

ÉCOLE DE TECHNOLOGIE SUPÉRIEURE
UNIVERSITÉ DU QUÉBEC

THESIS SUBMITTED TO
ÉCOLE DE TECHNOLOGIE SUPÉRIEURE

IN PARTIAL FULFILLMENT
OF THE REQUIRMENTS OF THE DEGREE OF
MASTER OF ELECTRICAL ENGINEERING
M. Eng.

BY
MOUSSADDY, Abdel-Rahman

A THRUST CONTROLLER FOR THE DYNAMIC TEST BED OF THE FLIGHT
MANAGEMENT SYSTEM

MONTRÉAL, APRIL 27 2009

THIS THESIS HAS BEEN EVALUATED
BY THE FOLLOWING BOARD OF EXAMINERS:

Mrs. Ouassima Akhrif, Thesis Supervisor
Département de génie électrique à l'École de technologie supérieure

Mr. Maarouf Saad, Thesis Co-supervisor
Département de génie électrique à l'École de technologie supérieure

Mr. Louis-A. Dessaint, President of the Board of Examiners
Département de génie électrique à l'École de technologie supérieure

Mr. Stéphane Roy, Industrial Representative
CMC Electronics

THIS THESIS HAS BEEN PRESENTED AND DEFENDED
BEFORE A BOARD OF EXAMINERS AND PUBLIC
MARCH 27, 2009
AT ECOLE DE TECHNOLOGIE SUPERIEURE

ACKNOWLEDGEMENTS

With gratitude I wish to express my appreciation and thank my supervisors Dr. Ouassima Akhrif, and Dr. Maarouf Saad, who welcomed me into their team at Ecole de technologie supérieure (ETS) and guided me throughout my studies. Without their help and support it would have been impossible to bring this project and study to term.

I would also like to thank CMC Electronics for supporting this project in every aspect throughout the process and giving me the chance to couple academic research to industrial integration. Stephane Roy, Engineer at CMC Electronics, has particularly made this project feasible with his constant support, availability and guiding.

Engenuity Technologies has been of great help too, providing the simulation tools and software platform necessary for the development and integration of the project.

I also thank the Consortium of Research and Innovation in Aerospace in Quebec (CRIAQ) for their constant involvement in aerospace research, their administrative coordination of the project between the different industrial and academic parties involved, as well as their financial support that made all this possible.

Furthermore, I thank all the Concordia University members working on the FMS Dynamic Test Bed CRIAQ project for their support, cooperation and activities coordination.

Finally, I express my gratitude and thanks to my family whom supported me and stood by me, throughout the good and not so good moments, always challenging me and pushing me towards higher goals and achievements.

A THRUST CONTROLLER FOR THE DYNAMIC TEST BED OF THE FLIGHT MANAGEMENT SYSTEM

MOUSSADDY, Abdel-Rahman

ABSTRACT

The following study presents the Thrust Controller (TC) design and validation for a large civil aircraft, the Boeing 747. The TC works with the Automatic Flight Control System (AFCS) and the Flight Management System (FMS) to provide automatic, full-flight-regime energy management with minimum pilot inputs. The controller is to be coupled to the pitch channel of the AFCS to eliminate any altitude variation during thrust manoeuvres and will be available throughout the entire flight envelope. The proposed system is integrated within the Dynamic Test Bed (DTB) developed by CMC Electronics and its partners. In fact, in an attempt to minimize costs and reduce "in air" testing, CMC wishes to fully develop the concept of a real-time DTB, a simplified aircraft simulator, and to expand it into a comprehensive FMS design and test tool, capable of certified service. The goal of the present study is the design, simulation, integration and qualification of the TC using the DTB as the integration platform.

After a brief review of the command law synthesis methods for aircraft thrust control two concepts were studied. First the thrust controller was developed using a classic single input-single output controller based on the Proportional-Integral-Derivative structure and was validated over the full flight envelope. The controller is coupled to the pitch channel using an altitude variation dependant gain on the speed channel to eliminate altitude variation during speed control manoeuvres using thrust increase and decrease. The second design architecture proposes a modern state-feedback controller that fully couples pitch and speed channels. Feedback gains were chosen for every linearized point using the Eigenvector assignment method for system robustness. A controller was thus designed for every linearized point and the TC system switches between controllers according to flight conditions. Many improvements were then added to both controllers' architectures to add speed and thrust limitations and placards protections prior to nonlinear integration.

The integration of the modern controller could not be validated within the nonlinear DTB framework in the scope of the current project due to autopilot and platform limitations. On the other hand, the classical controller proposed is integrated and tested within the nonlinear conditions of the DTB.

To qualify the TC as a design and test tool and integrate it in the DTB we sought to ensure its smooth functioning and to reach the performance awaited while respecting the various limits and placards. A set of qualification tests was written and run for that purpose. The results were judged satisfactory as the aircraft settles within the required limitations while improvements can be made and recommendations are presented.

Keywords : control, command, speed, pitch, state-feedback

UN SYSTÈME DE CONTRÔLE DE LA POUSSÉE POUR LE BANC D'ESSAI DYNAMIQUE DU SYSTÈME DE GESTION DE VOL

MOUSSADDY, Abdel-Rahman

RÉSUMÉ

Nous présentons dans cette étude le système de contrôle de la poussée d'un avion civil, le Boeing 747. Le système proposé, couplé à l'autopilote et le système de gestion de vol, assure un contrôle automatique de la poussée avec un minimum d'interaction par le pilote.

Dans le but de minimiser les frais de développement CMC développe un banc d'essai pour leur famille de systèmes de gestion de vol (FMS). Ce banc d'essai est utilisé comme outil de test, certification et démonstration. Le but du présent projet est de développer le système de contrôle de la poussée et de l'intégrer au banc d'essai, tout en le qualifiant comme outil de test faisant partie du banc. Afin de pouvoir qualifier le banc d'essai comme outil de tests, plusieurs étapes rigoureuses doivent être considérées durant le développement des composantes du banc d'essai, dont le contrôleur de poussée fait parti. Parmi ces étapes nous retrouvons l'identification des requis, la proposition du design qui satisfait ces requis, et finalement les tests de validation du système selon les exigences initiales, le tout dans un environnement contrôlé.

La tâche primaire du contrôleur étant d'assurer une commande de poussée en utilisant la manette de gaz, il est nécessaire que le contrôleur permette de poursuivre une commande de poussée et ce, sans dépassement ni erreur en régime permanent. Puisque la poussée n'est pas une entité mesurable directement, nous utilisons un ratio entre la pression à l'entrée et la pression à la sortie des turbines (Engine Pressure Ratio -EPR-) comme mesure de la force appliquée. De plus, couplé à l'autopilote, le contrôleur doit pouvoir poursuivre une commande de vitesse à l'aide du contrôle de la poussée, et ce sans affecter l'altitude de l'avion. Or, une variation de vitesse entraîne nécessairement une variation de portance et donc une variation d'altitude. Il est donc nécessaire de coupler le contrôle de la poussée au contrôle de tangage de l'autopilote pour minimiser les variations d'altitudes durant les variations de poussée. Afin d'assurer une sécurité accrue, les limites de vitesses, de poussée ainsi que d'accélération doivent être respecté. Finalement le contrôleur doit être valide sur l'enveloppe de vol entière. L'intégration du système de commande est faite sur le banc de test dynamique développé par CMC Électronique et ses partenaires. Cette intégration ajoute donc un requis, soit la nécessité que le système soit compatible avec la plateforme et qu'il soit développée en langage C.

Deux architectures de contrôles sont étudiées. Premièrement le système est développé en utilisant un contrôleur Proportionnel-Intégral-Dérivée (PID) classique et validée sur l'enveloppe de vol entière. Le contrôleur utilise deux boucles de contrôle PID indépendante, une pour le contrôle de la poussée et une pour le contrôle de la vitesse. La sélection entre les différentes boucles de contrôle est faite selon le mode d'autopilote choisi. En mode montée ou descente, comme au décollage, le contrôleur commande la poussée pour assurer une variation d'altitude pendant que l'autopilote contrôle la vitesse en utilisant les élévateurs. Une fois que l'altitude désirée est atteinte, le contrôleur de poussée à le mandat de contrôler

la vitesse tandis que l'autopilote, toujours en utilisant une commande de tangage sur l'angle des élévateurs, assure le maintien de l'altitude. Durant la transition entre ces deux phases de vols d'importantes variations de poussée sont observées. Cette variation cause une variation d'altitude, et une importante accélération, qui est ensuite annulée par la commande de tangage. Pour pouvoir assurer une transition douce sans variation d'altitude la commande de poussée et la commande de tangage sont couplées. Nous proposons une commande de poussée proportionnelle à la variation d'altitude pour assurer une transition lisse. Le gain dans la boucle de contrôle de vitesse est donc proportionnel à la variation d'altitude.

La deuxième architecture propose un contrôleur moderne à retour d'état qui couple vitesse et tangage. Cette architecture n'utilise pas de boucles indépendantes et donc permet d'assurer un contrôle en utilisant les élévateurs et la poussée conjointement pour atteindre une altitude et vitesse donnée. Cela permet d'éviter les variations d'altitude dues aux variations de vitesse ainsi que les variations de vitesse dues aux variations d'angle de tangage. Les gains de retour d'états sont calculés par placement de vecteurs propres pour assurer la robustesse du système. Un contrôleur distinct a été validé pour différentes positions de l'enveloppe de vol et le système de contrôle sélectionne les gains du contrôleur selon la zone de l'enveloppe de vol courante.

Plusieurs autres améliorations ont été apportées aux deux architectures pour assurer la protection des limites dynamiques de vitesses et poussées.

L'intégration du contrôleur à retour d'état, système de commande moderne, au banc d'essai fut impossible du aux limitations de l'autopilote et de la plateforme de développement. Cette intégration nécessite une action directe sur la manette des gaz et sur les élévateurs. Cependant, une loi de commande des élévateurs interne est incorporée dans la plateforme et les élévateurs sont contrôlés via l'angle de tangage seulement. Pour pouvoir accéder directement à l'angle des élévateurs une modification à la plateforme doit être effectué ce qui tombe malheureusement à l'extérieur des limites de cette étude. Cependant le contrôleur classique fut intégré et testé. Pour qualifier le contrôleur comme outil de test et développement intégré sur le banc d'essai il a fallu démontrer le respect des exigences et l'atteinte de la performance désirée. Plusieurs tests ont été écrits à cette fin et les résultats ont été concluants. Le contrôleur assure une poursuite de poussée et de vitesse sans dépassement ni erreur en régime permanent. Il permet aussi de poursuivre une vitesse sans affecter l'altitude. Les contraintes de vitesse, de poussée et d'accélération sont de plus respectées. Plusieurs améliorations possibles sont aussi présentées pour améliorer le respect des limites de vitesses durant les manœuvres de poursuite de poussée.

Mots-clés : contrôle, commande, poussée, vitesse, tangage, retour d'état

TABLE OF CONTENTS

	Page
INTRODUCTION	1
CHAPITRE 1 THRUST CONTROL: A LITERATURE REVIEW	4
1.1 Introduction.....	4
1.2 Autothrottle modes.....	5
1.2.1 Takeoff.....	8
1.2.2 Climb.....	8
1.2.3 Cruise.....	9
1.2.4 Descent.....	9
1.3 Control Laws.....	10
1.3.1 EPR Control Law.....	10
1.3.2 Speed Control Law.....	13
1.3.3 Thrust control with pitch coupling.....	16
1.4 Modern integrated control laws: MIMO Systems	17
1.4.1 The total energy control system.....	18
1.4.2 Longitudinal law using the eigenvector method.....	19
1.5 Entire flight envelope validation.....	21
CHAPITRE 2 SYSTEM REQUIREMENTS	24
2.1 Introduction.....	24
2.2 Thrust capture	25
2.3 Speed capture.....	25
2.4 EPR constraints and engine limits	27
2.5 Speed and acceleration limits and placards	27
2.6 Flight envelope validity	28
2.7 Pitch coupling	28
CHAPITRE 3 AIRCRAFT DYNAMICS AND MODELISATION.....	29
3.1 Basic flight principles	29
3.1.1 Forces of flight.....	29
3.1.2 Control surfaces	31
3.1.3 Coordinate systems	32
3.2 Equations of motion.....	37
3.2.1 Translational motion.....	38
3.2.2 Rotational motion.....	39
3.2.3 Contribution of the forces due to gravity.....	42
3.2.4 Linearization of the Inertial and gravitational terms.....	42
3.3 Linearization of the longitudinal equations of motion.....	45
3.4 Longitudinal state equations	47
3.5 Longitudinal stability.....	50
3.6 Linear state space model.....	51

