure 2.1.

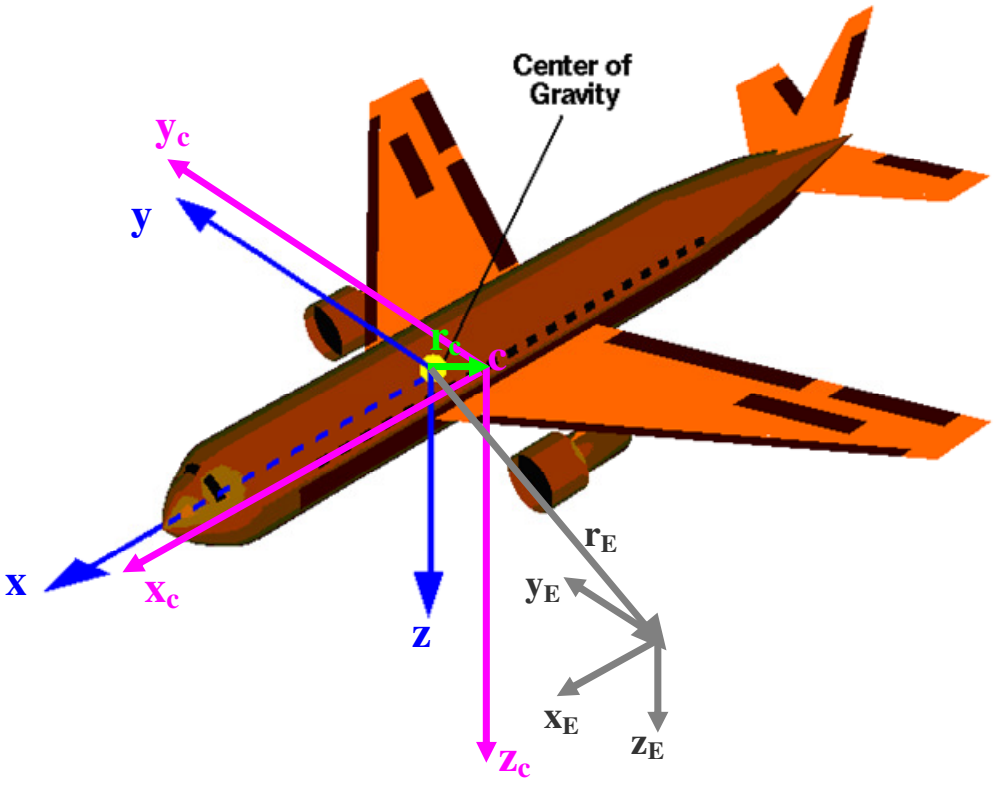

Figure 2.1: Aircraft axes with displacement vector of cg

The thrust component axes transformations from engine axes to body axes are indicated in Figure 2.2.

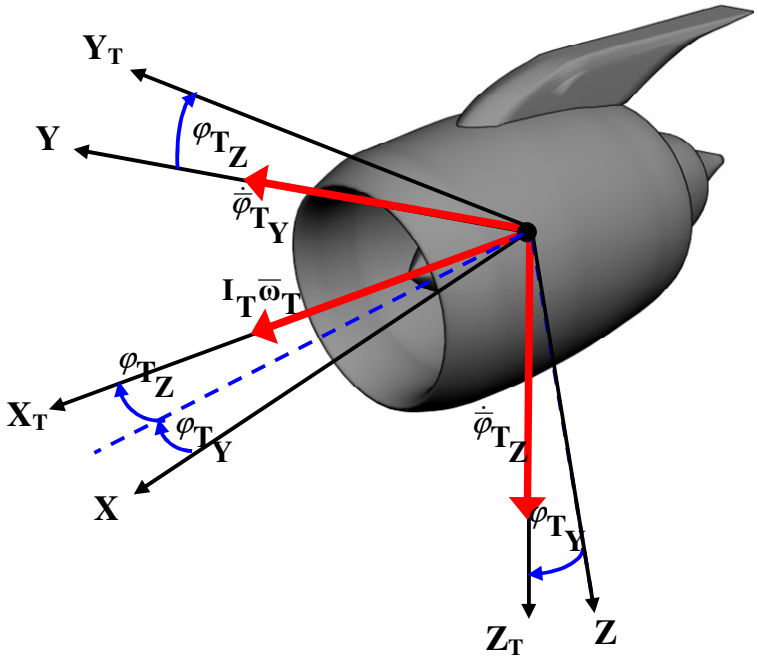

Figure 2.2: Axes and transformation angles for thrust components

The aerodynamic components of the force and moment equations are given in parametric form (Morelli, 1998):

$$C_x = C_x(\alpha, \delta_e) + C_{xq}(\alpha)\tilde{q} \quad \text{... (2.26)}$$

$$C_y = C_y(\beta, \delta_a, \delta_r) + C_{y_p}(\alpha)\tilde{p} + C_{y_r}(\alpha)\tilde{r} \quad \text{... (2.27)}$$

$$C_z = C_z(\alpha, \beta, \delta_e) + C_{zq}(\alpha)\tilde{q} \quad \text{... (2.28)}$$

$$C_l = C_l(\alpha, \beta) + C_{l_p}(\alpha)\tilde{p} + C_{l_r}(\alpha)\tilde{r} + C_{l\delta a}(\alpha, \beta)\delta_a + C_{l\delta r}(\alpha, \beta)\delta_r \quad \text{... (2.29)}$$

$$C_m = C_m(\alpha, \delta_e) + C_{mq}(\alpha)\tilde{q} + C_z\left(xcg_{ref} - xcg\right) \quad \text{... (2.30)}$$

$$C_n = C_n(\alpha, \beta) + C_{np}(\alpha)\tilde{p} + C_{nr}(\alpha)\tilde{r} + C_{n\delta a}(\alpha, \beta)\delta_a + C_{n\delta r}(\alpha, \beta)\delta_r - C_y\left(xcg_{ref} - xcg\right)\left(\frac{\bar{c}}{b}\right) \quad \text{... (2.31)}$$

where

$$\tilde{p} = \frac{pb}{2V}; \qquad \tilde{q} = \frac{q\bar{c}}{2V}; \qquad \tilde{r} = \frac{rb}{2V} \quad \text{... (2.32)}$$

Each function is defined below.

$$C_x(\alpha, \delta_e) = a_0 + a_1\alpha + a_2\delta_e{}^2 + a_3\delta_e + a_4\alpha\delta_e + a_5\alpha^2 + a_6\alpha^3 \quad \text{... (2.33)}$$

$$C_{xq}(\alpha) = b_0 + b_1\alpha + b_2\alpha^2 + b_3\alpha^3 + b_4\alpha^4 \quad \text{... (2.34)}$$

$$C_y(\beta, \delta_a, \delta_r) = c_0 + c_1\delta_a + c_2\delta_r \quad \text{... (2.35)}$$

$$C_{yp}(\alpha) = d_0 + d_1\alpha + d_2\alpha^2 + d_3\alpha^3 \quad \text{... (2.36)}$$

$$C_{yr}(\alpha) = e_0 + e_1\alpha + e_2\alpha^2 + e_3\alpha^3 \quad \text{... (2.37)}$$

$$C_z(\alpha, \beta, \delta_e) = \left(f_0 + f_1\alpha + f_2\alpha^2 + f_3\alpha^3 + f_4\alpha^4\right)\left(1 - \beta^2\right) + f_5\delta_e \quad \text{... (2.38)}$$

$$C_{zq}(\alpha) = g_0 + g_1\alpha + g_2\alpha^2 + g_3\alpha^3 + g_4\alpha^4 \quad \text{... (2.39)}$$

$$C_l(\alpha, \beta) = h_0\beta + h_1\alpha\beta + h_2\alpha^2\beta + h_3\beta^2 + h_4\alpha\beta^2 + h_5\alpha^3\beta + h_6\alpha^4\beta + h_7\alpha^2\beta^2 \ldots \quad \text{(2.40)}$$

$$C_{lp}(\alpha) = i_0 + i_1\alpha + i_2\alpha^2 + i_3\alpha^3 \quad \text{... (2.41)}$$

$$C_{lr}(\alpha) = j0 + j_1\alpha + j_2\alpha^2 + j_3\alpha^3 + j_4\alpha^4 \quad \text{... (2.42)}$$

$$C_{l\delta a}(\alpha, \beta) = k_0 + k_1\alpha + k_2\beta + k_3\alpha^2 + k_4\alpha\beta + k_5\alpha^2\beta + k_6\alpha^3 \quad \text{... (2.43)}$$

$$C_{l\delta r}(\alpha, \beta) = l_0 + l_1\alpha + l_2\beta + l_3\alpha\beta + l_4\alpha^2\beta + l_5\alpha^3\beta + l_6\beta^2 \quad \text{... (2.44)}$$

$$C_m(\alpha, \delta_e) = m_0 + m_1\alpha + m_2\delta_e + m_3\alpha\delta_e + m_4{\delta_e}^2 + m_5\alpha^2\delta_e + m_6{\delta_e}^3 + m_7\alpha{\delta_e}^2 \ldots \quad \text{(2.45)}$$

$$C_{mq}(\alpha) = n_0 + n_1\alpha + n_2\alpha^2 + n_3\alpha^3 + n_4\alpha^4 + n_5\alpha^5 \quad \text{... (2.46)}$$

$$C_n(\alpha, \beta) = o_0\beta + o_1\alpha\beta + o_2\beta^2 + o_3\alpha\beta^2 + o_4\alpha^2\beta + o_5\alpha^2\beta^2 + o_6\alpha^3\beta \quad \text{... (2.47)}$$

$$C_{np}(\alpha) = p_0 + p_1\alpha + p_2\alpha^2 + p_3\alpha^3 + p_4\alpha^4 \quad \text{... (2.48)}$$

$$C_{nr}(\alpha) = q_0 + q_1\alpha + q_2\alpha^2 \quad \text{... (2.49)}$$

$$C_{n\delta a}(\alpha, \beta) = r_0 + r_1\alpha + r_2\beta + r_3\alpha\beta + r_4\alpha^2\beta + r_5\alpha^3\beta + r_6\alpha^2 + r_7\alpha^3 + r_8\beta^3 + r_9\alpha\beta^3 \ldots \quad \text{(2.50)}$$

$$C_{n\delta r}(\alpha, \beta) = s_0 + s_1\alpha + s_2\beta + s_3\alpha\beta + s_4\alpha^2\beta + s_5\alpha^2 \quad \text{... (2.51)}$$

The values of constants a_0 to s_5 are given in Morelli (1998).

The angle of attack, α, the sideslip angle, β, and the total speed, V are calculated from:

$$\alpha = \tan^{-1}\left(\frac{w}{u}\right) \quad \text{... (2-52)}$$

$$\beta = \sin^{-1}\left(\frac{v}{V}\right) \quad \text{... (2-53)}$$

$$V = \sqrt{u^2 + v^2 + w^2} \quad \text{... (2-54)}$$

The thrust is modelled as a function of air density, total speed and throttle position and coefficients (A, B, C) specific to the engine (Pedro and Bigg, 2005)

$$T = \left(\frac{\rho}{\rho_0}\right)^{0.7} \left(AV^2 + BV + C\right)\frac{\delta_T}{\delta_{T_{MAX}}} \quad \text{... (2.55)}$$

where: A = -2.338; B = 574; and C = 65260. The constants A B and C are typical to a turbofan engine.

Examining the equations of motion as has been described, it is observed that there is coupling between the lateral-directional and longitudinal motions via S_x, S_y, and S_z. The force in the X-direction is seen to be dependent on the yaw acceleration and square of yaw rate via the centre of gravity position. The Y-direction force is a function of the roll acceleration due to the centre of gravity being misaligned with the reference cg. Similarly with the Z-direction force, coupling exists between the longitudinal and lateral-directional variables.

The moment equations also show very interesting cross coupling of terms related via the centre of gravity position. The yaw and roll moments have dependence on rate of change of vertical and forward speed respectively. All three moments have contributions from the mass of the aircraft due to the cg having changed position from the reference.

2.2 Flight Refuelling Model

For the UCAV configuration of the F-16, there are two 600 gallon tanks on each wing (Eshel, 2004) as depicted in Figure 2.3.

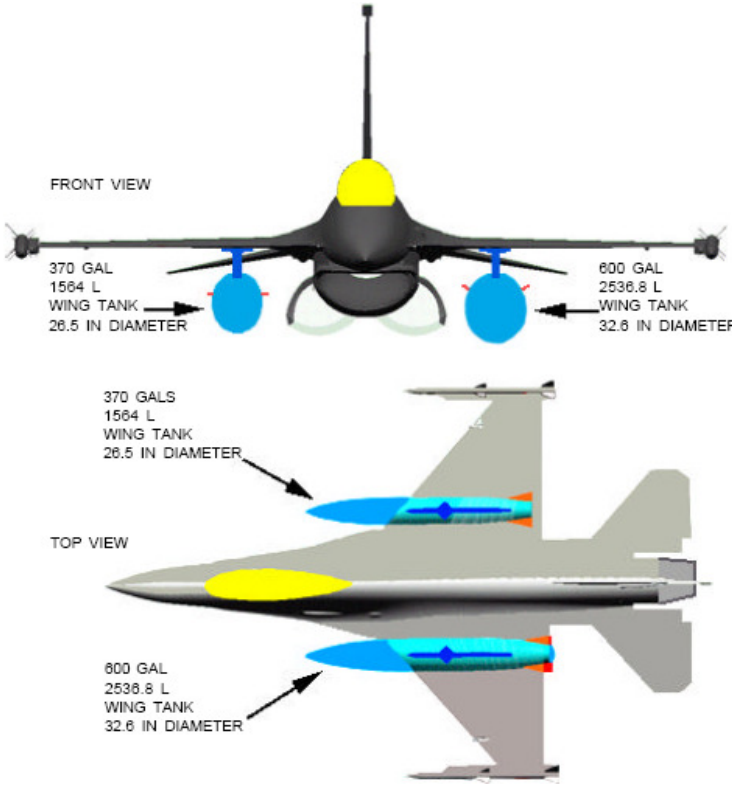

Figure 2.3: F-16 External fuel tank arrangement

This gives a total mass of 4290 kg of fuel. The fuelling rate for the F-16 is 10 000 lb of fuel in 7 minutes, that is, 10.886 kg/s (Koch, 2005). Based on this, the total time taken to refuel the F-16 external tanks is 395 s. Note, before refuelling, the F-16 mass is 9299 kg (Stevens and Lewis, 1992).

Figure 2.4 shows the cg of the aircraft and the cg of the fuel added into the external tanks.

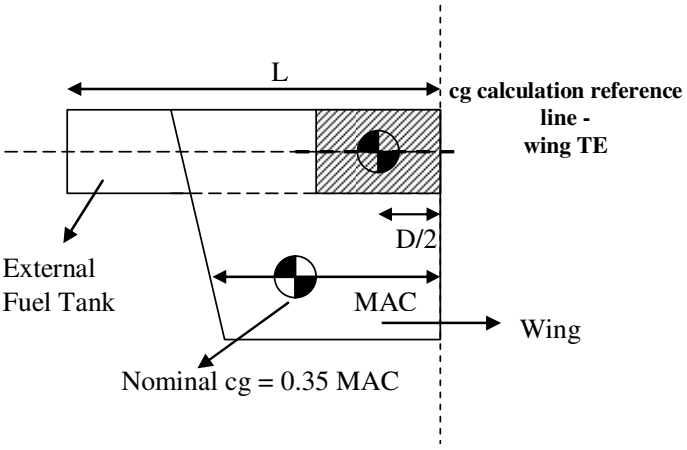

Figure 2.4: Layout of wing/external fuel tank

The new overall cg position is therefore given by the following relation:

$$\frac{m_{fuel} \times \frac{D}{2} + m_{aircraft} \times (1 - 0.35)\bar{c}}{m_{fuel} + m_{aircraft}} \quad \text{... (2.56)}$$

where 0.35 refers to the nominal cg position (see Figure 2.4).

2.3 Simplified Mathematical Model used for Aerial Refuelling Simulation

2.3.1 Assumptions

- The rate of fuel use during the refuelling period was negligible compared to the rate of fuel inflow.
- The mass moments of inertia of the aircraft remained constant throughout the refuelling period.
- Only the variation of the cg along the x-axis is considered

$$y_c = 0 \Rightarrow S_y = 0 \quad \text{... (2.57)}$$

$$z_c = 0 \Rightarrow S_z = 0 \quad \text{... (2.58)}$$

- The line of action of thrust force is assumed to be aligned in the plane of the aircraft cg, and thus gives no contribution to the force in the y-direction and z-direction; neither does it contribute to any of the aerodynamic moments, i.e.:

$$\varphi_{TZ} = \varphi_{TY} = 0 \quad \text{... (2.59)}$$

$$z_T = y_T = 0 \quad \text{... (2.60)}$$

- The gyroscopic effects of engine spinning rotors are negligible, i.e.:

$$I_T \omega_T \approx 0 \quad \text{... (2.61)}$$

- Using body axes, the plane Cxz is a plane of symmetry i.e.:

$$I_{yz} = I_{xy} = 0 \quad \text{... (2.62)}$$

2.3.2 Final Equations

After applying the assumptions, the final equations can be written:

Force Equations

$$m(\dot{u} + qw - rv) - S_x(q^2 + r^2) = -mg\sin\theta + T - \frac{1}{2}\rho V^2 SC_x \quad \text{... (2.63)}$$

$$m(\dot{v} + ru - pw) + S_x(\dot{r} + qp) = mg\cos\theta\sin\phi + \frac{1}{2}\rho V^2 SC_y \quad \text{.. (2.64)}$$

$$m(\dot{w} + pv - qu) - S_x(\dot{q} - pr) = mg\cos\theta\cos\phi + \frac{1}{2}\rho V^2 SC_z \quad \text{... (2.65)}$$

Moment Equations

$$I_x\dot{p} - (I_y - I_z)qr - I_{xz}(\dot{r} + pq) = \frac{1}{2}\rho V^2 SbC_l \quad \text{... (2.66)}$$

$$\begin{aligned} &I_y\dot{q} - (I_z - I_x)rp - I_{xz}(r^2 - p^2) - S_x(\dot{w} + vp - uq) \\ &= -mg(x_c\cos\theta\cos\phi) + \frac{1}{2}\rho V^2 S\bar{c}C_m \end{aligned} \quad \text{... (2.67)}$$

$$\begin{aligned} &I_z\dot{r} - (I_x - I_y)pq - I_{zx}(\dot{p} - rq) + S_x(\dot{v} - wp + ru) \\ &= mg(x_c\cos\theta\sin\phi) + \frac{1}{2}\rho V^2 SbC_n \end{aligned} \quad \text{... (2.68)}$$

Kinematic Equations

$$\dot{\phi} = p + (q\sin\phi\tan\theta + r\cos\phi\tan\theta) \quad \text{... (2.69)}$$

$$\dot{\theta} = q\cos\phi - r\sin\phi \quad \text{... (2.70)}$$

$$\dot{\psi} = (q\sin\phi + r\cos\phi)\sec\theta \quad \text{... (2.71)}$$

Navigation Equations

$$\begin{aligned} \dot{x}_E &= u\cos\theta\cos\psi \\ &+ v(\sin\phi\sin\theta\cos\psi - \cos\phi\sin\phi) \\ &+ w(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi) \end{aligned} \quad \text{... (2.72)}$$

$$\begin{aligned} \dot{y}_E &= u\cos\theta\sin\psi \\ &+ v(\sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi) \\ &+ w(\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi) \end{aligned} \quad \text{... (2.73)}$$

$$\dot{z}_E = -u\sin\theta + v(\sin\phi\cos\theta) + w(\cos\phi\cos\theta) \quad \text{... (2.74)}$$

2.4 Trimmed Solution to Equations of Motion and Modal Analysis

2.4.1 Trimmed Solution

"An aircraft is in trimmed, equilibrium flight when its velocity is fixed and its pitch and roll angles are unchanging" (Stengel, 2004).

The trim problem is to find the control settings that yield a steady flight condition. The object is to set the vector Eq. (2.75) to zero

$$\dot{\mathbf{x}}_{\mathbf{d}} = f_d(\mathbf{x}_d, \mathbf{u}, \mathbf{w}) = 0 \qquad \text{... (2.75)}$$

by the proper choice of control **u** subject to the constant values of $\mathbf{x_d}$ and disturbance **w**, where

$$\mathbf{x}_{\mathbf{d}} = [u, v, w, p, q, r]^T \qquad \text{... (2.76)}$$

The remaining components,

$$\mathbf{x}_k = [x, y, z, \phi, \theta, \psi]^T \qquad \text{... (2.77)}$$

may be treated as fixed or free parameters

The desired trim condition can be specified by various combinations of velocity and angle components. With conventional δ_e, δ_a, δ_r, δ_T, a typical specification would be V, γ, β and φ. θ and ϕ would be free to take the necessary values. The trim variables and attitude angles specify the corresponding u, v, w, p, q, and r. Although they are free variables, θ and μ do not change unless forced to do so i.e., an adjustment rule is required. That rule is provided by incorporating the free variables in the control vector; hence **u** is (Stengel, 2004):

$$\mathbf{u} \triangleq [\delta_e, \delta_T, \theta, \delta_a, \delta_r, \mu] \qquad \text{... (2.78)}$$

Trim equations were set up to find the trimmed values of u, v, w, α, δ_e, β, δ_a, δ_r, and δ_T, during the refuelling period. In general, trim, the following conditions must be satisfied:

$$\begin{aligned} X_{total} &= 0 \qquad Y_{total} = 0 \qquad Z_{total} = 0 \\ L_{total} &= 0 \qquad M_{total} = 0 \qquad N_{total} = 0 \end{aligned} \qquad \text{... (2.79)}$$

$$V - \sqrt{u^2 + v^2 + w^2} = 0$$

$$\tan\alpha - \frac{w}{u} = 0 \quad \text{... (2.80)}$$

$$\sin\beta - \frac{v}{V} = 0$$

where:

$$X_{total} = X_{aero} + T + mg\sin\alpha \quad \text{... (2.81)}$$

$$Y_{total} = Y_{aero} - mg\cos\theta\sin\phi \quad \text{... (2.82)}$$

$$Z_{total} = Z_{aero} + mg\cos\theta\cos\phi \quad \text{... (2.83)}$$

$$L_{total} = L_{aero} \quad \text{... (2.84)}$$

$$M_{total} = M_{aero} - mgx_c\cos\theta\cos\phi \quad \text{... (2.85)}$$

$$N_{total} = N_{aero} + mgx_c\cos\theta\sin\phi \quad \text{... (2.86)}$$

The aerodynamic forces, X_{aero}, Y_{aero} and Z_{aero} and moments L_{aero}, M_{aero}, and N_{aero} are given by

$$X_{aero} = 0.5\rho V^2 SC_x \quad \text{... (2.87)}$$

$$Y_{aero} = 0.5\rho V^2 SC_y \quad \text{... (2.88)}$$

$$Z_{aero} = 0.5\rho V^2 SC_z \quad \text{... (2.89)}$$

$$L_{aero} = 0.5\rho V^2 SbC_l \quad \text{... (2.90)}$$

$$M_{aero} = 0.5\rho V^2 ScC_m \quad \text{... (2.91)}$$

$$N_{aero} = 0.5\rho V^2 SbC_n \quad \text{... (2.92)}$$

2.4.2 Modal Analysis

Modal analysis arises from observing the behaviour of the uncontrolled aircraft, meaning, the motion of the aircraft with the controls locked in position. While an aircraft is in steady flight, it can be subjected to a momentary disturbance of various kinds. Here, the behaviour of the motion is analysed after the disturbance. The required dynamic behaviour of an aircraft is an important issue to consider, as disturbances in a steady flight condition must be small for a vehicle to be acceptable for use. The small

disturbance properties (i.e., the natural modes) must be kept to an acceptably small level, by automatic control, in the case of the UCAV (Etkin and Reid, 1996).

The small disturbance equations are of the form:

$$\Delta\dot{\mathbf{x}} = \mathbf{A}\Delta\mathbf{x} + \Delta\mathbf{f}_c \quad \textbf{... (2.93)}$$

As mentioned earlier, the controls are locked and hence this vector is zero, leaving:

$$\Delta\dot{\mathbf{x}} = \mathbf{A}\Delta\mathbf{x} \quad \textbf{... (2.94)}$$

This is a first-order differential equation to which the solution has the form:

$$\mathbf{x}(t) = \mathbf{x}_0 e^{\lambda t} \quad \textbf{... (2.95)}$$

$\mathbf{x}_0$ is the eigenvector and λ is the eigenvalue of the system.

Since any of the eigenvalues of the system can provide a solution to the equation $\Delta\dot{\mathbf{x}} = \mathbf{A}\Delta\mathbf{x}$ and since the system is linear, the general solution is:

$$\mathbf{x}(t) = \sum_i \mathbf{x}_{0i} \exp(\lambda_i t) \quad \textbf{... (2.96)}$$

Each solution of $\mathbf{x}(t) = \mathbf{x}_0 e^{\lambda t}$ is called a natural mode. In general,

$$\lambda = n \pm i\omega_n \quad \textbf{... (2.97)}$$

Depending on whether λ has an imaginary part and on the sign of n, four modes can be described.

Certain parameters are useful in giving the dynamic properties of the aircraft. These are the damped frequency, the damping ratio, settling time, period and the time constant. The equations used to evaluate these parameters are given below:

Damped frequency (rad.s^{-1}):

$$\omega_d = \sqrt{(\omega_n^2 + n^2)} \quad \textbf{... (2.98)}$$

Damping ratio:

$$\zeta = -\frac{n}{\omega_d} \quad \textbf{... (2.99)}$$

Settling time (s):

$$t_s = \frac{4}{\omega_d \times \zeta} \quad \textbf{... (2.100)}$$

Period(s):

$$T = \frac{2\pi}{\omega_n} \quad \textbf{... (2.101)}$$

Time constant (s):

$$\tau = \frac{1}{\omega_d \times \zeta} \quad \textbf{... (2.102)}$$

3 Controller Implementation

3.1 Nonlinear Dynamic Inversion

In general, aircraft dynamics can be expressed as follows:

$$\dot{\mathbf{x}} = F(\mathbf{x},\mathbf{u}) \quad \text{... (3.1)}$$

$$\mathbf{y} = H(\mathbf{x}) \quad \text{... (3.2)}$$

where **x** is the state vector, **u** is the control vector, and **y** is the output vector. Rewriting Eq. (3.1)

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u} \quad \text{... (3.3)}$$

where **f** and **g** are nonlinear state and control distribution functions respectively. The control law can be found by

$$\mathbf{u} = \mathbf{g}^{-1}(\mathbf{x})[\dot{\mathbf{x}} - \mathbf{f}(\mathbf{x})] \quad \text{... (3.4)}$$

under the assumption that **g**(***x***) is invertible for all values of **x**.

The aircraft is to be commanded to specified states, ***x***. But the rates of these states will be specified, $\dot{\mathbf{x}}_{des}$. Substituting this into Eq. (4.4) gives:

$$\mathbf{u} = \mathbf{g}^{-1}(\mathbf{x})[\dot{\mathbf{x}}_{des} - \mathbf{f}(\mathbf{x})] \quad \text{... (3.5)}$$

It is important to note that perfect inversion is not always possible. The assumption that **g**(**x**) may be invertible for all x may not necessarily be true. When there are more states than controls, inversion will not be possible. On the other hand, if **g**(**x**) is invertible and is small, the control inputs become large, thus leading to actuator saturation.

Taking the nonlinear characteristics of aircraft motion into account, the equations are:

$$\dot{p} = \frac{I_z L + I_{xz} N}{I_x I_z - I_{xz}^2} + \frac{I_{xz}(I_x - I_y + I_z)pq}{I_x I_z - I_{xz}^2} + \frac{[I_z(I_y - I_z) - I_{xz}^2]qr}{I_x I_z - I_{xz}^2} \quad \text{... (3.6)}$$

$$\dot{q} = \frac{1}{I_y}[M + (I_z - I_x)pr + I_{xz}(r^2 - p^2)] \quad \text{... (3.7)}$$

$$\dot{r} = \frac{I_{xz} L + I_x N}{I_x I_z - I_{xz}^2} + \frac{I_{xz}(I_x - I_y + I_z)pq}{I_x I_z - I_{xz}^2} + \frac{[I_x(I_x - I_y) - I_{xz}^2]pq}{I_x I_z - I_{xz}^2} \quad \text{... (3.8)}$$

For the case of dynamic inversion, the moments L, M and N are assumed to be linear with respect to aerodynamic derivatives (Ito et. al., 2002).

$$L = L_\beta \beta + L_{\delta_a} \delta_a + L_{\delta_r} \delta_r + L_p p + L_r r \quad \text{... (3.9)}$$

$$M = M_\alpha \alpha + M_q q + M_{\delta_e} \delta_e \quad \text{... (3.10)}$$

$$N = N_\beta \beta + N_{\delta_a} \delta_a + N_{\delta_r} \delta_r + N_p p + N_r r \quad \text{... (3.11)}$$

Substituting Eq. (3.9) – (3.11) in Eq (3.6) – (3.8):

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0 & L_\beta & L_p & 0 & L_r \\ M_\alpha & 0 & 0 & M_q & 0 \\ 0 & N_\beta & N_p & 0 & N_r \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ p \\ q \\ r \end{bmatrix} + \begin{bmatrix} 0 & L_{\delta a} & L_{\delta r} \\ M_{\delta e} & 0 & 0 \\ 0 & N_{\delta a} & N_{\delta r} \end{bmatrix} \begin{bmatrix} \delta e \\ \delta a \\ \delta r \end{bmatrix} \quad \text{.... (3.12)}$$

$$+ \begin{bmatrix} I_x & 0 & -I_{xz} \\ 0 & Iy & 0 \\ -I_{xz} & 0 & I_z \end{bmatrix}^{-1} \begin{bmatrix} I_{xz} pq + (I_y - I_z) qr \\ I_{xz} (r^2 - p^2) + (I_z - I_x) pr \\ -I_{xz} qr + (I_x - I_y) pq \end{bmatrix}$$

Inverting Eq. (3.5) and using the commanded, desired and measured values, one obtains:

$$\begin{bmatrix} \delta_e \\ \delta_a \\ \delta_r \end{bmatrix}^{cmd} = \begin{bmatrix} 0 & L_{\delta_a} & L_{\delta_r} \\ M_{\delta_e} & 0 & 0 \\ 0 & N_{\delta_a} & N_{\delta_r} \end{bmatrix}^{-1} \left\{ \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix}^{des} - \begin{bmatrix} 0 & L_\beta & L_p & 0 & L_r \\ M_\alpha & 0 & 0 & M_q & 0 \\ 0 & N_\beta & N_p & 0 & N_r \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \\ p \\ q \\ r \end{bmatrix}^{meas} \right. \quad \text{...}$$

$$\left. - \begin{bmatrix} I_x & 0 & -I_{xz} \\ 0 & Iy & 0 \\ -I_{xz} & 0 & I_z \end{bmatrix}^{-1} \begin{bmatrix} I_{xz} p^{meas} q^{meas} + (I_y - I_z) q^{meas} r^{meas} \\ I_{xz} (r^{meas^2} - p^{meas^2}) + (I_z - I_x) p^{meas} r^{meas} \\ -I_{xz} q^{meas} r^{meas} + (I_x - I_y) p^{meas} q^{meas} \end{bmatrix} \right\}$$

(3.13)

The complete dynamic inversion scheme is represented in Figure 3.1

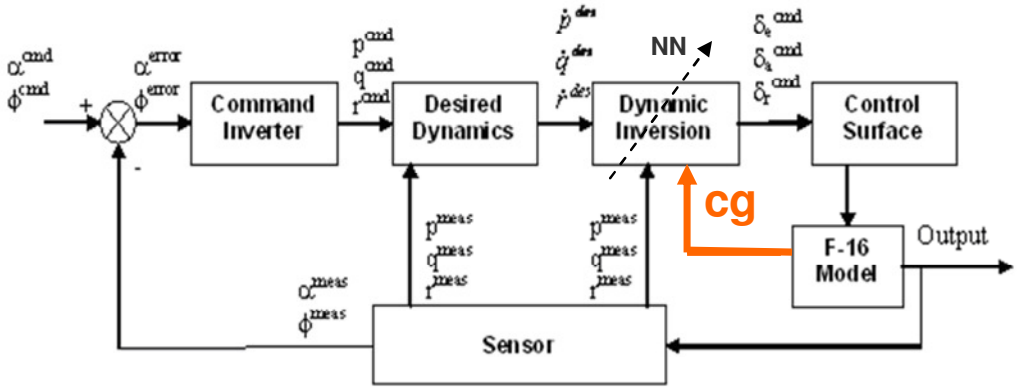

Figure 3.1: Overall dynamic inversion control block diagram (Ito et. al., 2002)

3.1.1 Command Inverter

Rate commands are needed as inputs to the desired dynamics block. The command inverter block changes displacement commands into rate commands so that displacement commands can be directly implemented.