3.7	Open loop characteristics.....	52
3.7.1	Eigenvalues of the longitudinal movement.....	53
3.7.2	Effect of speed and altitude variation on dynamic stability.....	56
3.7.3	Thrust and pitch coupling at trim conditions.....	61
3.8	Engine modelization.....	64
3.8.1	Turbofan multi-spool engines.....	65
3.8.2	Engine Model.....	66
3.8.3	Engine Pressure Ratio.....	67
CHAPITRE 4 CLASSIC CONTROL DESIGN.....		72
4.1	PID control.....	72
4.1.1	Loop gain tuning.....	74
4.2	Thrust control.....	76
4.3	Flight level change.....	80
4.4	Speed control.....	82
CHAPITRE 5 MODERN CONTROL DESIGN.....		83
5.1	Pole placement and Eigenstructure method.....	83
5.1.1	Pole assignment.....	84
5.1.2	Robust eigenstructure assignment.....	85
5.2	Controller design.....	87
5.2.1	Controllability.....	89
5.2.2	State feedback.....	90
5.2.3	Effect of various gains on feedback.....	94
5.2.4	Altitude hold.....	97
5.2.5	Simulation results.....	99
5.3	Gain scheduling.....	106
CHAPITRE 6 CRITICAL LIMITS AND PROTECTIONS.....		108
6.1	EPR and engines limit protection.....	108
6.2	Throttle position and rate limits.....	111
6.3	Speed limits protection.....	112
CHAPITRE 7 INTEGRATION AND SYSTEM INTERFACE.....		115
7.1	The Dynamic Test Bed.....	115
7.1.1	System Overview.....	115
7.1.2	Dynamic Test Bed Overview.....	115
7.2	Autothrottle and autopilot interface.....	120
7.3	System inputs.....	121
7.3.1	Flightsim systems.....	121
7.3.2	PTT (FMS).....	123
7.3.3	GUI.....	123
7.3.4	Additional Inputs.....	124

7.4	System outputs	125
7.4.1	Outputs to Flightsim	125
7.4.2	GUI	125
7.5	System code and development	126
7.6	Integration issues & limitations	127
7.6.1	Pitch inner loop	128
7.6.2	PID controller integral term	129
CHAPTER 8 SYSTEM TEST AND VALIDATION		130
8.1	Qualification standards and guidelines	130
8.1.1	The need for Tool Qualification	130
8.1.2	Qualification Requirements	131
8.1.3	Software Integration Tests	132
8.1.4	Requirements Tests	132
8.1.5	Code Testing	133
8.1.6	Documentation	133
8.2	TC Model Functionality Test Procedures	133
8.2.1	Thrust Idle	134
8.2.2	EPR Arm and Reference Limit Protection and EPR Capture	135
8.2.3	EPR Flight Level Change Climb	136
8.2.4	EPR Flight Level Change Descent	138
8.2.5	EPR Capture Speed Limits Protection	141
8.2.6	Low Altitude Speed Limit Protection	146
8.2.7	Speed Cruise IAS Capture	148
8.2.8	Speed Cruise MACH Capture	151
8.2.9	Speed Capture Mode EPR Limits Protection	154
CONCLUSION		157
RECOMMENDATIONS		159
ANNEX I LINEARISATION RESULTS		162
ANNEX II SOFTWARE/SYSTEM VALIDATION & TEST PROCEDURES		180
REFERENCES		225

LIST OF TABLES

	Page
Table 1.1	Basic autothrottle functions over the flight profile7
Table 3.1	System Eigenvalues53
Table 3.2	System Eigenvalues and Parameters.....54
Table 4.1	Effects of PID parameters variation.....75
Table 4.2	Gain values for EPR Control law.....77
Table 5.1	Gain values over flight envelope for Speed Control law.....106
Table 5.2	Non varying gain values for Speed Control law107
Table 7.1	Thrust Controller Input Variables122
Table 7.2	Thrust Controller Output Variables125
Table 7.3	GUI Output messages126

LIST OF FIGURES

	Page
Figure 1.1	Simplified schematic of an autoflight system.....4
Figure 1.2	Basic autothrottle functions over the flight profile7
Figure 1.3	EPR-Throttle position curve, Robbins & Simpson (1991) 11
Figure 1.4	EPR Control Block Diagram, Bader (1990)12
Figure 1.5	EPR Control Block Diagram for Takeoff phase, Bader (1990).....13
Figure 1.6	Speed control simplified block diagram, Collinson (1996).....13
Figure 1.7	IAS Control Block Diagram, Bader (1990)15
Figure 1.8	Total Energy Control System Block Diagram, Lambregts (2000)18
Figure 1.9	Block diagram of a typical state feedback controller.....20
Figure 1.10	Flight envelope for various weight values, Derbois (2002).....22
Figure 3.1	Forces of flight.....29
Figure 3.2	Basic Aircraft Control Surfaces31
Figure 3.3	ECI and ECEF frame33
Figure 3.4	Earth frame.....34
Figure 3.5	Body frame.....35
Figure 3.6	System root map55
Figure 3.7	Open loop time response.....56
Figure 3.8	Effect of speed variation on eigenvalues57
Figure 3.9	Effect of speed variation on phugoid mode eigenvalues58
Figure 3.10	Effect of altitude variation on eigenvalues59
Figure 3.11	Effect of altitude variation on phugoid mode eigenvalues60
Figure 3.12	Turbofan multi-spool engine.....65

Figure 3.13	EPR-Throttle position relation.....	71
Figure 4.1	PID controller.....	73
Figure 4.2	EPR Capture control channel.....	76
Figure 4.3	EPR controller Simulink simulation model.....	78
Figure 4.4	EPR controller outputs and EPR capture response.....	79
Figure 4.5	Controller architecture in Flight level change mode.....	81
Figure 4.6	Speed Controller architecture.....	82
Figure 5.1	State feedback blocks.....	88
Figure 5.2	System response for initial conditions $u=0$ m/sec and $\theta=0.5$ rad.....	91
Figure 5.3	Command plot for initial conditions $u=0$ m/sec and $\theta=0.5$ rad.....	91
Figure 5.4	System response for initial conditions $u=0.5$ m/sec and $\theta=0$ rad.....	92
Figure 5.5	Command plot for initial conditions $u=0.5$ m/sec and $\theta=0$ rad.....	93
Figure 5.6	Closed loop step response of speed and pitch, with state feedback matrix K.....	94
Figure 5.7	Closed loop step response with state feedback gain matrix K_m	96
Figure 5.8	PID controller.....	97
Figure 5.9	Speed Capture control law with altitude hold.....	98
Figure 5.10	Pitch-Speed state-feedback controller architecture.....	99
Figure 5.11	Speed and altitude response.....	100
Figure 5.12	Elevator command.....	101
Figure 5.13	Throttle command.....	102
Figure 5.14	Aircraft model states response.....	103
Figure 5.15	Altitude and climb rate response.....	104

Figure 5.16	G-load on aircraft body axes.....	105
Figure 6.1	Controller with EPR limit protection.....	109
Figure 6.2	EPR and Speed responses with no EPR limit saturation	110
Figure 6.3	EPR and Speed responses with EPR limit saturation	111
Figure 6.4	Forces applied to aircraft	112
Figure 7.1	High Level H/W Block Diagram	116
Figure 7.2	S/W Block Diagram.....	117
Figure 7.3	Flightsim Simulation Diagram.....	119
Figure 7.4	CMC AFCS GUI.....	124
Figure 8.1	Thrust variation for a Thrust idle command	134
Figure 8.2	EPR capture during an EPR thrust command.....	135
Figure 8.3	Altitude capture during climb.....	136
Figure 8.4	EPR variation during climb	137
Figure 8.5	Airspeed variation during climb	138
Figure 8.6	Altitude capture during descent	139
Figure 8.7	EPR variation during descent.....	140
Figure 8.8	Airspeed variation during descent	141
Figure 8.9	EPR variation during speed limit protection control	142
Figure 8.10	Speed variation with speed limit protection.....	143
Figure 8.11	Mach number variation with speed limit protection.....	144
Figure 8.12	Altitude variation during speed limit protection.....	145
Figure 8.13	Accelerations variation during speed limit protection.....	145
Figure 8.14	Speed variation during low altitude speed limit protection	146

Figure 8.15	Altitude variation during low altitude speed limit protection.....	147
Figure 8.16	Acceleration variation during low altitude speed limit protection.....	147
Figure 8.17	Airspeed decrease capture.....	148
Figure 8.18	Airspeed increase capture	149
Figure 8.19	Altitude variation during speed capture	150
Figure 8.20	Acceleration variation during speed capture.....	150
Figure 8.21	Mach speed capture.....	152
Figure 8.22	Altitude variation during Mach capture	153
Figure 8.23	Acceleration variation during Mach capture.....	153
Figure 8.24	Airspeed capture	154
Figure 8.25	EPR variation with EPR limits protection during airspeed capture.....	155
Figure 8.26	Altitude variation during speed capture with EPR limits protection	156
Figure 8.27	Acceleration variation during speed capture with EPR limits protection.....	156

LIST OF ABBREVIATIONS AND ACRONYMS

AFCS	Automatic Flight Computer System
ARINC	Air Radio Inc.
CDU	Control Display Unit
CMC	Canadian Marconi Company
CRIAQ	Consortium for Research and Innovation in Aerospace in Quebec
CRZ	Cruise mode
DoT	US Department of Transportation
DTB	Dynamic Test Bed
ECEF	Earth Centered Earth Fixed frame
ECI	Earth Centered Inertial frame
EPR	Engine Pressure Ratio
FAA	US Federal Aviation Administration
FLCH	Flight Level Change Mode
FLSIM	Engenuity technologies Flightsim Software
FMS	Flight Management System
FMSP	Flight Mode Selector Panel
GUI	Graphic User Interface
H/W	Hardware
IAS	Indicated Airspeed
IOS	Instructor Operation Station
MCDU	Multi-Purpose Control and Display Unit

MIMO	Multi-Input Multi-Output
MMO	Mach Maximum Operating Speed
MNE	Mach Never Exceed speed
N/A	Not Available
NED	North-East-Down frame
PC	Personal Computer
PID	Proportional-Integral-Derivative controller
PTT	Part Task Trainer
PT	Total Pressure
SISO	Single-Input Single-Output
SPD	Speed mode
S/W	Software
TAS	True Airspeed
TC	Thrust Control
TCCA	Transport Canada Civil Aviation
TCP/IP	Transmission Control Protocol
TECS	Total Energy Control System
TLA	Thrust Lever Angle
UDP/IP	User Datagram Protocol
VMO	IAS Maximum Operating Speed
VNAV	Vertical Navigation mode

LIST OF SYMBOLS AND MEASURE UNITS

SYMBOLS

α	Angle of attack
α_T	Engine's angle of attack
ε	Angle between the engines axis and the aircraft chord
β	Side-slip angle
γ	Path angle
θ	Pitch angle
φ	Bank angle
λ	Eigenvalue
Ψ	Euler Yaw angle
Θ	Euler Pitch angle
Φ	Euler Roll angle
δ_e	Elevators deviation angle
δ	Thrust lever position
ρ	Air density
\bar{x}	State vector
\bar{u}	Control vector
ζ	System damping
ω	Natural frequency
t_{half}	Time to half amplitude
T	Thrust
W	Weight
D	Drag
L	Lift
C_D	Drag coefficient
\vec{F}	Sum of all externally applied forces
\vec{F}_0	Equilibrium component
$\vec{\omega}$	Angular velocity
\vec{M}	Sum of all applied torques
\vec{H}	Angular momentum
\vec{I}	Inertia matrix
\vec{g}	Gravitational force
$a_{x_{cg}}$	Acceleration along the Body frame x axis

$a_{y_{cg}}$	Acceleration along the Body frame y axis
$a_{z_{cg}}$	Acceleration along the Body frame z axis
\vec{V}_T	Aircraft airspeed vector
U_T	Aircraft airspeed along the Body frame x axis
V_T	Aircraft airspeed along the Body frame y axis
W_T	Aircraft airspeed along the Body frame z axis
m	Mass
h	Altitude
U	Longitudinal speed, along X body axis
V	Lateral speed, along Y body axis
W	Vertical speed, along Z body axis
P	Roll angular velocity, around the body X axis
Q	Pitch angular velocity, around the Y body axis
R	Yaw angular velocity, around the Z body axis
X	Force applied to the aircraft along the X axis
Y	Force applied to the aircraft along the Y axis
Z	Force applied to the aircraft along the Z axis
L	Roll moment
M	Pitch moment
N	Yaw moment
$N1$	Engine high speed spool rotational speed
$N2$	Engine low speed spool rotational speed
η	Normalized throttle position

MEASURE UNITS

m	meters
sec	seconds
rad	radians
lbs	pounds
ft	feet
m/sec	meters per second
deg/sec	degrees per second
rad/sec	radians per second

INTRODUCTION

During the design of any new avionic or aircraft control system many challenges arise and many requirements have to be taken into account. Any new avionic development and design should show low development, implementation and certification costs. Furthermore, a new airplane control system must demonstrate more than just good dynamic response to be seriously considered as a replacement for existing systems. In the last couple of decades the dramatic increase in the price of oil, cost of manpower, and amount of air traffic have meant additional constraints on developing a new system. The result of these constraints is that a new system must be designed for:

1. Fuel economy and efficiency
2. Maximum safety
3. Minimum development, implementation and certification costs

A major part of the costs attached to the design of any avionic system are those in regards to the system testing and certification. During system validation, many hours are spent in flight, requiring high expenses in fuel and aircraft use not to mention the cost of manpower. While these tests are necessary to prove the efficiency of the developed system, one has to think of other venues that are more affordable, environment friendly and cost effective. In this regard, an attempt was made to rethink the development, implementation and certification procedures by the CRIAQ (Consortium for Research and Innovation in Aerospace in Quebec), with the collaboration of its partners CMC Electronics, Engenuity Technologies, École de technologie Supérieure and Concordia University. The main objective of the research project proposed by the CRIAQ is to fully develop the concept of a real-time Dynamic Test Bed (DTB) and to expand the existing system prototype into a comprehensive Flight Management System (FMS) design and test tool, capable of certified service. The test bed, which includes a control station or an operator/instructor station, may create artificial failure conditions in the different systems (engine fire, landing gear malfunction, electrical problems, and navigation system problems) and allows the pilots and other crew members to

practice their response to these situations and measure their level of proficiency. This technology is developed in parallel for airplanes and helicopters.