The rates of roll, pitch and yaw are given by Etkin (1996):

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \cos\theta\sin\phi \\ 0 & -\sin\phi & \cos\theta\cos\phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad \text{... (3.14)}$$

Substituting commanded values in Eq (3.14) with $\dot{\psi}^{cmd} = 0$ gives:

$$\begin{bmatrix} p^{cmd} \\ q^{cmd} \\ r^{cmd} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \cos\phi \\ 0 & -\sin\phi \end{bmatrix} \begin{bmatrix} \dot{\phi}^{cmd} \\ \dot{\theta}^{cmd} \end{bmatrix} \quad \text{... (3.15)}$$

Thus:

$$p^{cmd} = \frac{d}{dt}\phi^{cmd} \text{ and } q^{cmd} = \cos\phi\left(\frac{d}{dt}\theta^{cmd}\right) \quad \text{... (3.16)}$$

The function linking θ^{cmd} to angle of attack, α, sideslip angle, β, bank angle, ϕ and flight path angle, γ is given as: (Ito et. al., 2002)

$$\theta^{cmd} = \tan^{-1}\left(\frac{a^{cmd}b^{cmd} + \sin\gamma\sqrt{\left(a^{cmd}\right)^2 - \sin^2\gamma + \left(b^{cmd}\right)^2}}{\left(a^{cmd}\right)^2 - \sin^2\gamma}\right) \quad \text{... (3.17)}$$

where

$$\alpha^{cmd} = \cos\alpha^{cmd}\cos\beta \quad \text{... (3.18)}$$

$$b^{cmd} = \sin\phi^{cmd}\sin\beta + \cos\phi^{cmd}\sin\alpha^{cmd}\cos\beta \quad \text{... (3.19)}$$

3.1.2 Desired Dynamics

The desired dynamics block uses a proportional element to map the rate commands to desired acceleration terms (Ito et. al., 2002) (see Figure 3.2).

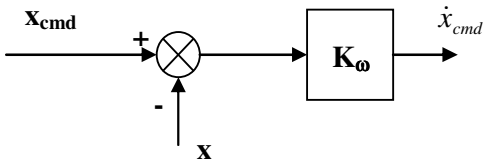

Figure 3.2: Proportional desired dynamics case

The term K_ω sets the bandwidth of the response. It also amplifies the error between the commanded state and its feedback term.

3.2 RBF Network Training and Optimisation

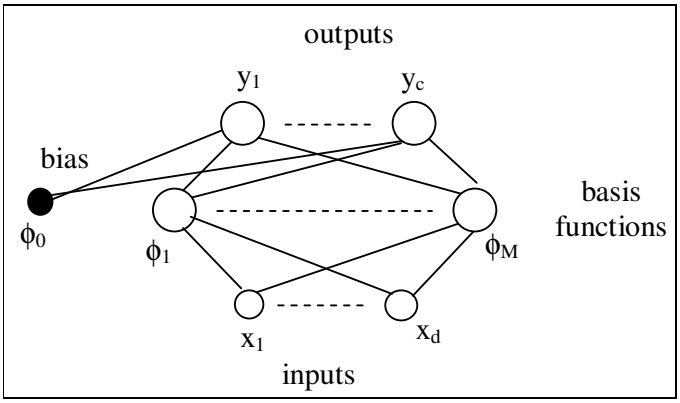

Figure 3.3: Radial basis function network

The radial basis function has originated from techniques for performing exact interpolation of a set of data points in a multi-dimensional space. This problem requires every input vector to be mapped exactly onto the corresponding target vector (Bishop, 1995). A schematic representing the network is shown in Figure 3.3.

Consider mapping from a d-dimensional space **x** to a one-dimensional target space *t*. The data set consists of N input vectors $\mathbf{x}^n$, with corresponding targets t^n. The aim is to find a function h(x) such that

$$h(x^n) = t^n \qquad n = 1,\ldots,N \qquad \textbf{... (3.20)}$$

The RBF approach introduces a set of N basis functions, one for each data point that has the form $\phi(\|x - x^n\|)$, where $\phi(\cdot)$ is a nonlinear function. The n^{th} function depends on the distance $\|x - x^n\|$. The output of the mapping is taken as a linear combination of the basis functions

$$h(x) = \sum_n \omega_n \phi(\|x - x^n\|) \qquad \textbf{... (3.21)}$$

This has the same form of a generalised linear discriminant function. The interpolation conditions (Eq. (3.20)) can be written in matrix form as:

$$\Phi w = t \qquad \textbf{... (3.22)}$$

where

$$t \equiv t^n \text{ , } w \equiv \omega_n \qquad \textbf{... (3.23)}$$

and the square matrix Φ has elements

$$\Phi_{nn'} = \phi(\|x - x^n\|) \qquad \textbf{... (3.24)}$$

Provided the inverse matrix Φ^{-1}exists Eq. (3-22) can be solved giving

$$w = \Phi^{-1} t \qquad \textbf{... (3.25)}$$

When the weights in Eq. (3.21) are set to the values given by Eq. (3.25), the function h(x) is a continuous differentiable surface which passes exactly through each data point (Bishop, 1995).

In the exact interpolation algorithm, many properties of the interpolation function are insensitive to the precise form of the nonlinear function $\phi(\cdot)$. There are several functions that have been identified to use, but the most common is the Gaussian function:

$$\phi(x) = \exp\left(-\frac{x^2}{2\sigma^2}\right) \quad \text{... (3.26)}$$

where σ is a factor that controls the smoothness properties of the interpolating function. The Gaussian is a localised basis function with the following property:

$$\phi \to 0 \text{ as } |x| \to \infty \quad \text{... (3.27)}$$

Revisiting the point that the surface passes exactly through each data point, the exact interpolating function for noisy data is usually a highly oscillatory function and is undesirable. Another drawback of the exact interpolation method is that the number of basis functions is the same as the number of patterns in the data set. In the case of large data sets, the mapping function is very costly to evaluate.

By introducing the following modifications, a smooth interpolating function in which the number of basis functions is determined by the complexity of the mapping rather than the size of the data set can be found (Bishop, 1995):

- The number *M* of basis functions need not be the same as the number of data points *N*. M is much less than N
- Basis function centres are not constrained to be given by the input data vectors. Suitable centres are determined as part of the training process
- A common σ is replaced by σ_j for each basis function. This parameter is also determined during training
- Bias parameters are included in the linear sum, compensating for the difference between the average value over the data set of the basis function activations and corresponding average of the targets.

Taking these changes into account, the following form of the radial basis function neural network mapping is obtained:

$$y_k(x) = \sum_{j=1}^{M} w_{kj}\phi_j(x) + w_{k0} \quad \text{... (3.28)}$$

w_{k0} can be absorbed into the summation by including an extra basis function, ϕ_0, for which the activation is set to 1. In the case of the Gaussian basis function,

$$\phi_j(x) = \exp\left(-\frac{\|x-\mu_j\|^2}{2\sigma_j^2}\right) \quad \text{... (3.29)}$$

x is the d-dimension input vector with elements x_i, and μ_j is the vector determining the centre basis function ϕ_j and has elements μ_{ji}. This is illustrated in Figure 3.3.

Radial basis functions have the property of best approximation. This means that in the set of approximating functions, there is one that has minimum approximating error for any function to be approximated.

The radial basis function has a two-stage training procedure. In the first stage, the parameters governing the basis functions are determined (μ_j, σ_j) using fast unsupervised methods. This means methods that only use the input data ($\{x^n\}$) and not the target data. The basis functions are then kept fixed while the second layer weights are found in the second part of training. This is usually the solution to a linear problem and is also fast. Assuming that the basis function parameters are chosen, the optimisation of the second layer weights can be discussed (Bishop, 1995).

Considering the RBF network mapping of Eq. (3.28), the bias parameters are absorbed into the weights to give:

$$y_k(x) = \sum_{j=0}^{M} w_{kj}\phi_j(x) \quad \text{... (3.30)}$$

In matrix notation, this is:

$$y(x) = W\phi \quad \text{... (3.31)}$$

with $W = w_{kj}$ and $\phi = \phi_j$

Since the basis functions are fixed, the network is equivalent to a single layer network. The weights can be optimised by the minimisation of a suitable error function. It is convenient to consider the sum-of-squares error (Bishop, 1995):

$$E = \frac{1}{2}\sum_n \sum_k \{y_k(x^n) - t_k^n\}^2. \quad \text{... (3.32)}$$

where t_k^n is the target value for output unit k when the network is presented the input vector x^n.

The error function is a quadratic function of the weights, and so its minimum can be found in terms of a set of linear equations. The weights are determined by

$$\Phi^T \Phi W^T = \Phi^T T \quad \text{... (3.33)}$$

where $(\mathbf{T})_{nk} = t_k^n$ and $(\mathbf{\Phi})_{nj} = \phi_j(x^n)$

The formal solution to the weights is given by

$$W^T = \Phi^* T \quad \text{... (3.34)}$$

where Φ^* is the pseudo-inverse of Φ.

Equation 3.33 is solved using singular value decomposition, avoiding problems of ill conditioning. Poor scaling of matrices leads to ill conditioning. The second layer weights are found by fast linear matrix inversion techniques.

3.3 Combined NDI-RBF

Applying the methodology described specifically to the nonlinear dynamic inversion, the state vector, x, control vector, u, and output vector y, is given as:

$$x = [x_1,...,x_{12}]^T = \begin{bmatrix} (u,v,w)^T \\ (p,q,r)^T \\ (\phi,\theta,\psi)^T \\ (x_E, y_E, h)^T \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} \in \Re^{12} \quad \text{... (3.35)}$$

$$u = [u_1,...,u_4]^T = \begin{bmatrix} (\delta_a, \delta_e, \delta_r)^T \\ \delta_T \end{bmatrix} = \begin{bmatrix} \delta \\ \delta_T \end{bmatrix} \in \Re^4 \quad \text{... (3.36)}$$

$$y = [y_1,..., y_4]^T = [\beta, p, \gamma, q]^T \in \Re^4 \quad \text{... (3.37)}$$

$$\dot{X}_1 = F_1(V, \alpha, \beta, p, q, r, \phi, \theta, h, \delta, \delta_T) \quad \text{... (3.38)}$$

$$\dot{X}_2 = F_2(V, \alpha, \beta, p, q, r, h, \delta, \delta_T) \quad \text{... (3.39)}$$

$$\dot{X}_3 = F_3(p, q, r, \phi, \theta, \psi) \quad \text{... (3.40)}$$

$$\dot{X}_4 = F_4(V, \phi, \theta, \psi) \quad \text{... (3.41)}$$

Equation (3.40) is solved simultaneously for the control inputs, (δ_a, δ_e, δ_r), needed to achieve the commanded accelerations (Pashikar et. al., 2007; Spaulding et. al., 2005; Ito et. al, 2002):

$$[\delta_a \quad \delta_e \quad \delta_r]^T = K_2(V, \alpha, \beta, p, q, r, h, \dot{X}_2, \delta_T) \quad \text{... (3.42)}$$

where the commanded accelerations $\dot{X}_2$ are given by:

$$\dot{X}_2 = [\dot{p} \quad \dot{q} \quad \dot{r}]^T = [\dot{p}_{com} \quad \dot{q}_{com} \quad \dot{r}_{com}]^T \quad \text{... (3.43)}$$

Full state feedback with measurements *u, v, w, p, q, r,* $\phi, \theta,$ and ψ is assumed. The desired inverse functions, K_2, in Eq. (3.42) can be replaced by a neural network to obtain the neuro-controller architecture:

$$\begin{aligned}\begin{bmatrix}\delta_a^d & \delta_e^d & \delta_r^d\end{bmatrix}^T &= NN_2\left(V,\alpha,\beta,p,q,r,h,\dot{X}_2,\delta_T\right)\\ &= \hat{w}_{02}+\hat{w}_2^T\hat{\Phi}_2\left(V,\alpha,\beta,p,q,r,h,\dot{X}_2,\delta_T\right)\end{aligned} \quad \text{... (3.44)}$$

where $\mathbf{w}_{02}$, $\mathbf{w}_2$, $\mathbf{\Phi}_2$ represent estimates of the network bias terms, the RBF weight vector and the basis functions vector respectively.

3.4 Additional Definitions

3.4.1 Controllability and Observability

The dynamical system $\dot{x} = Ax + Bu$, is said to be state controllable if, for any initial state $x(0) = x_0$, at any time $t_1 > 0$ and any final state x_1, there exists an input u(t) such that $x(t_1) = x_1$. Otherwise the system is said to be uncontrollable.

One method to check if a system is state controllable is to define the controllability matrix:

$$C = \begin{bmatrix}B & AB & A^2B & \ldots & A^{n-1}B\end{bmatrix} \quad \text{... (3.45)}$$

If and only if C has rank n (where n is the number of states), then the system is state controllable. The rank of a matrix is equal to the number of non-zero singular values of the matrix, that is, the number of linearly independent rows or columns of a full matrix (Skogestad and Postlethwaite, 1996).

The dynamical system $\dot{x} = Ax + Bu,\ y = Cx + Du$, is said to be state observable if, for any time $t_1 > 0$, the initial state $x(0) = x_0$ can be determined from the time history of the input u(t) and the output y(t) in the interval $[0, t_1]$. Otherwise the system is said to be unobservable. To check if a system is state observable, one can define the observability matrix:

$$O = \begin{bmatrix} C \\ CA \\ \ldots \\ CA^{n-1} \end{bmatrix} \quad \text{... (3.46)}$$

The dynamical system is state observable if and only if the observabilty matrix has rank n (where n is the number of states) (Skogestad and Postlethwaite, 1996).

This means that if a system is controllable, it can be brought from any initial state to a final state within a finite amount of time by its inputs, u. Also, a system is observable if, by measuring the output y(t) over a period of time, the values of all the individual states can be obtained. Controllability and oberservability are dependant on the number of states, observations, and inputs.

3.4.2 Condition Number

The condition number of a matrix is defined as the ratio between the maximum and minimum singular values:

$$\gamma(G) = \bar{\sigma}(G) / \underline{\sigma}(G) \qquad \textbf{... (3.47)}$$

An in-depth discussion on singular value decomposition is found in Skogestad and Postlethwaite (1996).

A matrix having a large condition number indicates that it is ill conditioned. This number is dependant on the scaling of the inputs and outputs. The condition number is used as an input-output controllability measure. When the condition number is small, the multivariable effects of uncertainty are likely to be minimal (Skogestad and Postlethwaite, 1996).

4 Simulation

4.1 System Parameters

The properties (mass, inertia, and geometry) of the F-16 aircraft are listed in Table 4-1, while its aerodynamic control surface properties and deflection conventions are given in Table 4-2.

Table 4-1: Aircraft mass, inertia and geometry properties

	Parameter	Value
m	Vehicle mass (kg)	9299
b	Wing span (m)	9.14
S	Wing area (m^2)	27.87
$\bar{c}$	Mean aerodynamic chord (m)	3.41
I_x	Roll moment of inertia ($kg.m^2$)	12874.8
I_y	Pitch moment of inertia ($kg.m^2$)	75673.3
I_z	Yaw moment of inertia ($kg.m^2$)	85551.8
I_{xz}	Product of inertia ($kg.m^2$)	1331.4
I_{xy}	Product of inertia ($kg.m^2$)	0
I_{yz}	Product of inertia ($kg.m^2$)	0

Table 4-2: Aerodynamic control surface properties

Symbol	Command Name	Deflection Limit	Rate Limit	Time Constant	Positive Sign Convention	Effect
δ_e	Elevator	±25º	60º/s	0.0495 sec lag	Trailing edge down	Negative pitching moment
δ_a	Aileron	±21.5º	80º/s	0.0495 sec lag	Right wing trailing edge down	Negative rolling moment
δ_r	Rudder	±30º	120º/s	0.0495 sec lag	Trailing edge left	Negative yawing moment, positive rolling moment

The state variables defined in the study were the following:

- velocity, V (ms^{-1})
- angle of attack, α (deg)
- sideslip angle, β (deg)
- roll angle, ϕ (deg)
- pitch angle, θ, (deg)
- yaw angle, ψ (deg)
- roll rate, p ($degs^{-1}$)
- pitch rate, q ($degs^{-1}$)
- yaw rate, r ($degs^{-1}$)
- north displacement, x_E (m)
- east displacement, y_E (m)
- altitude, $h = -z_E$ (m)
- power (%)

The control variables were as follows:

- throttle setting, δ_T (-)
- elevator deflection, δ_e (deg)
- aileron deflection, δ_a (deg)
- rudder deflection, δ_r (deg)

Four output variables were selected. These variables cover both the longitudinal and lateral modes

- pitch rate, q ($degs^{-1}$)
- flight path angle, γ (deg)
- sideslip angle, β (deg)
- roll angle, ϕ (deg)

4.2 Determination of System Matrices (A,B,C,D)

Huo's (2003) program was modified to determine the system matrices. Flight conditions were chosen for both the cruise and the refuelling. These were used in conjunction with

the state and control trimmed conditions as well as the derivative of the state equations to give the system matrices as the output.

4.3 Trim Program

The trim equations developed in Chapter 2 for the in-flight refuelling were implemented in an m-file in Matlab. The trimmed variables solved for were: α, β u, v, w, δ_e, δ_a, δ_r, δ_T.

An initial guess for the trimmed states was chosen

$$\mathbf{x}_0 = [\alpha_0, \beta_0, u_0, v_0, w_0, \delta_e, \delta_a, \delta_r, \delta_T] \quad \text{... (4-1)}$$

This was used to solve the objective function made up of:

$$\begin{aligned} X_{total} &= 0 \qquad Y_{total} = 0 \qquad Z_{total} = 0 \\ L_{total} &= 0 \qquad M_{total} = 0 \qquad N_{total} = 0 \end{aligned} \quad \text{... (4-2)}$$

$$\begin{aligned} V - \sqrt{u^2 + v^2 + w^2} &= 0 \\ \tan\alpha - \frac{w}{u} &= 0 \\ \sin\beta - \frac{v}{V} &= 0 \end{aligned} \quad \text{... (4-3)}$$

The formulation of X_{total}, Y_{total}, Z_{total} L_{total}, M_{total}, and N_{total} has already been discussed in Chapter 2.

The solution of the trimmed states was used as the commanded states in the nonlinear dynamic inversion scheme.

4.4 RBF Training

The radial basis function network was trained and set up in an m-file. The final trained neural network was called from Simulink via a Matlab function block. Three radial basis function neural networks were set up in Matlab using the Netlab Toolbox developed by Ian Nabney (2002). The individual networks consisted of 75 neurons and were trained over 5000 cycles each. The Gaussian activation function was used. The number of

neurons was selected to give the smallest training/prediction errors while taking computational constraints into account.

The inputs in each neural network were:

- α^{meas}
- β^{meas}
- p^{meas}
- q^{meas}
- r^{meas}
- $\dot{p}^{des}$
- $\dot{q}^{des}$
- $\dot{r}^{des}$
- x_{cg}

Three single outputs were defined in each network

- δ_e
- δ_a
- δ_r

Values for the 9 input variables were generated in Simulink using the random number generator set within the designed limits of the state variables. This was the basis of the training data. These randomly generated inputs were used in the Eq (3.13).

α, β, p, q, r and cg position were used to find the dimensional moments stability derivatives L_{δ_a}, L_{δ_r}, M_{δ_e}, N_{δ_a}, N_{δ_r}, L_β, L_p, L_r, M_α, M_q, N_β, L_p, L_r. Having all the required parameters, Eq (3.13) was then solved for δ_e, δ_a, δ_r.

At this point, a complete set of inputs $\begin{bmatrix}\alpha^{meas} & \beta^{meas} & p^{meas} & q^{meas} & r^{meas} & \dot{p}^{des} & \dot{q}^{des} & \dot{r}^{des} & x_{cg}\end{bmatrix}$ can give a complete set of outputs $\begin{bmatrix}\delta_e & \delta_a & \delta_r\end{bmatrix}$. Thus, the neural network could be trained to invert the plant dynamics to provide the correct command deflections to maintain the aircraft is steady flight.

4.5 Simulink Model

The model built in the Simulink environment was based on the work of Meyer (2005). Meyer had built a model for the purposes of controlling an aircraft for the landing phase while experiencing adverse weather conditions. The model was modified to consider the case of controlling the UCAV during aerial refuelling.

The following blocks were implemented in Simulink:

- Flight conditions
- ISA atmosphere model
- Equations of motion
  - Forces due to mass
  - Forces due to thrust
  - Aerodynamic forces
- Trim function
- Command inverter
- Desired dynamics
- Dynamic Inversion
- Time lag
- Auxiliary block

The thrust deflection, δ_T, used in the Simulink model is not part of the dynamic inversion, but rather comes from the requirement to keep the aircraft in steady flight and is used as an output from the trim program.

The complete Simulink model for the nonlinear dynamic inversion controller with neural network is shown in Figure 4.1

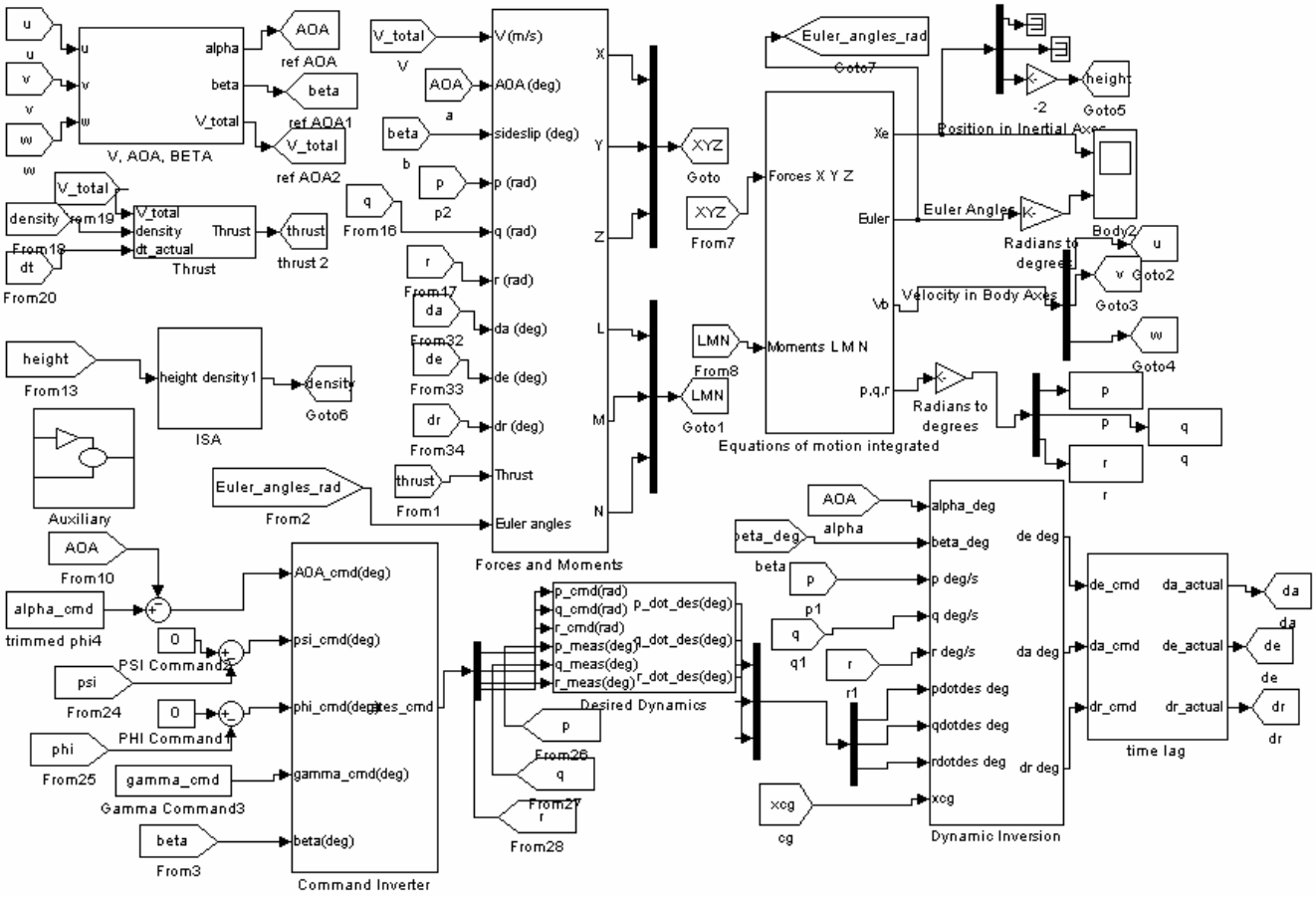

Figure 4.1: Complete Simulink model for NDI based neurocontroller

4.6 Complete Controller Model

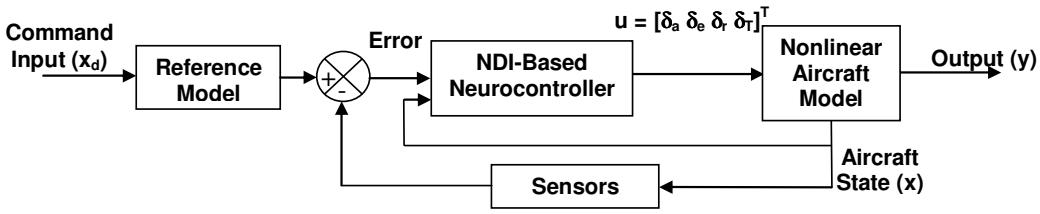

Figure 4.2: Overall controller model

A global view of the controller can be seen in Figure 4.2. x_d is the commanded states (input to reference model) of the aircraft. This is compared to the state variables in the feedback loop, and the error is sent to the dynamic inversion neural network controller. Manipulations are performed in the controller block (command inversion, desired dynamics and nonlinear inversion), and the output signal, u, is fed into the nonlinear plant. The output is y.

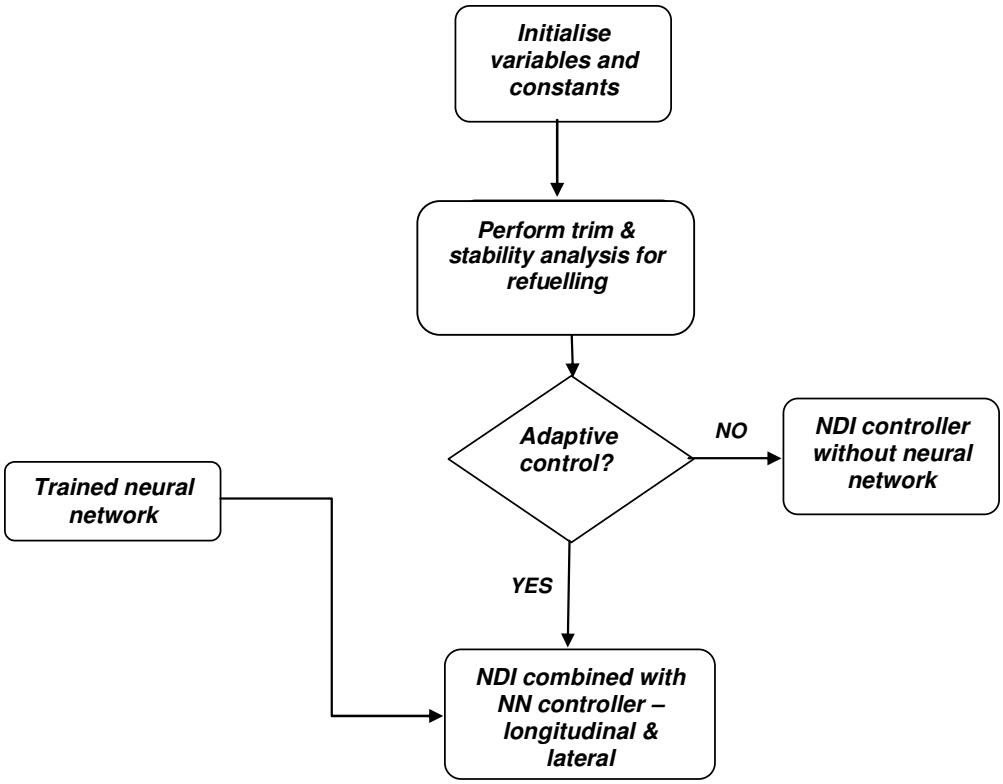

Figure 4.3: Simulation flow diagram

The simulation flow is outlined in Figure 4.3. In order to control the UCAV, all variables that require specifying and constants are initialised in Matlab. Thereafter, for the flight condition trim/modal and stability analysis are performed. If adaptive control is required, the trained neural network can be used. In the case of no neural network being simulated, then no further processing is required in order to perform the control.