Principal Benefits of the research project to Industry

- Provide a more systematic, less error-prone and more efficient design and development environment for the FMS product line of CMC.
- Facilitate a comprehensive testing in a controlled laboratory setting, while substantially reducing the prohibitively expensive flight testing in actual aircraft.
- Use the DTB in lieu of actual flight testing to the maximum extent possible, with accreditation from the relevant aviation regulatory bodies, such as Transport Canada Civil Aviation (TCCA), U.S. Department of Transportation (DoT), Federal Aviation Administration (FAA) and others.

Therefore, the DTB serves as a development and testing tool in which the FMS can be integrated and tested. Many efforts are being put into the development of the DTB in hope of certifying the DTB as a test tool, and hence the possibility to later certify the new line of FMS products, or any other avionic system for that matter, on ground and thus minimizing the in air certification costs.

The Dynamic Test Bed is effectively a simplified aircraft simulator, which recreates the signals for an avionic system such as the FMS. The development of the DTB leads to several other needs. The DTB includes a flight model, environment model, various avionics models, aircraft control models, etc. The simulation, fidelity and integration of such models are critical to the development and well functioning of the DTB. The FMS, which is at the heart of the developed DTB, is usually coupled with many control and navigation systems.

The present study presents the design, development, simulation as well as the integration within the DTB of one of these critical systems: the Thrust control system.

Thrust control systems, or autothrottle systems, are systems that help the pilot control the engine's thrust. These systems are used on jet aircrafts, particularly commercial jet aircraft, to provide automatic control of engine thrust.

The purpose of the Autothrottle is to enhance the ease, economy, and safety of flight operations. The Autothrottle works with the Automatic Flight control System (AFCS) and the Flight Management System (FMS) to provide automatic, full-flight-regime energy management with a minimum of pilot inputs. Several architectures are used in the industry, and several others are proposed in the literature. The goal of the present study is to present a modern control design, capable of coupling pitch and thrust to achieve the desired control.

Throughout this study, we will present the thrust control system, its present design, its performance, its integration and simulation. We will push the study further, comparing design architectures and exploring modern design architectures and their possible advantages. We will develop a modern controller based on the state feedback method using Eigenvector assignment to determine feedback gains. Since the DTB is not only a test tool, but also a dynamic test bed which serves as a design environment, autothrottle architectures will be implemented, compared and tested within the DTB.

We will start by a general introduction to aircraft dynamics and longitudinal modelization, presenting the model of the aircraft on which this study will be carried. We will then present an overview of the autothrottle systems, the used architectures and a general literature review of thrust control. Chapter 3 will present the general requirements of our system, requirements that will be used throughout the design and validation process. In chapter 4, the autothrottle architecture and classical control laws will be studied, we will then carry on with the modern control design in chapter 5. Chapter 6 however will present the security features added to our control system. Chapter 7 will study the integration and interface of the thrust controller within the DTB. Chapter 8 will present the simulation and testing procedures and results.

CHAPITRE 1

THRUST CONTROL: A LITTERATURE REVIEW

1.1 Introduction

Thrust control systems, or autothrottle systems, are systems that help the pilot control the engine's thrust. These systems are used on jet aircraft, particularly commercial jet aircraft, to provide automatic control of engine thrust. Thrust is increased by advancing the throttle and decreased by retarding the throttle. Such a change in thrust is obtained by altering the quantity of the fuel flowing to the engines by means of the throttle actuator. The flow of fuel to the engines of the aircraft is metered by the engine control in response to the throttle inputs.

The purpose of the Autothrottle is to enhance the ease, economy, and safety of flight operations. As shown in Figure 1.1, the Autothrottle works with the Automatic Flight control System (AFCS) and the Flight Management System (FMS) to provide automatic, full-flight-regime energy management with a minimum of pilot inputs.

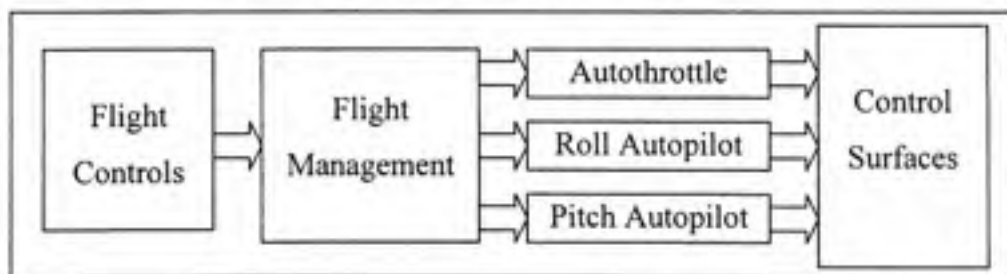


Figure 1.1 Simplified schematic of an autoflight system.

Flight economy is improved by accurate speed control and thrust management, while safety is enhanced by protecting maximum placards airspeeds, observing engine limits, and by reducing pilot workload. Further, aircraft parameters optimized by the FMS can be coupled

to the Autothrottle for effortless tracking of commands. Of course, the Autothrottle can always be set to maintain manually-selected values.

Aircrafts having both, automatic flight control systems and autothrottle systems, are referred to as fully coupled, whether the systems work independently or in a complimentary mode, as in performance management systems for controlling the vertical path of the aircraft or flight management systems for controlling both the vertical and horizontal path of the aircraft.

In view of the substantial increases in fuel costs, aircraft operators are very desirous of increasing fuel efficiencies throughout the entire vertical flight profile of their flight plans by assuring the most cost-effective operations possible. A particular objective of such operations is to obtain smooth, stable and accurate airspeed control during cruise and when the automatic flight control system commands the aircraft to accelerate, decelerate or change its flight path in the vertical plane. Further, during the capture of a cruise altitude, it is desirable, if not mandatory, to maintain the aircraft's speed within predefined limits. Once the aircraft has reached the desired altitude, the engine thrust must be increased or reduced to that value which will maintain the desired airspeed.

We will present in this chapter a brief overview of the autothrottle system. Our main concern in this chapter will be the examination of control laws presented in the literature, classical and modern, used in the autothrottle system.

1.2 Autothrottle modes

Autothrottle modes are closely tied to Flight Guidance modes and can also be influenced by the Flight Management System (Navigation and Performance Computers). The control and indication associated with the multiple autopilot and autothrottle modes are provided by a Flight Mode Selector Panel (FMSP) which enables the selection of the principal modes and

also provides information confirming that the various modes are correctly engaged and functioning properly.

Several modes are attributed to the thrust controller. We can observe two types of modes, thrust modes and speed modes. Both these types use the autothrottle to control engine thrust. In thrust control modes, we usually control the engine's pressure ratio (EPR). EPR is a direct measure of the force the engine generates. We can also control the engine's fan speed if desired like in some control systems (low speed spool fan speed, also expressed as N1 and the high speed spool fan speed, also expressed as N2). Generally we have two modes under thrust modes, EPR and IDLE. In EPR mode we want to maintain a desired or actual EPR.

Under the speed modes, we have two different modes, which are MACH and IAS. In MACH mode, we will want to track and maintain a desired MACH number, and in IAS mode, the objective is to track and maintain a desired indicated airspeed.

It is important here to specify that Mach Hold modes, or IAS hold modes, can also be found on the autopilot controls. In this case, the control law is achieved using pitch and not thrust. We will only address in this study the control of speed using thrust. Basically, the Autothrottle sets engine thrust for Takeoff, Flight Level Change (FLCH), and Go-Around modes. At all other times it controls airspeed.

As presented by Bader D.L. (1990), the best way to describe Autothrottle modes is to step through a typical flight and discuss system operation along the way. Figure 1.2 gives an outline of Flight Guidance and Autothrottle modes for each phase of flight.

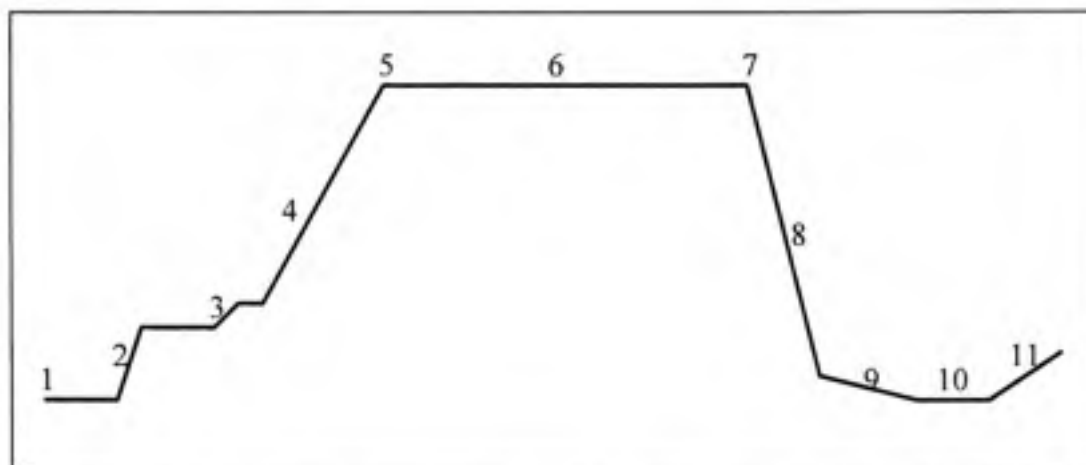


Figure 1.2 Basic autothrottle functions over the flight profile.

Table 1.1

Basic autothrottle functions over the flight profile

	FLIGHT PHASE	FLIGHT DIRECTOR MODE	AUTOTHROTTLE FUNCTION	AUTOPILOT FUNCTION
1	Takeoff	Takeoff	Maximum rated thrust OR manually-set thrust	N/A
2	Climb-out	Flight Level Change	Takeoff to climb thrust transition	Airspeed control
3	Small-step Climb	Flight Level Change	Reduced climb thrust for smooth transition	Airspeed control
4	Large-step climb	Flight Level Change	Full climb thrust	Airspeed control
5	Top-of-climb	Altitude Capture	Transition to airspeed control	Altitude capture
6	Cruise	Altitude Hold	Airspeed control	Altitude hold
7	Top-of-Descent	Flight Level Change	Transition to minimum thrust	Airspeed control
8	Descent	Flight Level Change	Maintain minimum thrust	Airspeed control
9	Approach	Glideslope Track	Airspeed control. Protect flap/gear limits.	Glideslope track
10	Landing	N/A	Disengage	Disengage
11	Go-Around	Go-Around	Maximum rated thrust	Disengage

1.2.1 Takeoff

In a typical takeoff mode, as the aircraft taxis out to the runway for takeoff, the pilot enables the Autothrottle system by pressing the arm button on the Flight Guidance Panel. Takeoff mode is typically selected by pressing the Takeoff/Go-Around button on either throttle. Once the aircraft has been cleared for takeoff, the pilot moves the throttles above a minimum Engine Pressure Ratio (EPR) threshold and engages the Autothrottle. Throttles are then automatically advanced to the required engine power setting for takeoff. On some aircrafts the appropriate EPR is read from a table and entered manually while on recent aircrafts the Autothrottle Computer has data stored internally to determine the appropriate EPR for a full-power takeoff (as a function of altitude and temperature). Ideally, each throttle is driven independently by a separate servo which allows accurate closed-loop EPR control of both engines.