5 Model Validation

5.1 Aircraft Trim Conditions

The aircraft model used in the present study was validated against two aircraft cruise cases from Stevens and Lewis (1992). The aircraft cruise speed in both cases was 153 m/s and the altitude used in the validation was 0 m, that is, sea level. The distinguishing factor between the two cases was the centre of gravity position, the first case having a cg position of 0.3 $\bar{c}$, and the second case cg position was 0.38 $\bar{c}$

The validation data is presented in Table 5-1. Both sets of data are presented alongside each other as well as the Matlab generated data to compare the accuracy of the aircraft model. The trimmed state variables as well as the trimmed control variables are also given.

Table 5-1: Comparison of trimmed aircraft conditions

		cg = 0.3 $\bar{c}$		**cg = 0.38 $\bar{c}$**	
		Simulated Results	**Stevens & Lewis**	**Simulated Results**	**Stevens & Lewis**
V	m/s	153	153	153	153
α	deg	2.257	2.257	2.028	2.028
β	deg	-6.217e-015	2.349e-07	3.316e-011	3.1e-08
ϕ	deg	0	0	0	0
θ	deg	2.257	2.257	2.028	2.028
p	deg/s	0	0	0	0
q	deg/s	0	0	0	0
r	deg/s	0	0	0	0
δ_T	-	0.14851	0.1485	0.13256	0.1325
δ_e	deg	-1.930	-1.931	-0.056	-0.056
δ_a	deg	1.715e-015	-7.0e-08	-9.388e-012	-5.1e-07
δ_r	deg	-1.686e-014	8.3e-7	8.102e-011	4.3e-06

The results of the Matlab simulation and the cited values are the same up to three decimal places in some cases. Some of the values (viz. the sideslip angle, aileron deflection and rudder deflection) are extremely small, and can be considered to be zero. Thus, the model is accurate in estimating the trim conditions.

5.2 Eigenvalues

The eigenvalues were found and compared for the single case of the aircraft in cruise at a speed of 153 m/s at sea level with a forward cg of 0.3 $\bar{c}$. The eigenvalues are given in Table 5-2 and are compared for all longitudinal and lateral modes.

Table 5-2: Comparison of eigenvalues

	Simulated Results	**Stevens & Lewis**
Short Period	-1.2906 ± 1.2679i	-1.2904 ± 1.4922i
Phugoid	-0.0097 ± 0.0689i	-0.0087± 0.0740i
Dutch Roll	-0.4291 ± 3.2521i	-0.4399 ± 3.220i
Spiral	-0.0021	-0.00128
Roll	-3.6168	-3.601

The comparison shows that the accuracy for the short period mode, Dutch roll and roll mode lies to two decimal places at the most, while the spiral mode and the Phugoid are in the same region of values. The discrepancies are not large enough to cause the model to be inaccurate, and the model will be deemed sufficient to analyse the behaviour of the F-16 accurately.

5.3 Dynamic Behaviour

In addition to validating the trimmed data and eigenvalues, the accuracy of the dynamic behaviour was also considered. As with the eigenvalues above, a single case is considered here for the model validation. This is the same as given above for the eigenvalues with the F-16 flying at a cruise speed of 153 m/s at sea level with a forward cg of 0.3$\bar{c}$. The dynamic parameters are presented in Table 5-3 and are compared for the short period, Phugoid, and Dutch roll modes.

Table 5-3: Comparison of dynamic parameters

	Simulated Results		Stevens & Lewis		Error [%]	
Dynamic Parameter	ζ [-]	T [s]	ζ [-]	T [s]	ζ [-]	T [s]
Short Period	0.7133	4.96	0.628	4.21	13.60	15.04
Phugoid	0.1496	88.35	0.117	84.9	27.8	3.85
Dutch Roll	0.1308	1.93	0.135	1.95	3.11	1.02

The damping coefficients and periods are compared. The Dutch roll mode is the most accurate, with the short period following, and the Phugoid is the least accurate. The values are not exact but they are in the same range. The model can still be considered accurate in the estimation of the dynamic behaviour. The discrepancy in the values given above arises directly from the slight disagreement between the eigenvalues given by Matlab and those found in Stevens and Lewis.

6 Results and Discussion

6.1 Inherent Behaviour

The frequency response plot of the UCAV is shown in Figure 6.1 at the start of refuelling

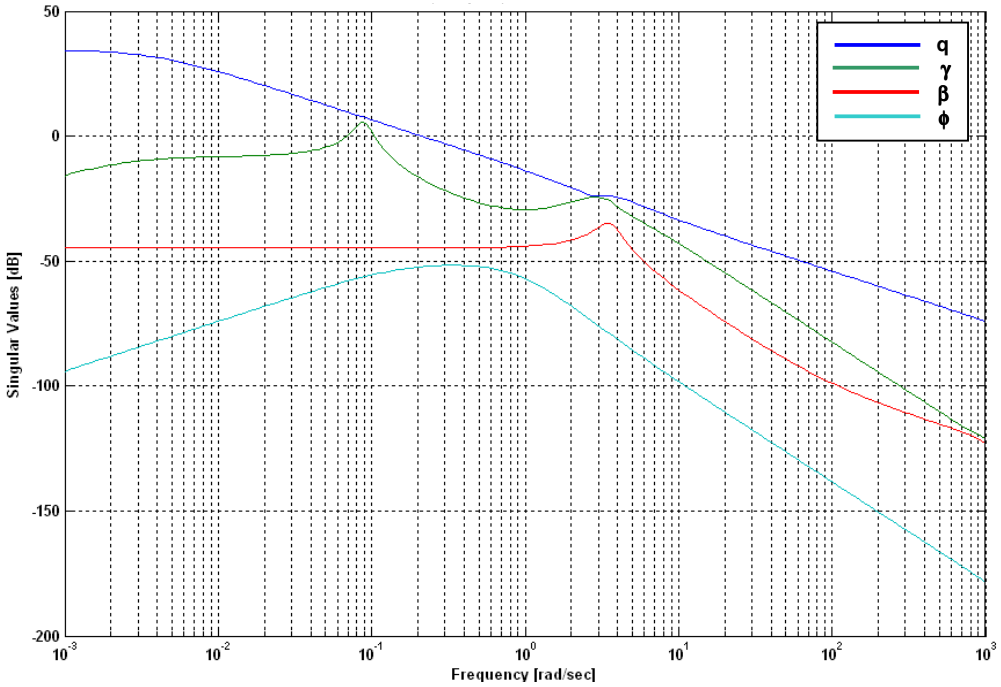

Figure 6.1: Frequency response plot of UCAV – start of refuelling

In multi-input multi-output (MIMO) systems, the maximum singular values need to be small at high frequencies for good stability robustness (Stevens and Lewis, 1992). One observes good stability robustness. At low frequencies, the condition number of the aircraft is high indicating that inherently the plant can pose some control problems. The rank of the controllability and observability matrices were found to be 10 for the cg position considered, indicating that all states of the system were observable and controllable at the onset of refuelling.

Figure 6.2 shows the four cases chosen to be analysed during the flight refuelling process. The cg has a parabolic shape variation with mass (Figure 6.2). The initial mass, when fuelling starts, was taken as 9298.6 kg with the cg at 0.35. The maximum cg position was found to be 0.406.

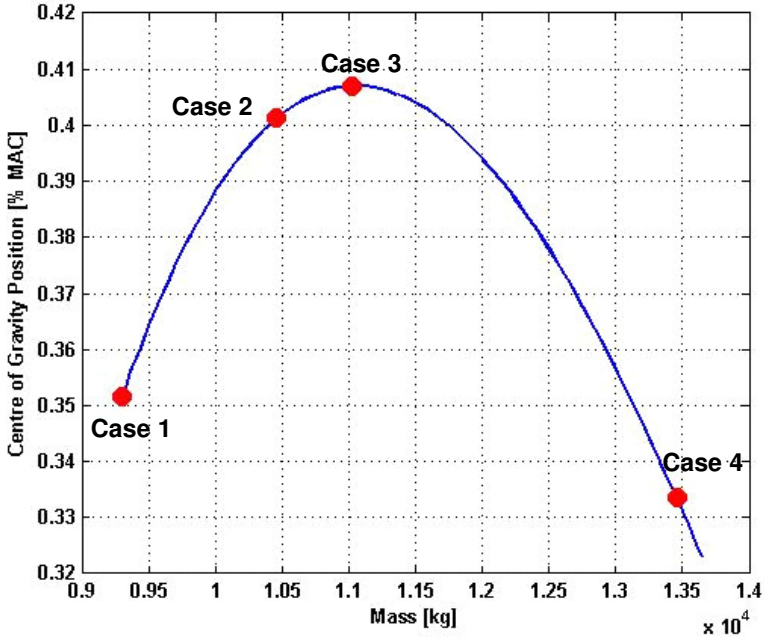

Figure 6.2: Cases considered in refuelling inherent behaviour analysis

Table 6-1 gives the changing eigenvalues of the aircraft for the four cases considered.

Table 6-1: Eigenvalue movement during refuelling

Mode	Case 1	Case 2	Case 3	Case 4
Longitudinal Mode	-1;-0.0014	-1;-0.0015	-1;-0.0015	-1;-0.0015
Dutch Roll	-0.271±2.45i	-0.278±2.37i	-0.287±2.38i	-0.347±2.67i
Spiral	-0.00491	-0.00536	-0.00488	-0.00172
Phugoid	-0.0193±0.118i	-0.00855±0.1i	-0.0083±0.093i	-0.122±0.056i
Roll	-1.8	-2.18	-2.21	-1.5

The Dutch roll movement in Table 6-1 shows that as the mass increases, the poles move further into the left hand plane, the mode gets more stable as mass is introduced to the system. The movement of the poles below the real axis along the imaginary axis shows that initially the mode becomes less oscillatory, but when the cg passes the turning point and starts to decrease again, the mode becomes more oscillatory again. The Phugoid mode pole movement in Table 6-1 shows that as the mass is increased the eigenvalue becomes closer to the imaginary axis and then moves left again. This mode can be viewed

to be less stable initially. The mode becomes more oscillatory as the aircraft mass increases.

The SPO decomposed into two first order modes and remains largely unchanged throughout the flight refuelling process. The decomposition of the SPO can be attributed to the addition of mass into the system, as well as an aft cg. When mass is introduced steadily and the cg moves further aft, the results show that the longitudinal dynamics are affected. In the model validation, (Chapter 5), results were presented for a forward cg position (Table 5-2) with no refuelling. However, when examining the eigenvalues for sea-level cruise (no refuelling) as described earlier considering an aft cg, a similar trend is observed with the short period dynamics as seen in Table 6-2. Table 6-2 gives the 0.3 $\bar{c}$ results from Table 5-2 compared to the 0.38 $\bar{c}$ eigenvalues for the oscillatory modes. The decomposed longitudinal mode eigenvalues for nominal cruise at sea level with cg at 0.38 are -1 and -0.0019. This is similar to that presented in Table 6-1 for the refuelling case.

Table 6-2: Comparison of eigenvalues for forward and aft cg positions

	0.3 $\bar{c}$		0.38 $\bar{c}$
Short Period	-1.2906 ± 1.2679i	Longitudinal mode	-1; -0.0019
Phugoid	-0.0097 ± 0.0689i	Phugoid	-0.0110 + 0.1138i
Dutch Roll	-0.4291 ± 3.2521i	Dutch Roll	-0.4045 - 3.0098i

The dynamic behaviour of the aircraft is presented in Table 6-3.

Table 6-3: Natural frequencies, damping coefficients and settling times

	ω_d [rad/s]	**ζ**	**T [s]**	**t_s [s]**
Phugoid [cg case 1]	0.120	0.161	52.360	207.25
[cg case 2]	0.100	0.085	62.832	467.83
[cg case 3]	0.0933	0.089	63.084	481. 93
Dutch Roll [cg case 1]	2.460	0.110	2.56	14.76
[cg case 2]	2.390	0.117	2.65	14.39
[cg case 3]	2.400	0.120	2.64	13.94
[cg case 4]	2.700	0.129	2.35	11.52

	ω_d [rad/s]	ζ	T [s]	t_s [s]
Spiral [cg case 1]	0.0049	1	-	814.668
[cg case 2]	0.0054	1	-	746.268
[cg case 3]	0.0049	1	-	819.672
[cg case 4]	0.0017	1	-	2326
Roll [cg case 1]	1.80	1	-	2.22
[cg case 2]	2.18	1	-	1.84
[cg case 3]	2.21	1	-	1.81
[cg case 4]	1.50	1	-	2.67
Decomposed Longitudinal [1]	1.00	1	-	4.00
Decomposed Longitudinal [2]	0.014	1	-	2857

Table 6-3 shows the change in dynamic parameters as the aircraft is refuelled. The damping of the Phugoid mode decreases as the cg moves rearward to the maximum point at cg position three. After the turning point of case three, the cg starts to move forward. This causes the damping to increase again. The natural frequency decreases first as the cg moves to case two and three. This means that the mode gets less oscillatory. After the turning point when case four is reached, the natural frequency increases again.

The Dutch roll mode, which is a fast mode, has damping that is moderate compared to other modes. The damping of the aircraft in this mode is seen to steadily increase as the mass increases. The natural frequencies however, decrease and then increase again, showing a less oscillatory to more oscillatory transition.

For the spiral mode, the damping increases and then decreases again as the maximum cg point is reached. The roll mode shows the heaviest damping which steadily decreases as the aircraft mass is increased.

The aircraft trim conditions change with the mass of the aircraft as the vehicle is refuelled. The longitudinal control surfaces constantly need to change their position in order to maintain steady level flight. Considering the case of cruise, all trimmed lateral variables are zero and unchanging, so only the longitudinal trimmed variable change with mass will be shown.

The aircraft trimmed angle of attack variation with aircraft mass is plotted in Figure 6.3. The trimmed angle of attack has a half-parabolic variation with change in mass.

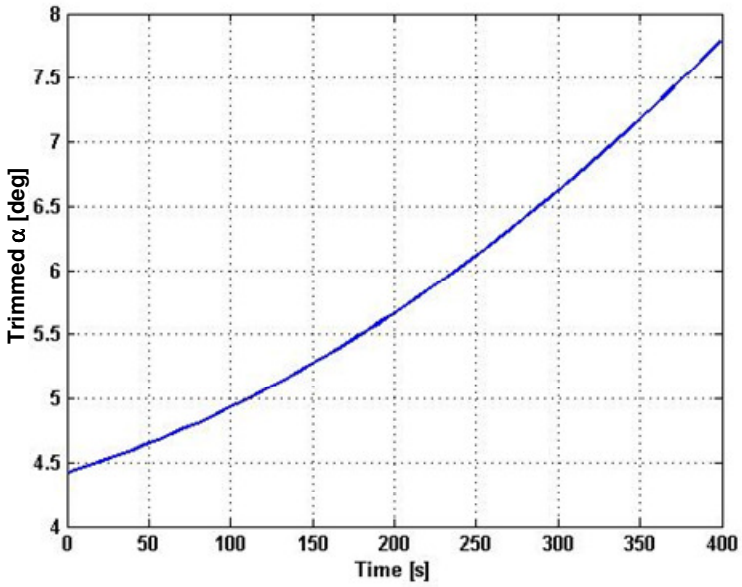

Figure 6.3: Refuelling trimmed α change with mass

The elevator deflection required to trim the aircraft changes with mass (see Figure 6.4). The elevator deflection required follows the same trend as the change in cg position.