1.2.2 Climb

Once the aircraft is airborne, the pilot normally engages the AFCS (Autopilot) and selects Flight Level Change (FLCH). When FLCH climb is initiated, the Autothrottle smoothly sets thrust to an appropriate level to accomplish the climb; the AFCS controls airspeed through the elevator. In a large-step climb, the Autothrottle sets and maintains thrust at the selected engine rating (usually Maximum Climb EPR). Without an Autothrottle, the pilots would have to constantly monitor engine parameters and make throttle adjustments throughout the climb. In a small-step climb, a unique proportionality scheme provides an appropriate energy level while always ensuring a minimum rate of climb. This reduced-thrust-level feature for small climbs has been popular with the pilots.

Other Flight Guidance modes can be used for climbing. Pitch hold is the default mode for the AFCS. If the Autopilot is engaged without selecting a mode, it maintains current pitch angle

while the Autothrottle controls airspeed (within engine limits). When Vertical Speed mode is selected, the AFCS controls the aircraft rate-of-climb and the Autothrottle maintains selected airspeed.

1.2.3 Cruise

As the airplane nears the selected cruise altitude, the AFCS automatically initiates an Altitude Capture and the Autothrottle smoothly transitions to controlling airspeed. Control laws have to be developed to give a smooth transition with good speed control and without excessive throttle movement. Altitude Hold by elevator control (AFCS) and Speed Hold by throttle control (Autothrottle) are the normal modes of operation in cruise.

The goal in autothrottle control laws is to maximize passenger comfort and minimize unneeded response to temporary environmental variations (e.g., a gust of wind). This is especially true in high altitude cruise where gradual throttle response is appropriate for controlling speed, and large engine excursions are not desirable.

1.2.4 Descent

To descend, the pilot may select Flight Level Change (FLCH) mode. When FLCH is selected, the Autothrottle smoothly reduces the throttle setting to a minimum thrust level to allow the aircraft to achieve a reasonable rate of descent. The AFCS controls airspeed through the elevator, as in FLCH climb. Other potential Flight Guidance modes for descent include Vertical Speed and Pitch Hold. Also, a VNAV Path mode is available for descent where the FMS coordinates with the AFCS to maintain a programmed path angle. The Autothrottle controls airspeed (within engine limits) in all three of these descent modes.

1.3 Control Laws

Thrust control is achieved by providing a Throttle rate command. To be able to achieve the desired behaviour in these different modes, control laws have to be designed and incorporated in the autothrottle system. We can observe two main families of control laws, either classical or modern. Most civil aircrafts use classical control laws. These laws work on different channels for each mode separately as a single input single output control law.

Cross-coupling errors in speed and altitude occur when manoeuvring due to the design of autopilots and autothrottles as single input single output control systems. For example, a speed change can not be accomplished by only a change in throttle setting, but must be accompanied by an elevator retrim if altitude is to be maintained. Autopilot, autothrottle, and flight management systems (FMS) control laws have been developed over a long period of time which has led to the duplication of functions in the autopilot and FMS computer.

These problems led to many studies on the integration of modern control laws. The objective of such studies was to develop an all longitudinal control modes for the autopilot/autothrottle and flight management system using a common, generalized flight path and speed control algorithm, thereby eliminating unnecessary control law replication and providing a flight management system (FMS) simplification. Further elaboration on some of these methods is presented later on in this chapter.

1.3.1 EPR Control Law

The objective of an EPR control law is to capture a commanded EPR. Hanke & Nordwall (1970) as well as Robbins & Simpson (1991) cite that the relationship between EPR and an equivalent throttle rate command is a function of temperature as seen in Figure 1.4

The curve between two points A and B for a selected temperature can be approximated by a straight line. From this approximation, the slope of the line may be computed:

$$\Delta EPR = K \cdot \Delta \delta T \quad (1.1)$$

The variation in the value of K compared to the change in temperature is small, as cited by Robbins & Simpson (1991), and thus a constant value of K may be chosen for use at all reasonable temperatures. The control achieves then the required EPR level by commanding a throttle rate proportional to the EPR error.

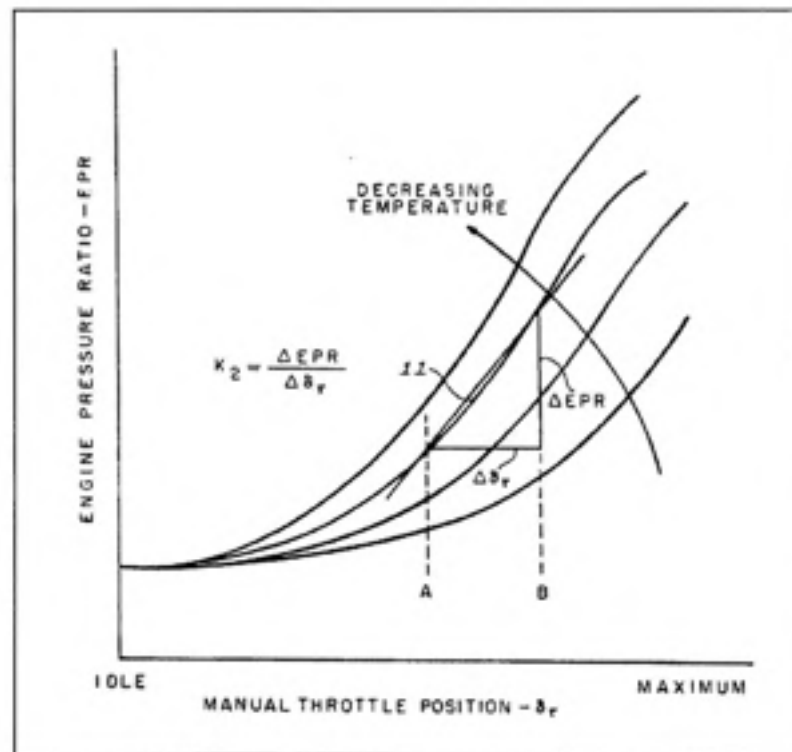


Figure 1.3 EPR-Throttle position curve, Robbins & Simpson (1991).

Figure 1.4 shows the basic simplified block diagram for an EPR control device:

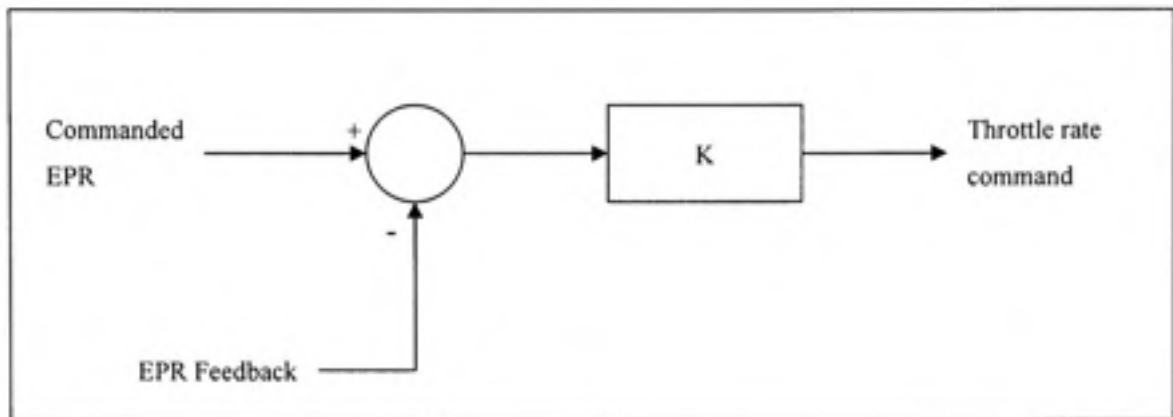


Figure 1.4 EPR Control Block Diagram, Bader (1990).

Usually EPR control laws are used in flight level change mode or takeoff (and go-around). These are the two main phases in a flight envelope where an EPR control system would be in need.

As a matter of fact, takeoff is a critical stage of flight. The increased bandwidth requirement during Takeoff (the commanded thrust must be attained prior to throttle-hold) calls for a high loop gain. This could result in thrust overshooting the command if additional compensation were not applied. A possible compensation used to preclude overshoots proposed by Bader D.L. (1990) is in the form of command biases which are proportional to EPR rate and throttle rate, as shown in Figure 1.5:

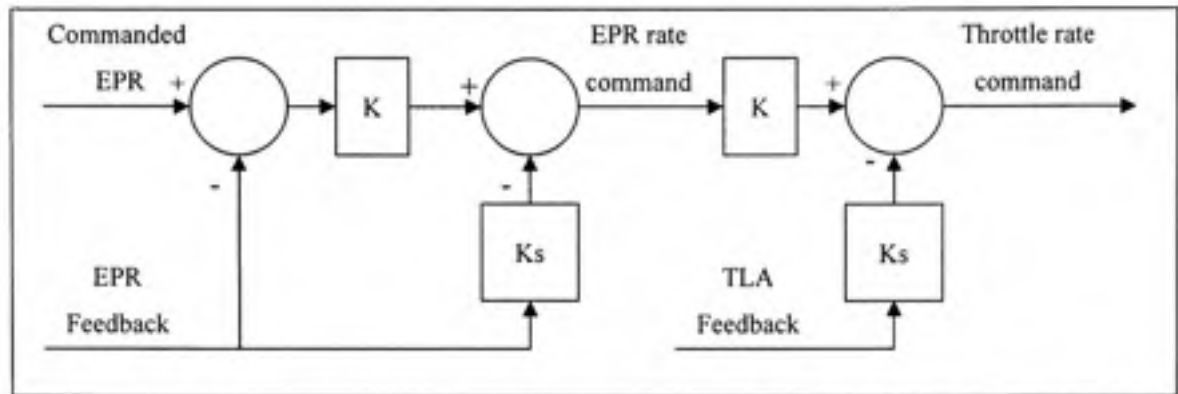


Figure 1.5 EPR Control Block Diagram for Takeoff phase, Bader (1990).

1.3.2 Speed Control Law

Aircraft speed is controlled by changing the engine thrust. This is achieved by altering the quantity of fuel flowing to the engines using the engines throttles. Typical automatic control of the aircraft's airspeed can be achieved by a closed loop control system whereby the measured airspeed error is used to control throttle servo actuators which operate the engine throttles. The engine thrust is thus automatically increased or decreased to bring the airspeed error to near zero and minimize the error excursions resulting from disturbances. Collinson R.P.G. (1996) presents a typical airspeed control system as shown in the block diagram in Figure 1.6.

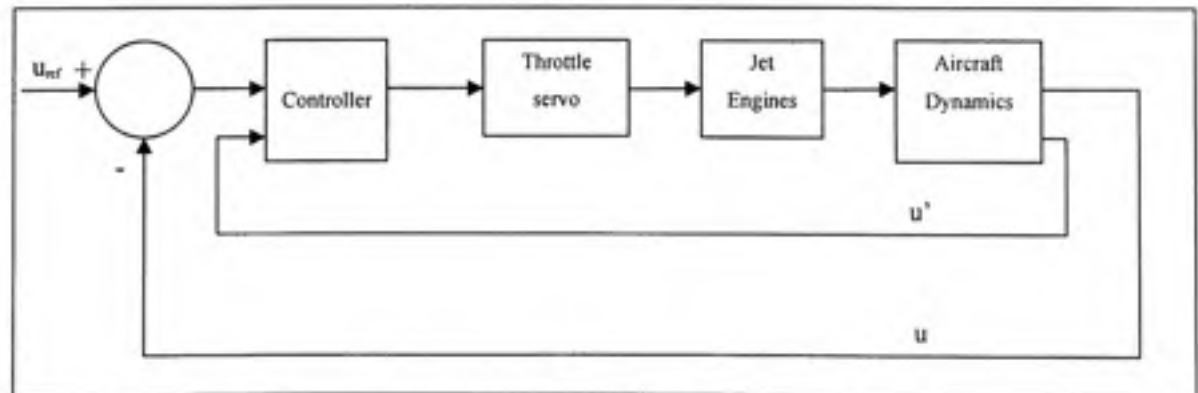


Figure 1.6 Speed control simplified block diagram, Collinson (1996).

In any closed loop system, the lags in the individual elements in the loop resulting from energy storage processes (eg. accelerating inertias) exert a destabilising effect and limit the loop gain and hence the performance of the automatic control system. The dynamic behaviour of the engines over the range of flight conditions, the throttle actuator response and the aircraft dynamics must thus be taken into account in the design of the speed control system. Collinson (1996) cites that the response of the jet engine thrust to throttle angle movement is not instantaneous and approximates to that of a simple first order filter with a time constant which is typically in the range 0.3 to 1.5 seconds, depending on the thrust setting and flight condition. Clearly, the lag in the throttle servo actuator response should be small compared with the jet engine response. The aircraft dynamics introduces further lags as a change in thrust produces an acceleration (or deceleration) so that integration is inherent in the process of changing the airspeed. The derivation of airspeed from the air data system can also involve a lag.

The rate of change of forward speed, U' , derived from a body mounted accelerometer with its input axis aligned with the aircraft's forward axis, can provide a suitable stabilising term for the control loop. A proportional plus integral of error control is usually also provided to eliminate steady state airspeeds errors.

Bader D.L. (1990) proposes another approach, in which the EPR feedback is used as the stabilizing term for the control loop. EPR which is a pressure ratio serves as a measure of thrust and is hence usually proportional to acceleration. Having EPR in the loop presents several advantages such as EPR and engines limits protection in the loop. However, the design of such a control law would require an accurate EPR sensor modelization as well as accurate EPR/Thrust curves representations.

1.3.2.1 IAS Hold Control law

In the Indicated airspeed hold loop suggested by Bader D.L. (1990) and presented in Figure 1.7, the commanded IAS is first compared to the measured IAS and a speed error is generated. An EPR rate command is calculated by applying lead-lag compensation (with a gain based on aircraft response) to the speed error. To enhance the performance, this command is biased by a thrust prediction term which anticipates required thrust adjustments that result from changes in aircraft drag. A second thrust prediction term which compensates for changes in flight path angle can also be included during certain aircraft manoeuvres. The EPR rate command is then integrated and bounded by engine limits to determine the commanded EPR level. The EPR rate command is then integrated and bounded by engine limits to determine the commanded EPR level.

Then a closed EPR loop, similar to the one in the EPR control law section, provides an increased degree of stability and accuracy to the Autothrottle system during Speed Hold operation. As cited before, the resulting throttle rate command is proportional to EPR error.

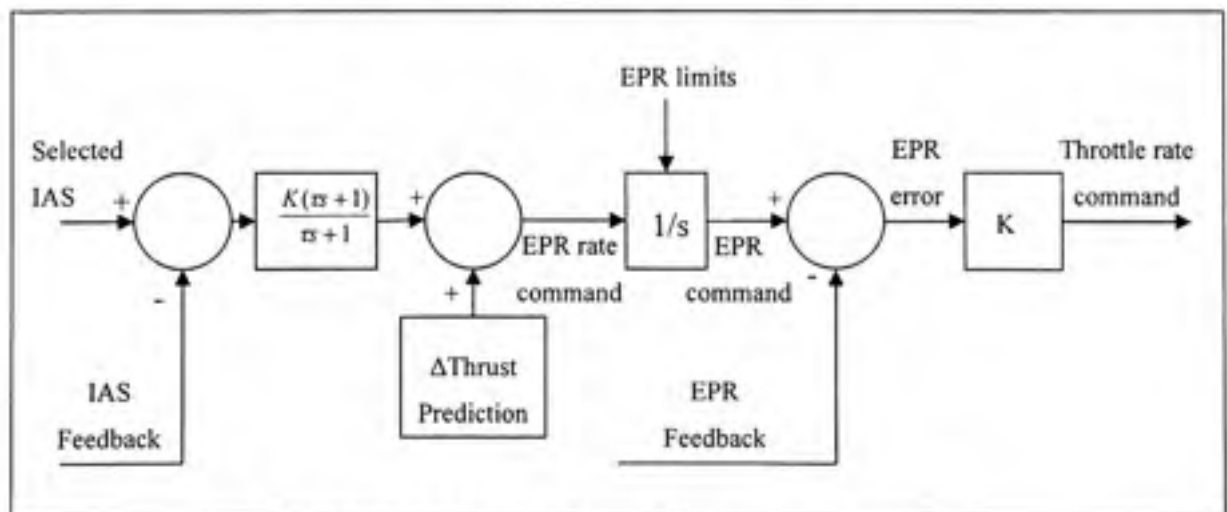


Figure 1.7 IAS Control Block Diagram, Bader (1990).

To add speed protection and restrict speed to the speed limits, we may add a speed command processor. The airspeed target input to the autothrottle is first evaluated by the mentioned

processor. This processor ensures that maximum airspeeds are not violated (flap, gear, and VMO/MMO limits).

Also we may apply a second order filter with rate limiting to speed target changes to achieve smooth throttle movements. We may also add a complementary filter to the airspeed feedback to reduce the effects of windgusts. A complementary filter is used when rate information must be extracted from a signal containing high frequency noise terms. These unwanted terms must be filtered out without the loss of actual signal information above the cut-off frequency of the filter. The required compliment signal is the derivative of the control signal. It is then important that the derivative be obtained from a different source to ensure that it does not contain the unwanted noise. We can use for example the true airspeed and as another source for the derivative the aircraft inertial acceleration along the flight path.

1.3.2.2 MACH Hold

The MACH Hold control law could be designed similar to the IAS Hold control law. It is a speed ratio control law and MACH number is obtained directly from speed and air data measures:

$$MACH = \frac{\textit{Airspeed}}{\textit{Speed of sound}} \quad (1.2)$$

1.3.3 Thrust control with pitch coupling

If a pitch attitude variation is required, thrust has to be corrected to keep speed constant and, conversely, action on thrust to increase speed induces a pitch movement. It is possible to obtain such coupling by sending a thrust variation prediction term to the thrust control law as cited earlier. Similarly, we may send an elevator variation prediction term to the altitude hold or pitch control law to compensate for the pitch variation due to thrust action. Since the

engine lag and slow thrust response usually limits the effect of elevator variation on thrust control, it is of more interest to couple the thrust variation effect on the elevator to compensate for any variation in pitch due to speed and thrust variation.

It is possible to determine a relation between pitch and thrust around a trim point, and thus predicting the effects a pitch or thrust variation would have on the other control output. We elaborate and present the relation further in section 3.7.4.

1.4 Modern integrated control laws: MIMO Systems

In this section we will present the main modern control methods, without citing major details. As cited earlier in this review, cross-coupling errors in speed and altitude occur when manoeuvring due to the design of autopilots and autothrottles as single input single output control systems.

A multi-input pitch movement approach will make it possible to coordinate the two control channels, by means of pitch and thrust control coupling, in order to perform pitch manoeuvres at constant speed without adjusting thrust on one hand, and speed variations at constant pitch attitude without acting on the stick, on the other. Such algorithms provide full-time thrust and elevator control coordination for all modes and flight conditions, thus eliminating speed deviation due to flight path manoeuvres and flight path deviations due to speed manoeuvres. Two main methods have been cited in the literature:

- The total energy control system
- Longitudinal law using the eigenvector method

These methods will be presented briefly. A more detailed study could be necessary if such a method would be used during the implementation of the autothrottle system.

1.4.1 The total energy control system

Bruce K.R. (1988) avoided the classical controls coupling errors by integrating a total energy control system. The objective of the total energy control system (TECS) design was to develop all longitudinal control modes for the autopilot/autothrottle and flight management system using a common, generalized flight path and speed control algorithm, thereby eliminating unnecessary control law replication and providing a flight management system (FMS) simplification. Lambregts (2000) also proposes a total energy based flight system that computes thrust and elevator commands. An overview of the controller architecture is presented in Figure 1.8:

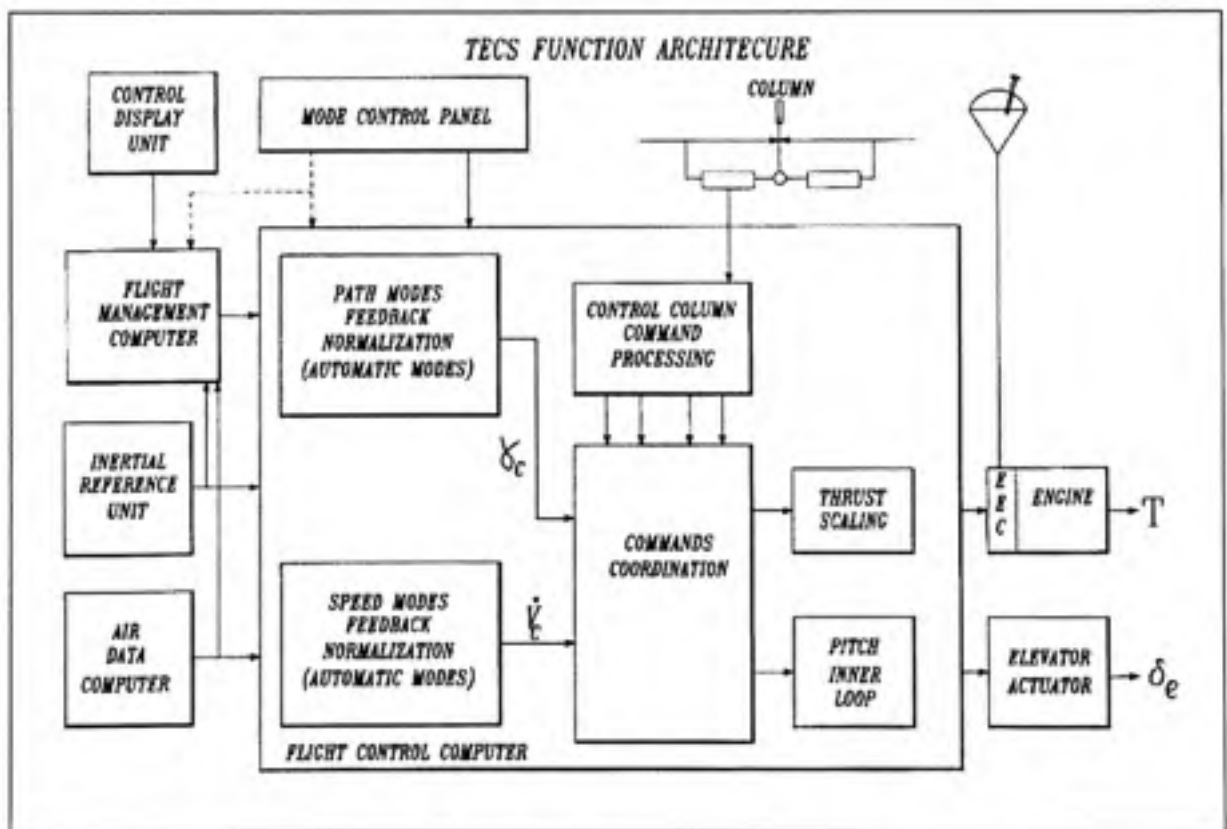


Figure 1.8 Total Energy Control System Block Diagram, Lambregts (2000).

In the presented system, an integrated aircraft longitudinal flight control system uses a generalized thrust and elevator command computation, which accepts flight path angle, longitudinal acceleration command signals, along with associated feedback signals, to form energy rate error and energy rate distribution error signals. The basic design philosophy of a TECS is to compute an aircraft's total energy state and its desired energy state, as represented by flight path, speed and their associated targets. The engine thrust command is developed as a function of the energy rate distribution error and the elevator position command is developed as a function of the energy distribution error. For any vertical flight path and speed mode the outer loop errors are normalized to produce flight path angle and longitudinal acceleration commands. The system provides decoupled flight path and speed control for all control modes previously provided by the longitudinal autopilot, autothrottle and flight management systems. For all flight conditions, thrust is the most effective means to change the aircraft's energy state, whereas elevator control provides an effective means to modulate energy distribution and stabilize the aircraft's attitude.

1.4.2 Longitudinal law using the eigenvector method

The problem of multi-input multi-output control in this case is solved using the Eigenvector theory. A number of constructive methods for eigenstructure assignment by state feedback are described in the literature. Kautsky & Nichols (1985) describe a robust pole assignment method that uses eigenstructure assignment to achieve robustness. It is possible, with this approach, to control the aircraft closed loop dynamics, obtain good decoupling by a suitable eigenvector structure while taking servo-controls engine time-constant and delays due to computers and parameter acquisitions into account. Larramendy P. (1991) presents the steps used for the design of such a controller applied to a civil aircraft.

The interest in such a method lays in the fact that thanks to the use of the Eigenvectors, the system will be implemented very easily. On each channel in order to calculate the feedback loop, all that has to be done is sum each aircraft parameter measured multiplied by a gain.

Figure 1.9 presents the block diagram of a typical state feedback controller. The theory and design of such a controller is presented further in detail in chapter 5.