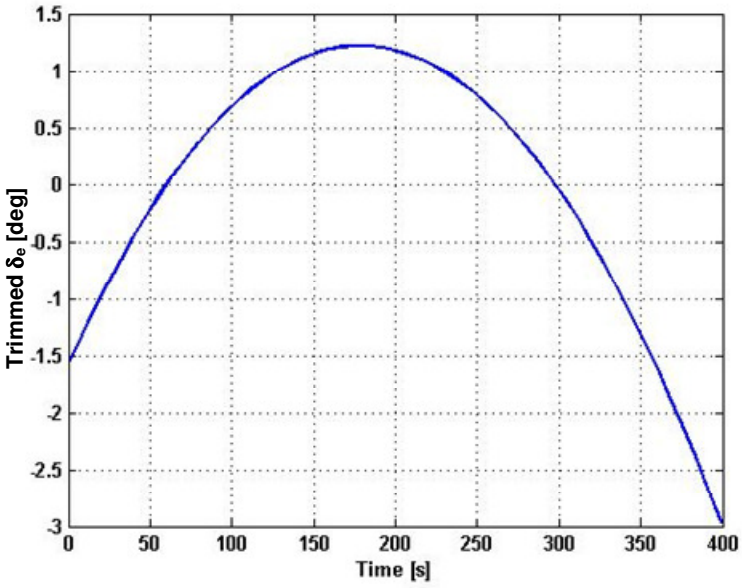

Figure 6.4: Refuelling trimmed elevator deflection change with mass

As the aircraft mass changes the aircraft throttle position will change to keep the aircraft in steady flight. This is shown in Figure 6.5. A higher throttle deflection is required as mass increases.

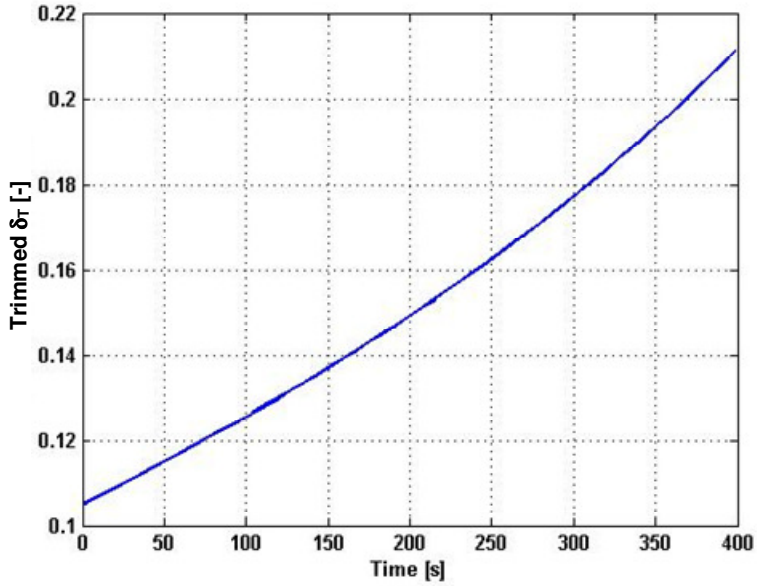

Figure 6.5: Refuelling trimmed throttle position change with mass

Figure 6.6 shows the variation of the forward speed, u, with the aircraft mass. The required aircraft forward speed decreases as the mass increases

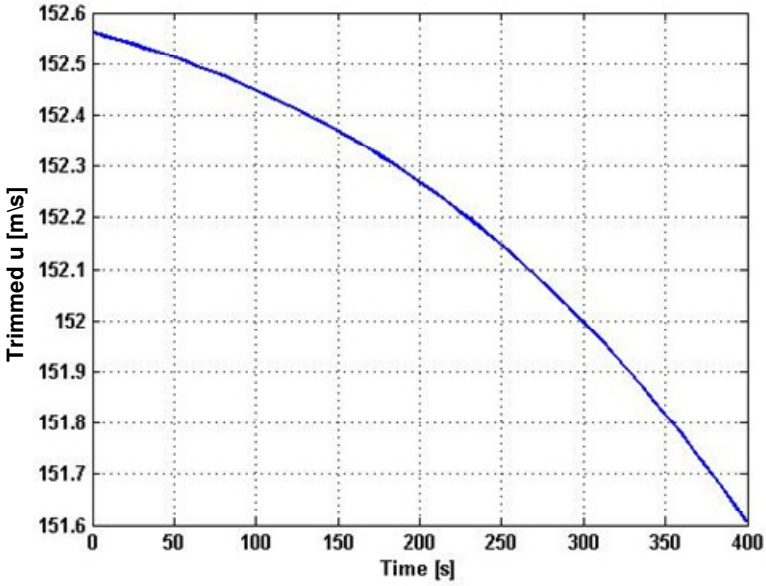

Figure 6.6 Refuelling trimmed forward speed, u, change with mass

6.2 Controlled Behaviour

6.2.1 Neural Network Training

In order for the neural networks to learn the inverted dynamics of the aircraft as described in Chapter 3 and Chapter 4, data was generated as shown in Figure 6.7 for the measured α, β, p, q, r, ϕ, and ψ. The figure shows data for only α^{meas}. Random data of the form in Figure 6.7 was used for all the other variables mentioned. This data served as the inputs to the dynamic inversion equations, and was used together with the desired rates p, q, and r as explained previously.

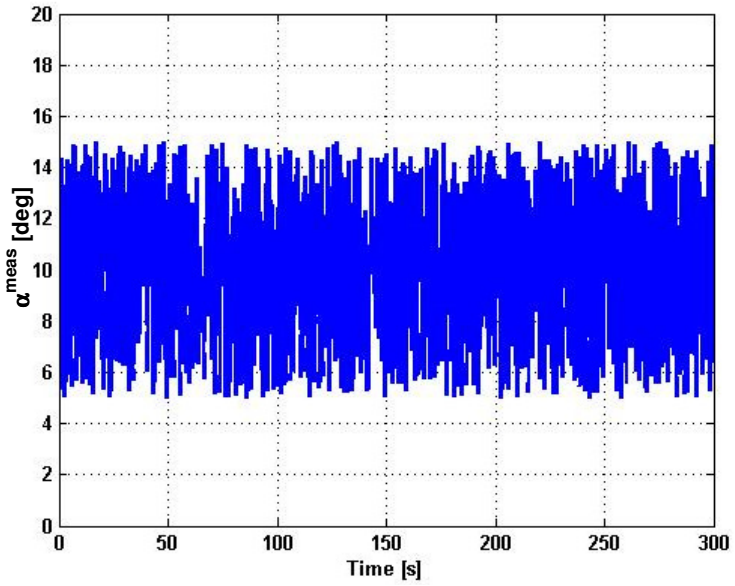

Figure 6.7: Neural network training data – α^{meas}

The neural network comprising 75 neurons was trained over 5000 cycles and the characteristics are summarised in Table 6-4. The weights and centres are large arrays and are not repeated here.

Table 6-4: Characteristics of RBF used in study

Netlab Toolbox Variable	Description	Value
actfn	hidden unit activation function	'gaussian'
outfn	output error function	'linear'
nwts	total number of weights and biases	826
c	centres	75×9 array

Netlab Toolbox Variable	Description	Value
w_i	squared widths	1×75 array
w_2	second layer weight matrix	75×1 array
b_2	second layer bias vector	6138.4

The networks were tested after training and predictions were compared with the actual values. The neural network errors are plotted in Figure 6.8 to Figure 6.10 for the elevator, aileron and rudder respectively. The elevator prediction was the most accurate. The error in prediction of the elevator deflection reached a maximum of just over 1%. However, for the aileron and rudder, excessively high errors were found in certain instances. Errors of 6000% in the aileron prediction and 200% in the rudder prediction can be considered as single outliers. The average of the errors in the aileron prediction was 5.8% and in the rudder prediction 0.5 %. This means that the majority of the results are reasonable. However, the neural network is liable to make any erroneous prediction up to 6000% and 200% of the original value.

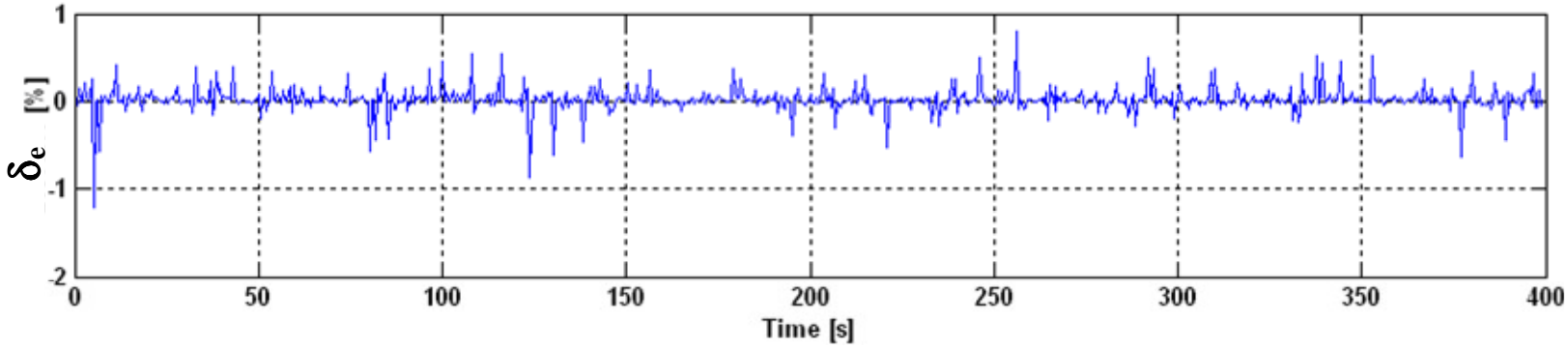

Figure 6.8: Neural network error in predicting commanded elevator deflection

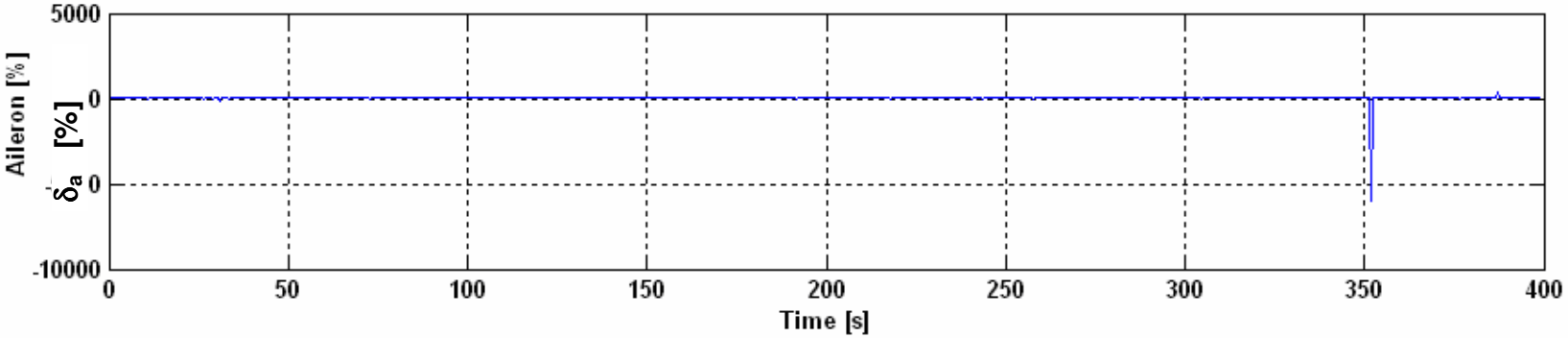

Figure 6.9: Neural network error in predicting commanded aileron deflection

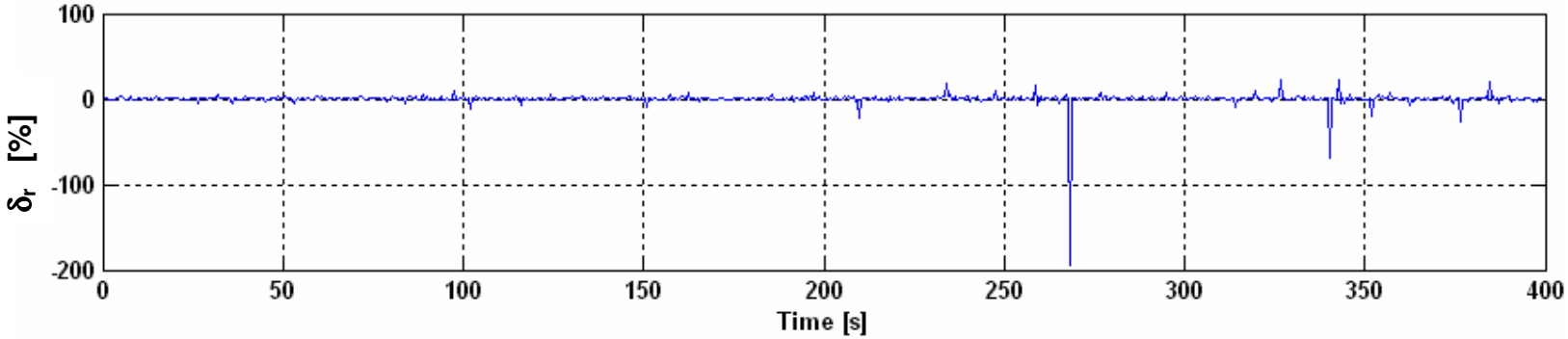

Figure 6.10: Neural network error in predicting commanded rudder deflection

6.2.2 Nonlinear Dynamic Inversion Controller without Neural Network

Before the neural network results can be analysed, the conventional DI controller results are presented. Figure 6.11 shows the UCAV state variables over the refuelling period.

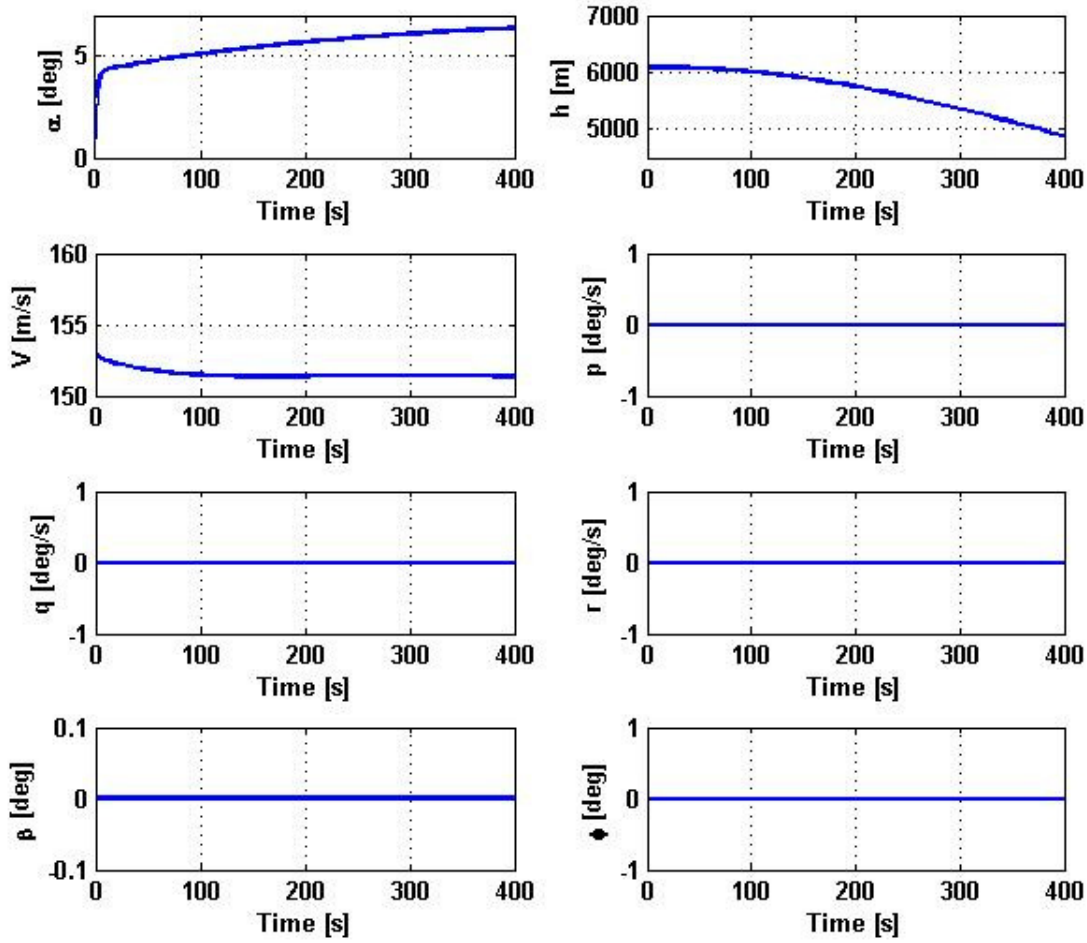

Figure 6.11: Controlled state variables – NDI without NN

The aircraft angle of attack is not constant, nor can the controller maintain the aircraft at a constant altitude and speed during the refuelling period. No change is observed in the pitch rate, sideslip angle, and roll angle. The commanded control surface deflections are shown in Figure 6.12. These are found to be very small and insufficient to keep the aircraft in steady flight.