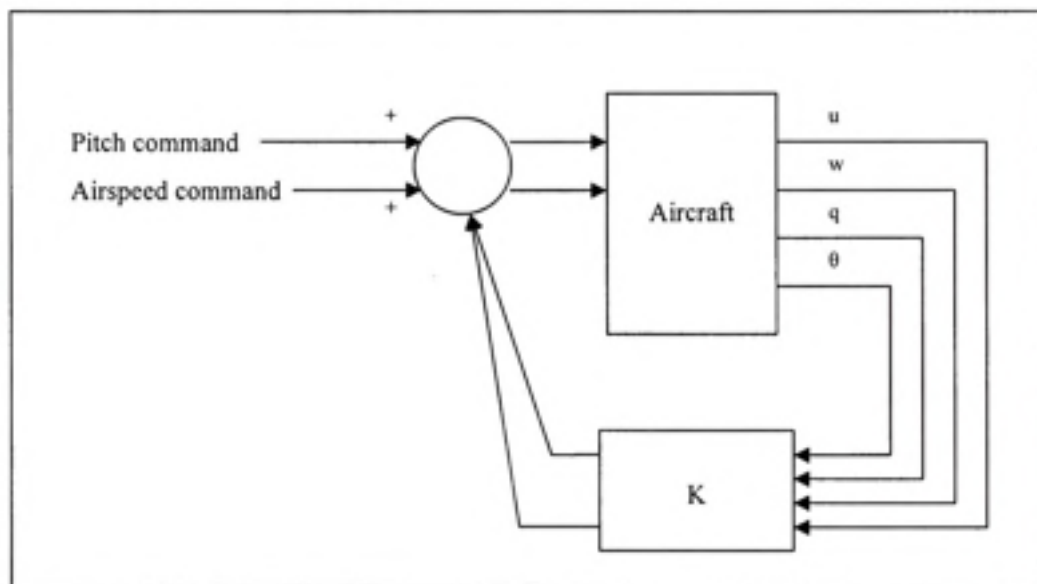


Figure 1.9 Block diagram of a typical state feedback controller.

1.5 Entire flight envelope validation

If we consider altitude and speed as variables, flight envelope can be given by the area which gives for each altitude the values of speed tolerated (for a secured flight). This area varies according to the aircraft gross weight. We may consider the weight constant over a certain small lapse of time and hence study the envelope for a given weight.

For example, we may consider three weight configurations for the flight of a Boeing 747:

- minimal weight, at landing = 450 000 lbs
- median weight, at cruise conditions = 550 000 lbs
- maximal weight, at takeoff = 650 000 lbs

The envelope of a simulated Boeing 747 for these weight configurations is then given by the graph presented on the next page (Figure 1.10), as presented by Derbois A. (2002).

When one of these weights is chosen, it is considered constant throughout the simulation and we then obtain a flight envelope considering mass is constant throughout the whole flight. Hence it is not an exact envelope but a valid approximation if used in the correspondent phase of flight.

We want to be able to apply the autothrottle control laws to the entire flight envelop. The control laws presented previously have been studied for a linear system, and the aircraft dynamics represent, as most systems, a nonlinear system. Hence, it is attractive to adopt a divide and conquer approach whereby the analysis/design task for a nonlinear system is decomposed into a number of linear tasks.

The design of the Autothrottle system could then be based initially on linear conditions such in cruise conditions of the aircraft. We may then address different modifications to ensure a valid system for other phases of the flight.

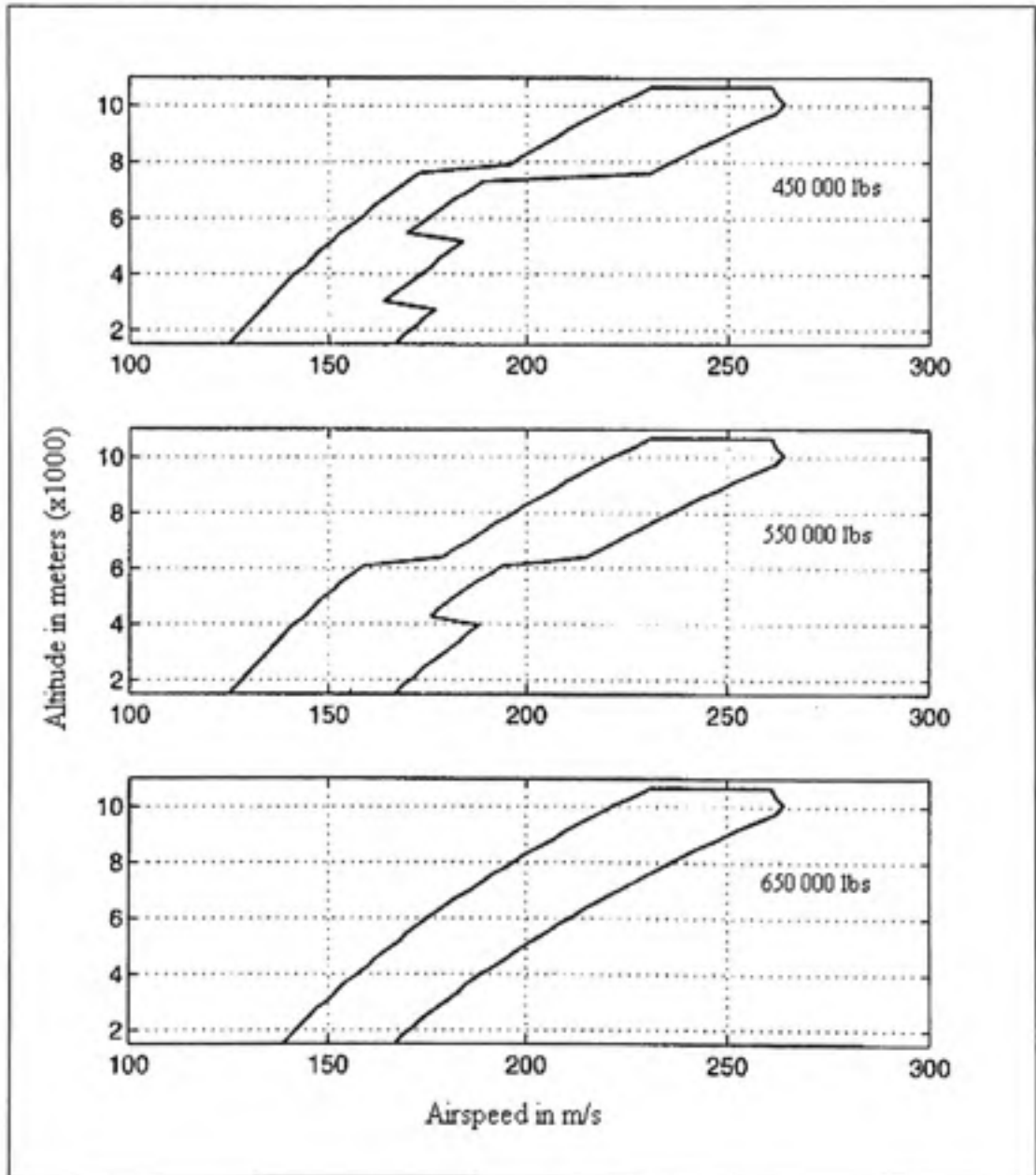


Figure 1.10 Flight envelope for various weight values, Derbois (2002).

Gain-scheduling is a divide and conquer approach for the design of nonlinear control systems. The conventional gain-scheduling design approach typically involves:

1. Linearize the nonlinear plant about a number of equilibrium points
2. Design a linear controller for each of the plant linearizations
3. Combine the linear controllers to obtain a nonlinear controller

Leith & Leithead (2002) present a survey of gain-scheduling methods. Nonlinear control methods may permit to design an entire flight envelope system, but such approaches are difficult to implement and analysis/design of nonlinear systems control is poorly developed.

CHAPITRE 2

SYSTEM REQUIREMENTS

2.1 Introduction

As any controller, the Thrust Control System must meet a list of requirements. These requirements were subject to approval by CMC Electronics. In fact, CMC is the client to which the TC is delivered, and so the controller has to meet a list of needs formulated by the company. As a matter of fact, the requirements were formulated after extensive research within the industry's controllers, and then submitted to CMC for approval. We will present in the next sections the main requirements for our system. The requirements will later on be the base on which the controller will be tested and evaluated for approval, as presented and discussed in chapter 7.

As cited in the previous chapter, the purpose of the TC System is to enhance the ease, economy, and safety of flight operations. These are the main guidelines that define the controller's requirements. Flight economy is improved by accurate speed control and thrust management, while safety is enhanced by protecting maximum placards airspeeds, observing engine limits, and by reducing pilot workload.

The TC system will control the engine's thrust by outputting a throttle rate command to the TLA (Thrust Lever Actuator) servo. All control laws will be developed around a Boeing 747-400 aircraft model, the same model used within the DTB.

Requirements will be presented through the main TC features. The features of the proposed TC System model shall include:

1. Thrust capture
2. Speed capture

3. EPR constraints and engine limits
4. Speed and acceleration limits and placards
5. Flight envelope validity
6. Pitch coupling

2.2 Thrust capture

The objective of thrust capture is to command thrust of the aircraft directly thru EPR. The Automatic control of the aircraft's EPR can be achieved by a closed loop control system whereby the measured EPR error is used to control throttle servo actuators which operate the engine throttles. This mode is usually used during flight level change or takeoff where maximum thrust is required. It may also be used during conditions where a certain IDLE throttle position is to be attained.

When Thrust mode is selected, one of two sub modes can be selected. When FMS is in Thrust mode EPR the Thrust Controller shall generate throttle rate commands to maintain the engine at the desired EPR. When FMS is in Thrust mode IDLE the TC shall generate throttle rate commands to retard the throttle.

For step command inputs, the response shall be smooth and overshoot free. To assure a smooth change in thrust, throttle rate command shall be limited. Throttle rate command limit shall be between 8 and 10 deg/sec., or 10% of thrust per second as proposed by Hanke & Nordwall (1970).

2.3 Speed capture

The objective of speed capture is to control the speed of the aircraft through throttle's rate. Automatic control of the aircraft's airspeed can be achieved by a closed loop control system

whereby the measured airspeed error is used to control throttle servo actuators which operate the engine throttles. The engine thrust is thus automatically increased or decreased to bring the airspeed error to near zero and minimise the error excursions resulting from disturbances. Whenever a difference exists between the commanded airspeed and the indicated airspeed the TC shall generate throttle rate commands to reach the requested IAS when FMS is in Thrust mode SPD, with IAS selected, and throttle rate commands to reach the requested MACH when FMS is in Thrust mode SPD, with MACH selected.

For step command inputs, the response shall be smooth and overshoot free.

A general idea concerning autothrottle performance on a Boeing 747 obtained from several simulations presented by Hanke & Nordwall (1970) show that acceleration from Mach 0.65 to maximum speed (around Mach 0.85) would require around 4 to 5 minutes for gross weight of 450 000 lbs and altitude of 35 000 ft.

The deceleration from Mach 0.85 to thrust IDLE position (around Mach 0.5) would require around 3 minutes for gross weight of 450 000 lbs and altitude of 35 000 ft.

In general response time depends on the flight phase and flight conditions (approach, speed constraints, etc...). Response time should not be a specific constraint during this study. The time the system would take to achieve minimum or maximum velocity will depend on the throttle rate command limit. Throttle rate command shall be limited to a value of 8 to 10 deg/sec. or 10% of max thrust per second. This limit will assure a smooth change in thrust.

When the Autothrottle speed mode is engaged, an airspeed error greater than 10 knots shall cause a warning visible to the pilot.

2.4 EPR constraints and engine limits

The TC shall respect the EPR limit constraint from the FMS. The design should incorporate foolproof safeguards to prevent stall and overboost of the engine for all modes and flight conditions. It shall limit the forward and aft throttle motion. The forward position is set to avoid exceeding the maximum allowable engine pressure ratio or temperature. The aft position closely corresponds to the flight IDLE thrust value.

2.5 Speed and acceleration limits and placards

The airspeed target input to the autothrottle must first be evaluated to ensure that maximum airspeeds are not violated (VMO/MMO limits). The TC shall then restrict the speed to the VMO and MMO. These limits will be received during all phases of flight by the FMS. A demand to capture a speed outside of the permitted limits shall cause a warning visible to the pilot. If the pilot selects a speed to capture via the TC system outside of the permitted limits, the TC shall modify the speed command in order to respect speed limits.

The speed shall not exceed 250 knots at 10 000 ft or below, as restricted by the FAA.

For our implementation case on a Boeing 747, placards are given by (Hanke & Nordwall (1970)):

- MMO (Mach Max Operating Speed): 0.85
- MNE (Mach Never Exceed Speed): 0.92
- VMO (Max Operating Speed IAS): 507 knots.

The TC shall maintain the speed over the Minimum Operating Speed to avoid stall. Stall speed for a Boeing 747 at full flap/gear down is around 120 knots (Hanke & Nordwall (1970)).

The TC shall limit the acceleration imposed on the aircraft to within normal operating limits (G limit) to guarantee passengers comfort. Passenger's comfort may be guaranteed by a G-limit of 0.1g.

The speed protection shall be the highest priority protection between the TC protections listed above (including EPR protection).

2.6 Flight envelope validity

The TC should support operation over the flight envelope. There should be a consistent response over the aerodynamic envelope. The control laws should not command the aircraft to leave its envelope and commands should be restricted to within the flight envelope.

2.7 Pitch coupling

When the autothrottle is commanding speed, the controller shall send a Short term deviation command to the pitch inner loop to anticipate the pitch change due to speed acceleration (maintain the altitude) if necessary. The main objective and requirement is to maintain altitude while increasing or decreasing speed using thrust.

CHAPITRE 3

AIRCRAFT DYNAMICS AND MODELISATION

In this chapter, the equations of movement of the aircraft shall be presented. We will then linearize and simplify these equations to obtain a simplified longitudinal model of the aircraft, which will be used in our control law design, applied to a specific linear Boeing 747-400 aircraft model. We will also present and study the general characteristics of our linear model.

3.1 Basic flight principles

3.1.1 Forces of flight

Four main forces are applied to the aircraft as shown in Figure 3.1: Weight, Lift, Drag and Thrust.

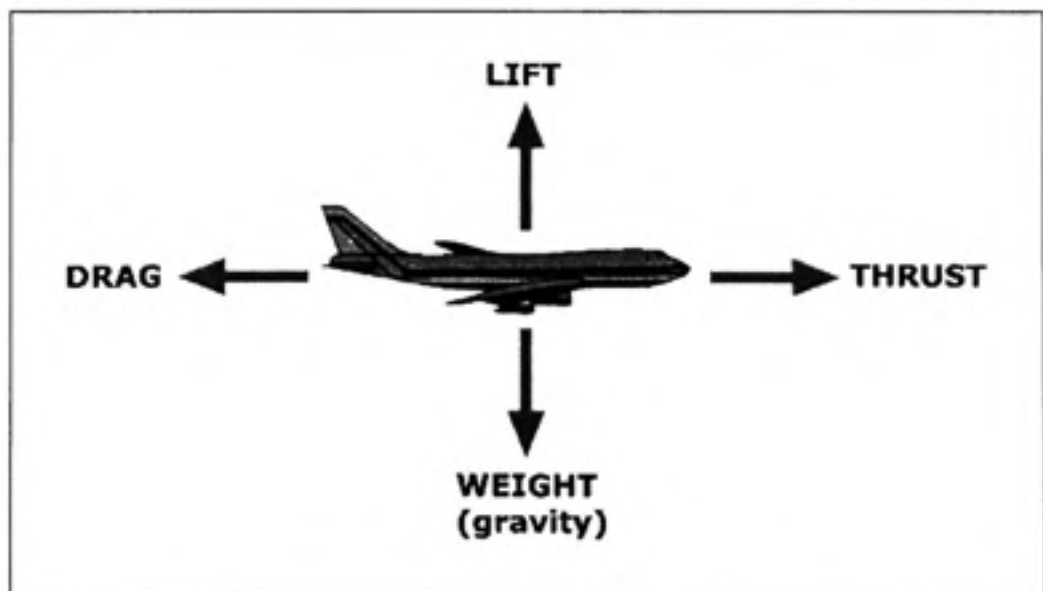


Figure 3.1 Forces of flight [30].

- **Weight:** The weight of the aircraft is always directed toward the center of the earth. The magnitude of the weight depends on the mass of all the airplane parts, plus the

amount of fuel, plus any payload on board. The weight is distributed throughout the airplane. It is however often represented as collected and acting through a single point, the center of gravity. In flight, the airplane rotates about the center of gravity. Flying encompasses two major problems; overcoming the weight of an object by some opposing force, and controlling the object in flight. Both of these problems are related to the object's weight and the location of the center of gravity. During a flight, an airplane's weight constantly changes as the aircraft consumes fuel. The distribution of the weight and the center of gravity also changes. So the pilot must constantly adjust the controls to keep the airplane balanced, or trimmed.

- **Lift:** Lift is generated by the motion of the airplane through the air and is an aerodynamic force. Lift is directed perpendicular to the flight direction. The magnitude of the lift depends on several factors including the shape, size, and velocity of the aircraft. As with weight, each part of the aircraft contributes to the aircraft lift force. Most of the lift is generated by the wings. Aircraft lift acts through a single point, the center of pressure. The center of pressure is defined just like the center of gravity, but using the pressure distribution around the body instead of the weight distribution. Aerodynamic surfaces are used to control the aircraft in roll, pitch, and yaw.
- **Drag:** Drag is the aerodynamic force caused by the air resistance as the aircraft moves through the air. Drag is directed along and opposed to the flight direction. Like lift, there are many factors that affect the magnitude of the drag force including the shape of the aircraft, the "stickiness" of the air, and the velocity of the aircraft.
- **Thrust:** Thrust is the force generated to overcome drag using a propulsion system. The direction of the thrust force depends on how the engines are attached to the aircraft.

3.1.2 Control surfaces

Every aircraft has control surfaces or other means which are used to mainly modify the aerodynamic properties of the aircraft, thus generating the forces and moments required to produce the accelerations which cause the aircraft to be steered along its three dimensional flight path to its specific destination. A conventional aircraft is represented in Figure 3.2. It is shown with the usual control surfaces, namely elevator, ailerons and rudder. Such conventional aircraft have a fourth control, the change in thrust, which is generally obtained from the engines.

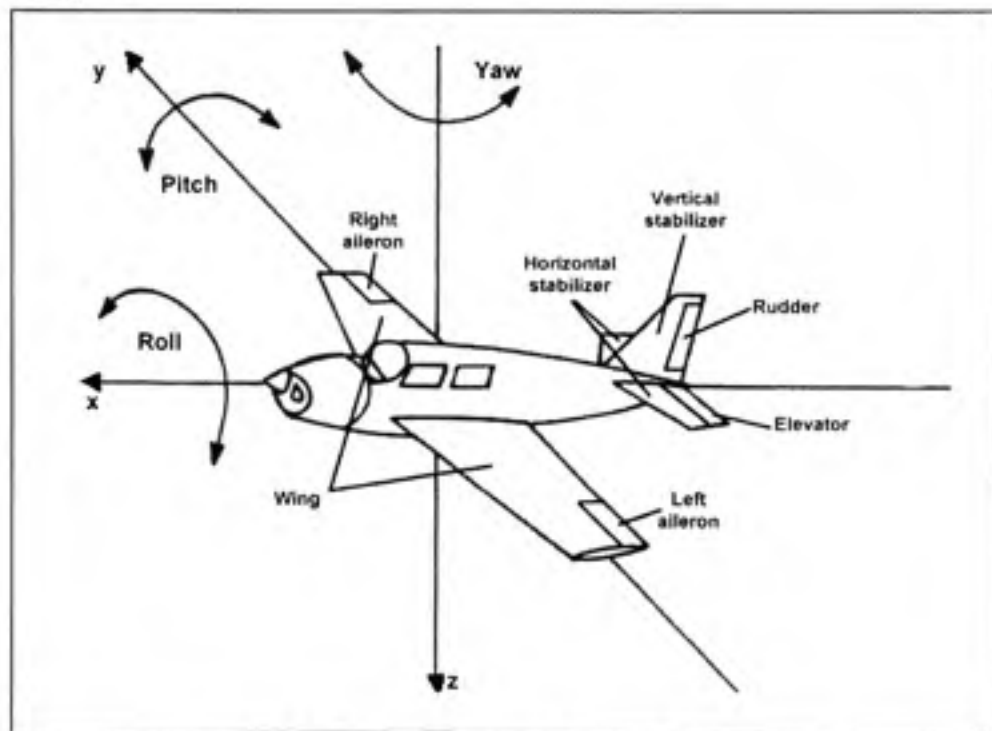


Figure 3.2 Basic Aircraft Control Surfaces [28].

The elevators which are on the tail section are used to control the pitch of the plane. A pilot uses a control wheel to raise and lower the elevators, by moving it forward to back ward. Lowering the elevators causes the plane nose to pitch down while raising the elevators generates a pitch up movement. The two elevators move with the same angle and generate an aerodynamic couple around the lateral axis, causing the pitching movement.

The ailerons are disposed symmetrically with respect to the pitch axis, and move in opposite directions. The pilot controls the roll of the plane by raising one aileron or the other with a control wheel. For example, turning the control wheel clockwise raises the right aileron and lowers the left aileron, which rolls the aircraft to the right. The ailerons generate an aerodynamic couple around the longitudinal axis, and thus causing roll movement.

The rudder works to control the yaw of the plane. The pilot moves rudder left and right, with left and right pedals. Pressing the right rudder pedal moves the rudder to the right. This yaws the aircraft to the right. Used together, the rudder and the ailerons are used to turn the plane.

Engine power is controlled using the throttle. Obviously, pushing the throttle increases power, and pulling it decreases power.

We may find other control surfaces on aircrafts. It is possible that some aircrafts have two or more control surfaces linked to produce coordinated manoeuvres. It is also possible to have combined control surfaces as on the delta wings of a Concorde. In that case one control surface may be used to roll or pitch the aircraft (same control surface used as elevators and ailerons).

3.1.3 Coordinate systems

3.1.3.1 Inertial frame

Often referred to as Earth centered Inertial or ECI, it has its origin at the Earth's centre. As shown in Figure 3.3, the Z axis is on the rotational axis pointing towards North. X and Y axis are in the equatorial plane, their direction is arbitrary. The Earth Centered Earth Fixed (ECEF) frame is also presented in Figure 3.3. It is a particular case of the ECI frame, in which the direction of the X axis indicates zero longitude and zero latitude.

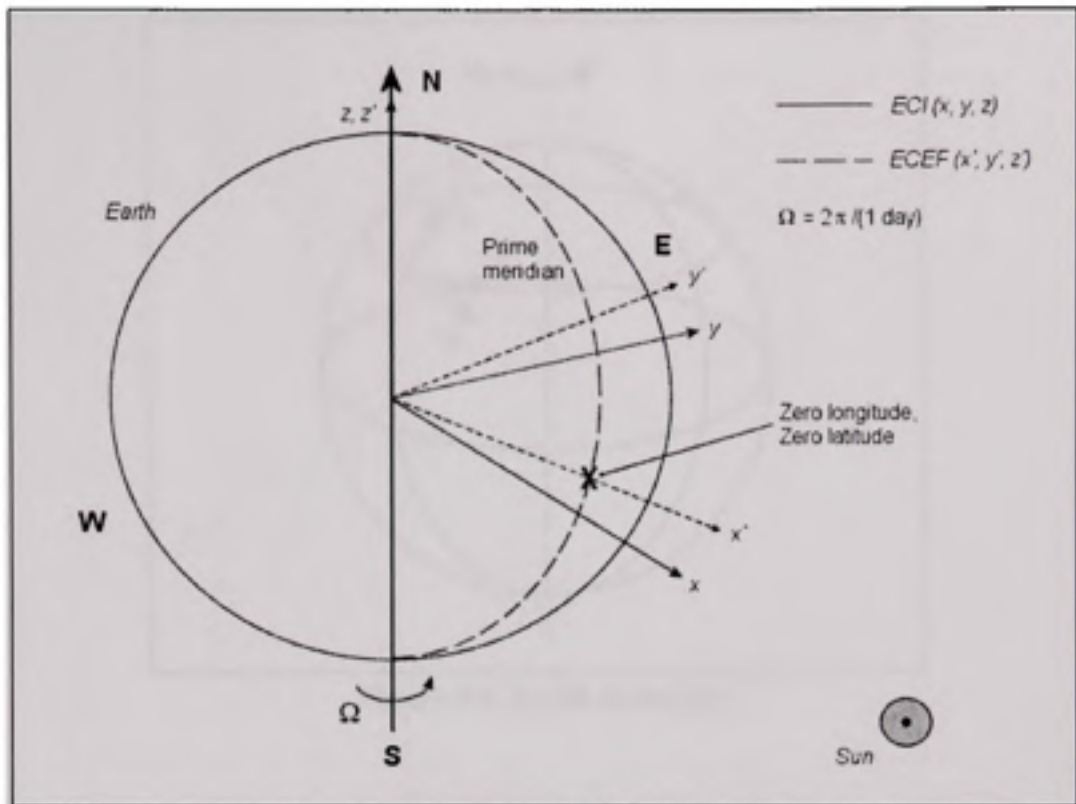


Figure 3.3 ECI and ECEF frame [31].

3.1.3.2 Earth frame

Often referred to as NED frame, is located at Earth's surface. The centre of the frame is the projection of the aircraft's Centre of Gravity onto the Earth's surface (moves with aircraft). The X axis points towards North direction, Y axis towards East, and the Z axis towards the centre of gravity of the Earth (along the Gravity direction), hence the name NED: North, East, Down. The plane defined by the X and Y axes is tangent to the surface of the Earth. Figure 3.4 shows the NED frame.