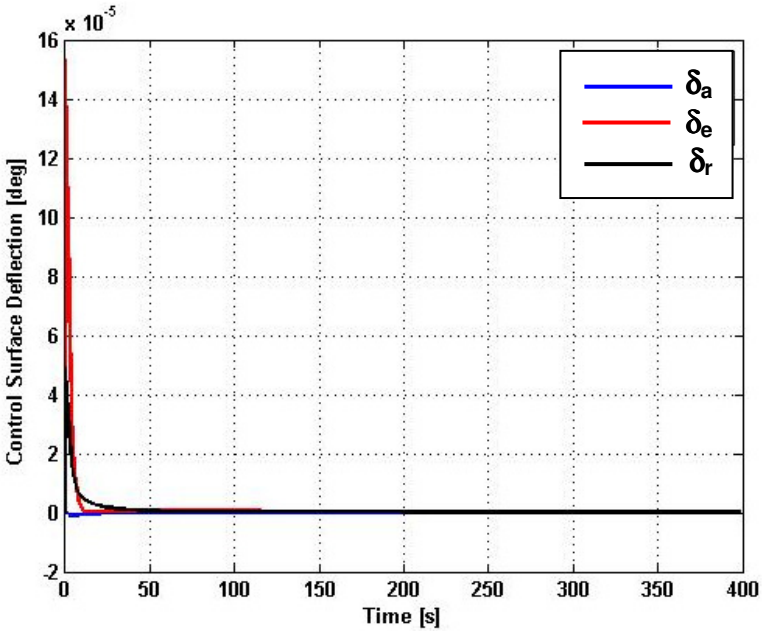

Figure 6.12: Commanded control surface deflections – NDI with no NN

The time domain results presented show that the performance of this controller is inadequate. This can be attributed to the use of the aerodynamic model of Ito et. al. (2002) in conjunction with Morelli (1998). In order to invert the plant dynamics, Eq. (3.9) to (3.11) of Ito et. al. (2002) has to be used, whereas the aerodynamic coefficients required for the force and moment integration are calculated using Morelli's (1998) Eq. (2.26) to (2.32). The higher order terms in Eq. (2.26) to (2.32) which have cross coupling between longitudinal and lateral variables are not present in the dynamic inversion. In order to calculate the Euler angles and body rates for the inversion, Eq. (2.26) to Eq. (6.32) are used. In the inversion, the model has changed, and it is clear that this controller is not sensitive to changes in the aircraft model.

6.2.3 Longitudinal Nonlinear Dynamic Inversion Controller with Neural Network

The DI controller coupled with the NN shows excellent results. All states are maintained as constant during the 400 seconds duration of the flight refuelling (see Figure 6.13). The elevator deflection required to maintain steady flight is shown in Figure 6.14. This variation is well within the physical limits of the elevator actuator. The results here can be attributed to the low inversion error of the trained neural network. The results also show that the controller

with the NN is less sensitive to the model of Ito et. al. (2002) and Morelli (1998) as explained earlier. The NN is trained using the model of Ito et. al. (2002) and is implicitly used in this controller.

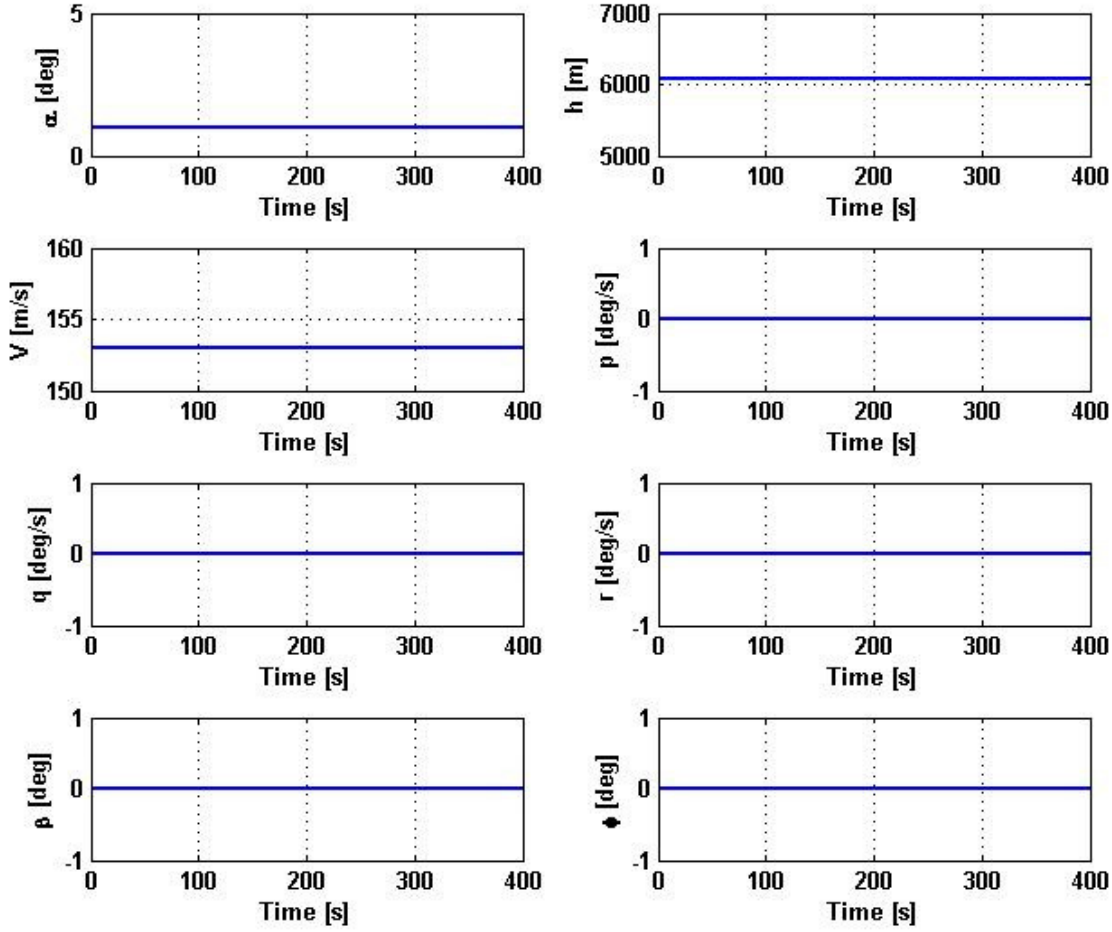

Figure 6.13: Controlled state variables vs. time – Longitudinal NDI-based neurocontroller

This result compared to the basic NDI controller with no NN ties up with the conclusions of Li et. al. (2001). The neural controller was able to overcome modelling errors which posed a challenge to the NDI controller without the NN.

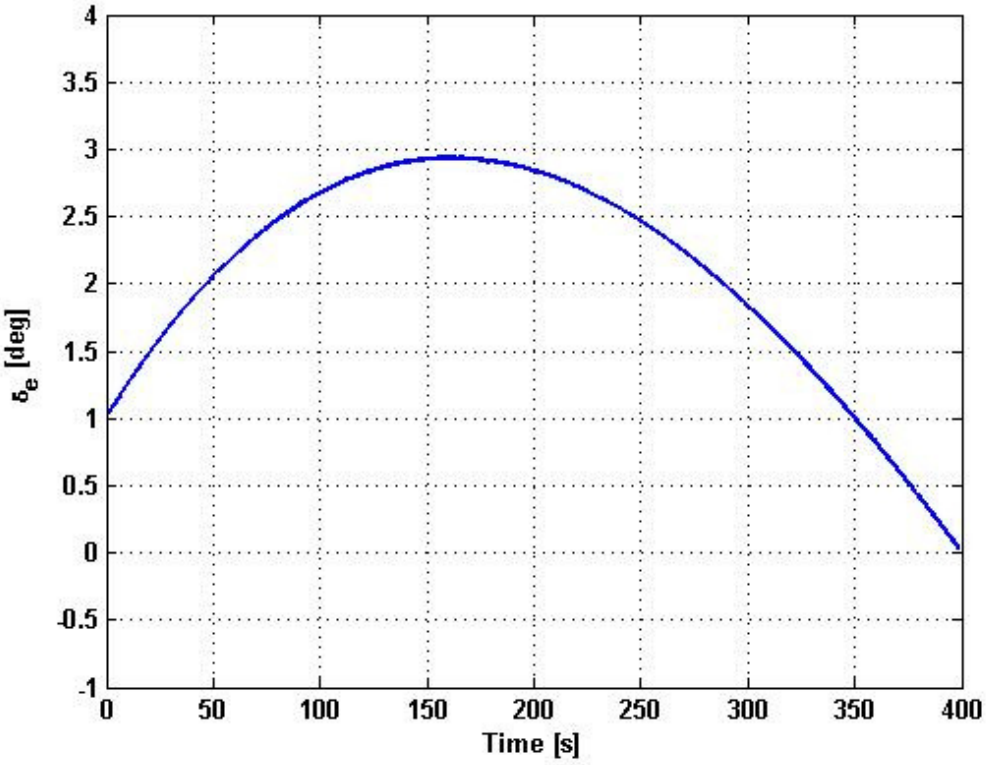

Figure 6.14: δ_e vs. time – Longitudinal NDI-based neurocontroller

6.2.4 Lateral Nonlinear Dynamic Inversion Controller with Neural Network

The results of the lateral controller are promising. The sideslip angle shown in Figure 6.15 does not stay steady, a small deviation is observed. The sideslip angle quickly steadies to a fixed value thereafter. However all other states are steady throughout the refuelling. The aileron and rudder deflections are shown in Figure 6.16 and Figure 6.17. These are within the physical limits. The aileron deflection was observed to be unaffected by the refuelling process/mass increase of the aircraft. The rudder deflection followed a similar trend to that of the elevator in Figure 6.14. It is also observed that the commanded aileron and rudder deflections are non zero. This is to maintain the sideslip angle that was observed in Figure 6.15. The adequate performance of the lateral neural controller over the plain DI controller can also be attributed to the addition of the RBF network, which is less sensitive to model errors as described earlier.

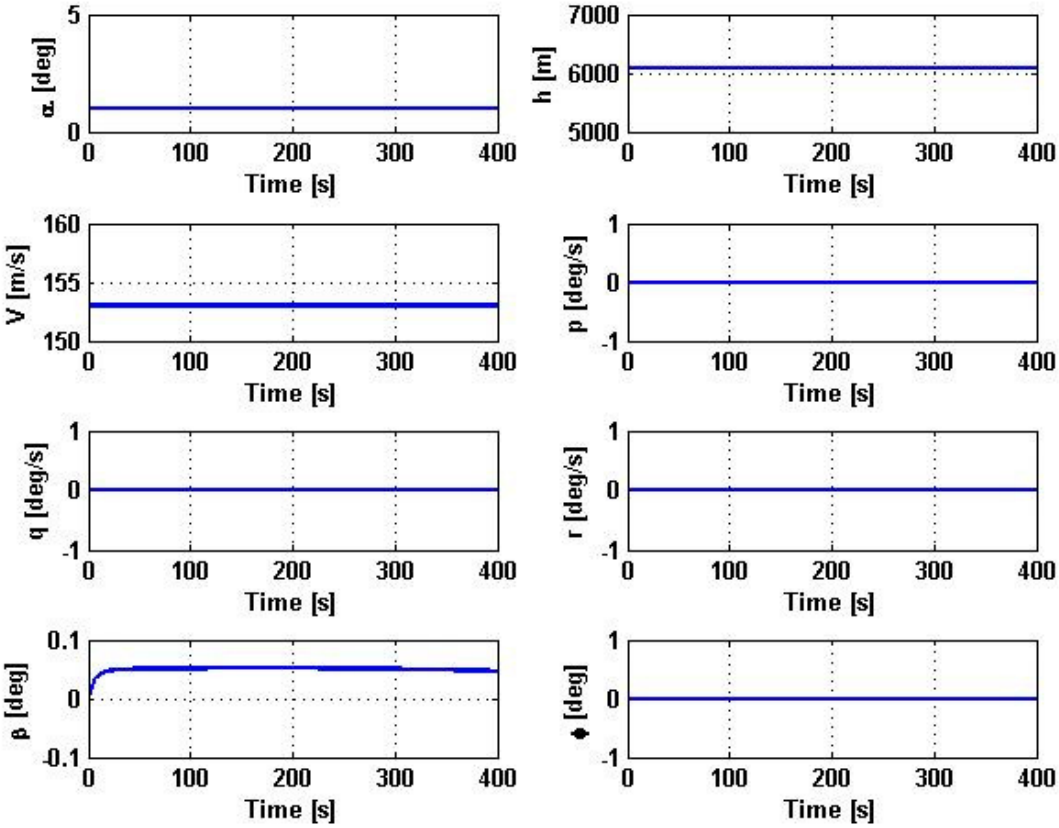

Figure 6.15: Controlled state variables vs. time – Lateral NDI-based neurocontroller

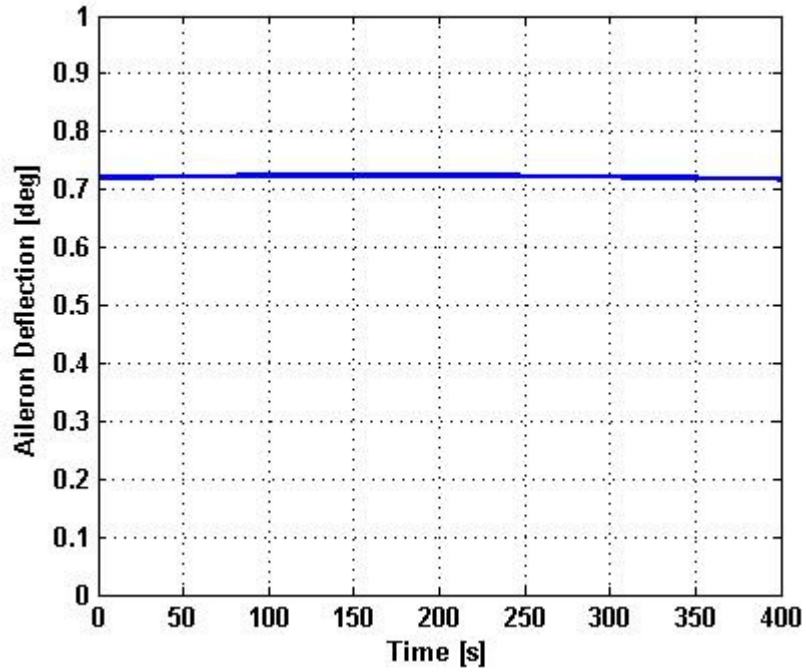

Figure 6.16: δ_a vs. time – Lateral NDI-based neurocontroller

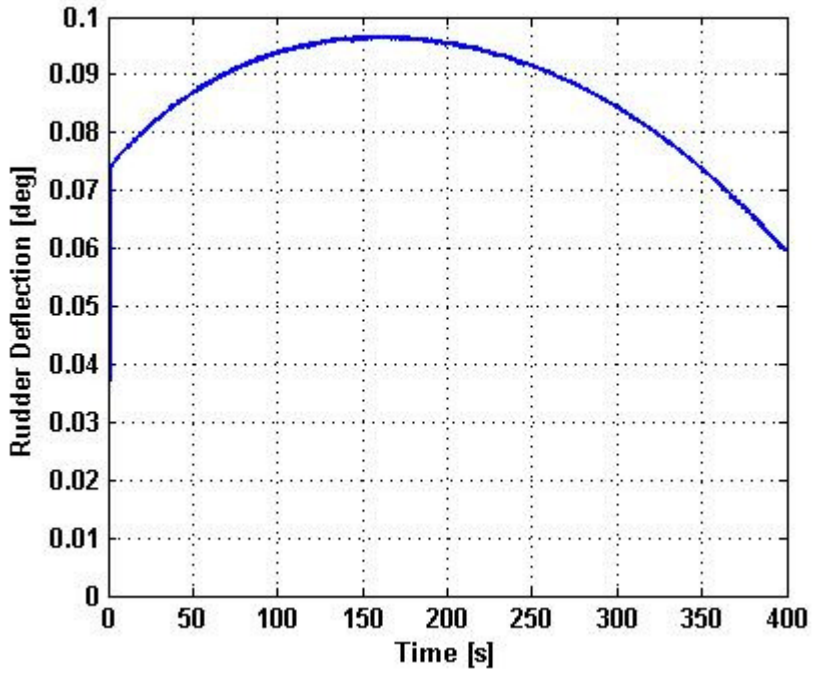

Figure 6.17: δ_r vs. time – Lateral NDI-based neurocontroller

7 Conclusions and Recommendations

7.1 Conclusions

The background leading to the modern day UCAV, as well as issues surrounding current developments has been researched and presented.

- UAVs/UCAVs have been identified to be an essential part of a nation's defence arsenal.
- Using automated aerial refuelling (AAR), the modern UCAV can achieve extended endurances.
- Studies also highlighted that the success of an unmanned vehicle lies in the application of neural network control techniques, which are being widely researched as they offer many solutions to the limitations of classical control.

For the in-flight-refuelling model, a nonlinear six-degree-of-freedom model including the centre of gravity movement during refuelling was built. Published data verified the accuracy of the aircraft model.

Modal analysis of the aircraft during refuelling was done.

- When refuelling, the cg variation with increasing mass was found to be parabolic.
- For the Dutch roll mode, as refuelling occurs, the mode is initially less oscillatory, but as the mass increases further, the cg starts to move forward again and the Dutch roll mode becomes more oscillatory.
- The opposite is observed with the Phugoid mode.
- The increase in mass pushes the eigenvalues further into the left plane – increased stability is achieved with the higher mass.
- The aircraft trim conditions change as the mass is increased.

A dynamic inversion controller with no neural network was used initially. With respect to the nonlinear dynamic inversion controller without neural networks, the following can be concluded:

- The performance of the controller was not satisfactory.

- This is attributed to the difference in aerodynamic models (Ito et. al., 2002; Morelli, 1998) used to evaluate the aerodynamic forces in the 6 DOF model and in the inversion.
- This controller was not sensitive to model discrepancies.

After implementing the dynamic inversion controller with the radial basis function network, the following can be concluded:

- The error in the elevator prediction was the lowest.
- The longitudinal DI controller with NN gave excellent results.
- Control surface deflections were within the physical limits of the aircraft.
- The lateral DI controller with NN showed promising results.
- Results are dependent on the training of the neural network via the number of neurons used.

Finally, it can be concluded that a nonlinear dynamic inversion controller with a neural network can be applied to control a UCAV during aerial refuelling.

7.2 Recommendations for Future Work

The following recommendations are made for future work:

1. Investigate the dynamic inversion scheme using a multi-layered-perceptron neural network architecture.
2. Model the effects of the changing inertia tensor in the 6 DOF model.
3. Aerodynamic effects of tanker-receiver can be added in the dynamic model.
4. Use of historical data to train neural networks instead of random data.
5. Consider implementation of on-line neural network in the nonlinear dynamic inversion controller.
6. Explore the idea of suitably combining the longitudinal and lateral controller.

References

Battipede, M. Gili, P. Napolitano, M.R. Perhinschi, M.G. Massotti, L. and Lando, M. (2003), "Implementation of an Adaptive Predictor-Corrector Neural Controller within the NASA IFCS F-15 WVU Simulator", *Proceedings of the American Control Conference*, Denver, Colorado.

Bishop, C. M. (1995) *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford.

Blake, W. (2003), "New simulation tools enable aerial refuelling", *Air Vehicles Directorate news@afrl Magazine.*

Bone, E. and Bolkcom, C. (2003), "Unmannned Aerial Vehicles: Background and Issues for Congress", CRS Web RL31872.

Clark, R. M. Lt. Col. (2000), "Uninhabited Combat Vehicles - Airpower by the People, For the People, But Not with the People", Air University Press.

Dell'Aquila, R.V., Campa, G., Napolitano, M.R., and Mammarella, M. (2007), "Real-time Machine-Vision-Based Position Sensing System for UAV Aerial Refueling" *Journal of Real-Time Image Processing*, Vol. 1, No. 3, pp 213–224.

Dogan, A. Sato, S. and Blake, W. (2005), "Flight Control and Simulation for Aerial Refuelling", *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, California.

Doitsidis, L. Valavanis, K.P. Tsourveloudis, N.C. and Kontis, M. (2004), "A Framework for Fuzzy Based UAV Navigation and Control", *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, New Orleans.

Dryden, J. B. (1986), "F-16 Aerodynamics", *Code One Magazine*, Lockheed Martin Aerodynamics Company.

Dufrene, W.R. Jnr. (2004), "AI Techniques in Uninhabited Aerial Vehicle Flight", *IEEE A&E Systems Magazine*, August, pp 8 – 12.

Etkin, B. and Reid, L. D. (1996), *Dynamics of Flight Stability and Control*, John Wiley and Sons Inc., New York.

Eshel, T. (2004), *Extending the F-16 Range*, Defense Update - International Online Defense Magazine, Issue 1, http://www.defense-update.com/products/f/f-16-fuel.htm, Last accessed: 18 September 2006.

Fravolini, M. L. Ficola A, Napolitano M. R. Campa G. and Perhinschi G. (2003), Development of Modelling and Control Tools for Aerial Refuelling for UAVs, *Proceedings of AIAA Guidance, Navigation and Control Conference and Exhibit*, Texas.

Gili, A. P. and Battipede, M. (2001), "Adaptive Neurocontroller for a Nonlinear Combat Aircraft Model", *Journal of Guidance, Control and Dynamics*, Vol.24, No. 5, September-October, pp. 910-917.

Hansen, J.L., Murray, J.E., and Campos, N.V. (2004), "The NASA Dryden AAR Project: A Flight Test Approach to an Aerial Refueling System", *Proceedings of the AIAA Atmospheric Flight Mechanics Conference*, 16-19 Aug, Providence.

Hou, Y. (2003), "Model of F-16 Fighter Aircraft - Equations of Motion", University of Southern California, Department of Electrical Engineering - Systems, http://www.usc.edu/dept/ee/catt, Last accessed: 18 September 2006.

Hunt, K.J. Sbarbaro, D. Zbikowski, R. and Gawthrop, P.J. (1992) "Neural Networks for Control Systems - A Survey", *Automatica*, Vol. 28, No. 6, pp. 1083–1112.

Ito, D. J. Georgie, J. Valasek, and Ward D.T. (2002), "Reentry Vehicle Flight Controls Design Guidelines: Dynamic Inversion", *NASA/TP 2002-210771*, Flight Simulation Laboratory.

Jin, Z., Shima, T., and Schumacher, C.J. (2006), "Optimal Scheduling for Refuelling Multiple Autonomous Aerial Vehicles", *IEEE Transactions on Robots*, Vol 22, No. 4, pp 682 – 693.

Lane, S.H. and Stengel, R.F. (1988), "Flight Control Design using Non-linear Inverse Dynamics", *Automatica*, Vol 24, No 4, pp471 – 483.

Kaneshige, J. Bull J. and Totah J.J. (2001), "Generic Flight Control and Autopilot System", AIAA-2000-4281.

Koch, A.(2005), "How to Aerial Refuel the F-16",
Available at: http://www.virtualtigers.com/htm/refuel.htm, Last accessed: 18 September 2006.

Li, Y. Sundararajan, N. and Saratchandran. P. (2001), "Neuro-controller Design for Nonlinear Fighter Aircraft Manoeuvre using Fully Tuned RBF Networks", *Automatica*, Vol. 37, pp 1293–1301.

Mammarella, M., Campa, G., Napolitano, M.R., and Fravolini, M.L. (2008) "Comparison of Point Matching Algorithms for the UAV Aerial Refueling Problem" Submitted to: *Machine Vision and Applications*, November 2007, Accepted March 2008. Final proof copy submitted June 2008.

McFarland M.B., and Calise A.J. (2000), "Adaptive Nonlinear Control of Agile Antiair Missiles Using Neural Neyworks", *IEEE Transactions on Control Systems Technology*, Vol 8, No 5, pp 749 - 756

Meyer, N. (2005), Control and Simulation of Arbitrary Flight-Trajectory using Neural Network Techniques, *University of the Witwatersrand*, Final Year Research Project MECN 446.

Mitchell, D. (2003), "Australian Contribution to UAVs", *Australian Government Department of Industry, Tourism & Resources Flier*.

Morelli, E.A. (1998), "Global Nonlinear Parametric Modeling with Application to F-16 Aerodynamics", *Dynamics and Control Branch, NASA Langley Research Centre*, Hampton, Virginia.

Nabney, I.T. (2002), *NETLAB: Algorithms for Pattern Recognition*, Springer, London.

Nikolos, I.K. Doitsidis, L. Christopoulos, V.N. and Tsourveloudis, N. (2003), "Roll Control of Unmanned Aerial Vehicles using Fuzzy Logic", *WSEAS Transaction on Systems*, Issue 4, Vol. 2, pp 1039-1047.

Ochi, Y. and Kominami, T. (2005), "Flight Control for Automatic Aerial Refuelling via PNG and LOS Angle Control", *AIAA Guidance, Navigation and Control Conference and Exhibit,* California.

Ollero, A., and Merino, L. (2004), "Control and perception techniques for aerial robots", *Annual reviews in control*, Vol 28, pp 167 – 178.

Pardesi, M.S. (2005), "Unmanned Aerial Vehicles/Unmanned Combat Aerial Vehicles – Likely Missions and Challenges for the Policy-Relevant Future", *Air and Space Power Journal*, Fall Edition, Available at:
http://www.airpower.maxwell.af.mil/airchronicles/apj/apj05/fal05/pardesi.html

Pashikar, A.A. Sundararajan, N. and Saratchandran, P.(2007), "Adaptive Nonlinear Neural Controller for Aircraft Under Actuator Failures", *Journal of Guidance, Control, and Dynamics*, 30(3), pp 835-847.

Pedro J.O. (1992), "Numerical Simulations of Transport Aircraft in Variable Wind Field", *Faculty of Mechanical, Power and Aeronautical Engineering, Warsaw University of Technology*, Warsaw, Poland, PhD Thesis (unpublished).

Pedro, J.O. and Bigg, C.G. (2005), "Development of a Flexible Embedded Aircraft/Control System Simulation Facility", *Proceeding of the AIAA Modeling and Simulation Technologies, Conference and Exhibit*, San Francisco, 2005.

Pollini, L. Campa, G. Giulietti, F. and Innocenti, M. (2003), "Virtual Simulation Set-Up for UAVs Aerial Refuelling", *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, Texas.

Reiner, J., Balas, G.J., and Garrard, W.L. (1996), "Flight Control Design Using Robust Dynamic Inversion and Time Scale Separation", *Automatica*, Vol 32, No 11, pp 1493 – 1504.

Sanders, R.(2002), "UAVs – An Israeli Military Innovation", *JFQ Magazine*, Winter 2002 – 03

Skogestad, S., and Postlethwaite, I. (1996), *Multivariable Feedback Control*, John Wiley & Sons Ltd, England.

Soares, F., Burken, J., and Marwala, T. (2006), "Neural Network Applications in Advanced Aircraft Flight Control System, a Hybrid System, a Flight Test Demonstration", *ICONIP*, Part III, LNCS 4234, pp. 684–691.

Spaulding, C.M. Mansur, H.M. Tischler, M.B. Hess, R.A. and Franklin J.A., (2005), "Nonlinear Inversion Control for a Ducted Fan UAV", *Proceedings of the AIAA Atmospheric Flight Mechanics Conference and Exhibit*, San Francisco, California.

Steinberg, M. L. (2001), "Comparison of Intelligent, Adaptive, and Nonlinear Flight Control Laws", *Journal of Guidance, Control and Dynamics*, Vol. 24, No. 4, July-Aug., pp. 693–699.

Stengel, R.F. (2004), *Flight Dynamics*, Princeton University Press, New Jersey.

Stevens, B.L. and Lewis F.L. (1992), *Aircraft Control and Simulation*, John Wiley and Sons Inc., New York.

Suresh, S. Omkar, S.N. and Mani, V. (2006), "Direct Adaptive Neural Flight Controller for F-8Fighter Aircraft", *Journal of Guidance, Control and Dynamics*, Vol. 29, No. 2, March-April, pp. 454–464.

Thampi, K.G. Principe, J. Motter, M.A. Cho J.H., and Lan J. (2002), "Multiple Model Based Flight Control Design"*, Midwest Symposium on Circuits and Systems*, Tulsa, Oklahoma.

Thompson K.E. Major (1998), "F-16 Uninhabited Air Combat Vehicles", *Air Command and Staff College University*.

Vendra S., Campa G., Napolitano M.R., Mammarella M., Fravolini M.L., and Perhinschi M. (2007), "Addressing corner detection issues for machine vision based UAV aerial refuelling", *Machine Vision and Applications*, Volume 18, No. 5, pp. 261 – 273.

Withrow M. (2004), "UAV automated aerial refuelling one step closer to reality", *Air Vehicles Directorate news@afrl Magazine*.

Wilson J.R. (2005), "UAV Worldwide Roundup – 2005", *Aerospace America*, American Institute of Aeronautics and Astronautics, September, pp. 26-31.

Won, T.H. Song, D.S. Nam, S.K. and Lee, M.H. (1999), "A Design for Fuzzy Control for High Performance Aircraft", *ISIE*, Slovenia.

Wong, K.C. and Bill, C. (1998), "UAVs over Australia - Market and Capabilities", Aerospace Technology Forum Report.

Zhu Z.H., and Meguid S.A. (2007), "Modeling and Simulation of Aerial Refueling by Finite Element Method", International Journal of Solids and Structures, Vol 44, pp 8057–8073.