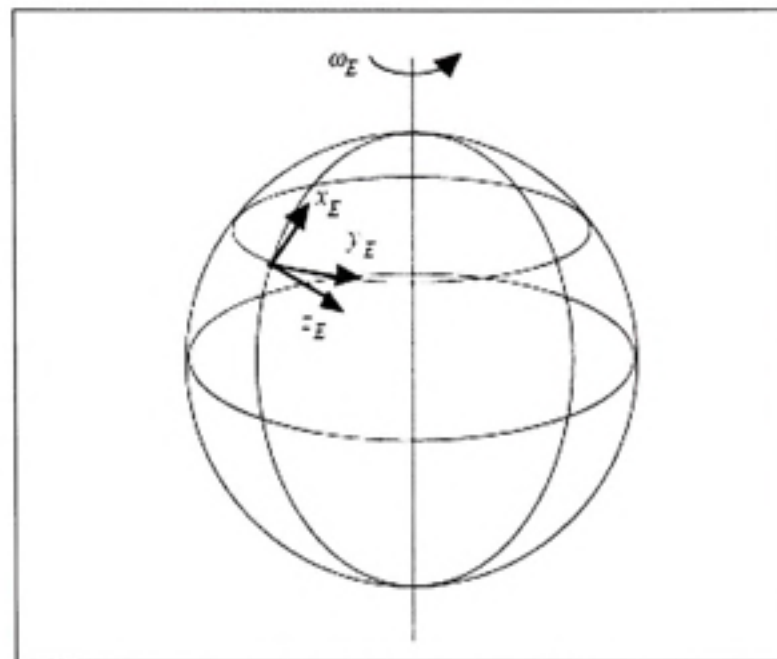


Figure 3.4 Earth frame [6].

3.1.3.3 Body frame

The set of body axes is specially chosen to coincide with the principal axes of the aircraft. The origin is the Centre of Mass of the aircraft. The X axis, same as the aircraft longitudinal axis, points towards the aircraft's nose. The Y axis points towards the right of the pilot, usually in the right wing direction, while the Z axis points down. Figure 3.5 shows the body frame:

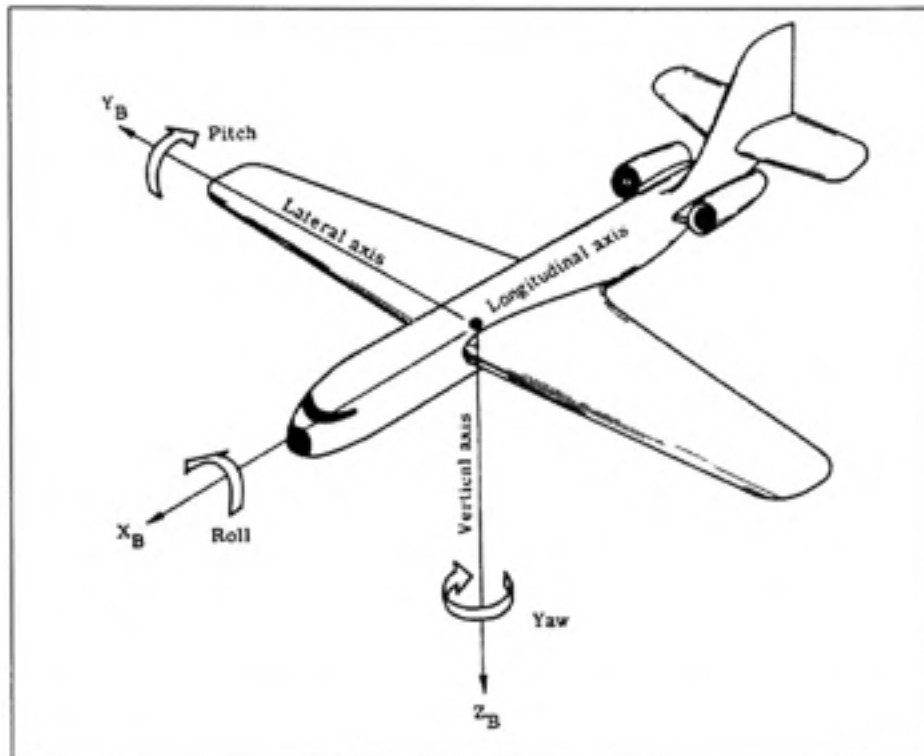


Figure 3.5 Body frame [19].

Variables and notations in the body frame

The speeds, angular velocities, forces and moments in the body frame are defined as follows:

- U: Longitudinal speed, along X body axis
- V: Lateral speed, along Y body axis
- W: Vertical speed, along Z body axis
- P: Roll angular velocity, around the body X axis
- Q: Pitch angular velocity, around the Y body axis
- R: Yaw angular velocity, around the Z body axis
- X: Force applied to the aircraft along the X axis
- Y: Force applied to the aircraft along the Y axis
- Z: Force applied to the aircraft along the Z axis

- L: Roll moment
- M: Pitch moment
- N: Yaw moment

3.1.3.4

Euler angles

Any body rotation within a frame can be expressed as three fundamental rotations. In the aeronautical community, the convention for the three rotations is as follows:

- Yaw, Ψ , the angle of rotation around the body Z axis
- Pitch, Θ , the angle of rotation around the body Y axis, with respect to the horizon
- Roll, Φ , the angle of rotation around the X body axis.

The angles of rotation Ψ , Θ and Φ are referred to as the Euler angles. These angles are used to pass from the Earth (NED) frame to the Body frame and vice-versa.

3.1.3.5 Wind frame

The wind frame is oriented with respect to the aircraft flight path, time varying terms which correspond to the moments and cross products of inertia appear in the equations of motion. Such terms considerably complicate the analysis of aircraft motion and, consequently, the wind frame is not used within this study. However, a couple of notations are relevant to note.

Aircraft airspeed, or velocity vector V_T

The aircraft velocity with respect to the Inertial frame is the sum of the Aircraft airspeed (velocity of the aircraft with respect to the air), and the speed of the air with respect to the Inertial frame. In most references, the authors make the assumption that the air speed with respect to the Inertial frame is zero, as mentioned by Mclean (1990), and that the aircraft velocity is in fact the aircraft airspeed. The aircraft airspeed (V_T) is located in the body frame using two angles, angle of attack (α) and the slide-slip angle (β).

Projecting the Velocity vector on the Body frame axes, we may write:

$$U_T = |\vec{V}_T| \cos \beta \cos \alpha \quad (3.1)$$

$$V_T = |\vec{V}_T| \sin \beta \quad (3.2)$$

$$W_T = |\vec{V}_T| \cos \beta \sin \alpha \quad (3.3)$$

It is important to note that the angle of attack is different from the pitch angle. While the angle of attack is the angle between the X body axis and the velocity vector V_T direction, the pitch angle is defined as the angle between the X body axis and the horizon. The angle between the velocity vector direction and the horizon is defined as the path angle, γ (Figure 6.4). We may then note (for longitudinal movement with no Bank angle):

$$\theta = \alpha + \gamma \quad (3.4)$$

3.2 Equations of motion

In this section we shall study the aircraft movement with respect to the earth and air. It is necessary to make a couple of assumptions in order to study the aircraft kinematics.

We assume that:

- 1- The aircraft is a rigid body, and thus the movement of the aircraft is expressed as the translation and rotation of its center of gravity. We can therefore apply Newton's second law to obtain the equations of movement.
- 2- The inertial frame has no acceleration. The earth is fixed within space. Movement is described with respect to the inertial Earth centered and fixed frame.

3.2.1 Translational motion

From Newton's second law it can be deduced that:

$$\vec{F} = \vec{F}_0 + \Delta\vec{F} = m \frac{d}{dt}(\vec{V}_T) \quad (3.5)$$

where \vec{F} represents the sum of all externally applied forces. The subscript 0 denotes the equilibrium component while the Δ represents the component to the perturbation. The axis system being used is the Earth frame.

By definition, equilibrium flight must be unaccelerated flight along a straight path; during this flight the linear velocity vector relative to fixed space is invariant, and the angular velocity is zero. Thus \vec{F}_0 is zero.

The rate of change of V_T relative to the Earth axis system is given by:

$$\frac{d}{dt}(\vec{V}_T)_E = \frac{d}{dt}(\vec{V}_T)_B + \vec{\omega} \times \vec{V}_T \quad (3.6)$$

where $\vec{\omega}$ is the angular velocity of the aircraft with respect to the fixed axis system.

Expressing the vectors in the body frame axis system, both vectors can be written as the sum of their corresponding components, as follows:

$$\vec{V}_T = \vec{i}U + \vec{j}V + \vec{k}W \quad (3.7)$$

$$\vec{\omega} = \vec{i}P + \vec{j}Q + \vec{k}R \quad (3.8)$$

$$\frac{d}{dt}(\vec{V}_T)_B = \vec{i}\dot{U} + \vec{j}\dot{V} + \vec{k}\dot{W} \quad (3.9)$$

In a similar fashion, we can write:

$$\Delta\vec{F} = \vec{i}\Delta X + \vec{j}\Delta Y + \vec{k}\Delta Z \quad (3.10)$$

From which it can be inferred that:

$$\begin{aligned} \Delta X &= m(\dot{U} + QW - VR) \\ \Delta Y &= m(\dot{V} + UR - PW) \\ \Delta Z &= m(\dot{W} + VP - UQ) \end{aligned} \quad (3.11)$$

3.2.2 Rotational motion

From Newton's second law it can be deduced that:

$$\vec{M} = \vec{M}_0 + \Delta\vec{M} = m \frac{d}{dt}(\vec{H}) \quad (3.12)$$

Where \vec{M} represents the sum of all applied torques, and \vec{H} is the angular momentum. The subscript 0 denotes the equilibrium component while the Δ represents the component to the perturbation. The axis system being used is the Earth frame. By definition, equilibrium flight must be unaccelerated flight along a straight path; during this flight the linear velocity vector relative to fixed space is invariant, and the angular velocity is zero. Thus \vec{M}_0 is zero.

For a rigid body, angular momentum may be defined as:

$$\vec{H} = \vec{I}\vec{\omega} \quad (3.13)$$

The inertia matrix \vec{I} is defined as:

$$\vec{I} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{yz} \\ -I_{xz} & -I_{yz} & I_{zz} \end{bmatrix} \quad (3.14)$$

The rate of change of H relative to the Earth axis system is given by:

$$\vec{M} = \frac{d}{dt}\vec{H} + \vec{\omega} \times \vec{H} \quad (3.15)$$

Transforming from body axes to the Earth axis system allows equation 3.15 to be re-expressed as:

$$\vec{M} = \vec{I} \left(\frac{d}{dt}\vec{\omega} + \vec{\omega} \times \vec{\omega} \right) + \vec{\omega} \times \vec{H} \quad (3.16)$$

However $\vec{\omega} \times \vec{\omega} = \vec{0}$. Expressing the vectors in the body frame axis system, vectors can be written as the sum of their corresponding components, as follows:

$$\vec{\omega} = \vec{i}P + \vec{j}Q + \vec{k}R \quad (3.17)$$

$$\frac{d}{dt}\vec{\omega} = \vec{i}\dot{P} + \vec{j}\dot{Q} + \vec{k}\dot{R} \quad (3.18)$$

$$\Delta\vec{M} = \vec{i}\Delta L + \vec{j}\Delta M + \vec{k}\Delta N \quad (3.19)$$

Which leads us to the following results:

$$\Delta L = I_{xx}\dot{P} - I_{xz}(\dot{R} + PQ) + (I_{zz} - I_{yy})QR \quad (3.20)$$

$$\Delta M = I_{yy}\dot{Q} - I_{xz}(P^2 - R^2) + (I_{zz} - I_{xx})PR \quad (3.21)$$

$$\Delta N = I_{zz}\dot{R} - I_{xz}\dot{P} + PQ(I_{yy} - I_{xx}) + I_{xz}QR \quad (3.22)$$

ΔL , ΔM and ΔN are the torques around the roll, pitch and yaw axes respectively.

The third term on the right hand side of equations 3.20 to 3.22 is a non-linear inertial coupling term. Mclean (1990) cites that, for large aircraft, such as transports, which cannot generate large angular rates, these terms are frequently neglected so that the moment equations become:

$$\Delta L = I_{xx}\dot{P} - I_{xz}(\dot{R} + PQ) \quad (3.23)$$

$$\Delta M = I_{yy}\dot{Q} - I_{xz}(P^2 - R^2) \quad (3.24)$$

$$\Delta N = I_{zz}\dot{R} - I_{xz}(\dot{P} - QR) \quad (3.25)$$

3.2.3 Contribution of the forces due to gravity

The forces due to gravity are always present in an aircraft. It can be properly assumed that gravity acts at the centre of gravity of the aircraft. When the centre of mass and the center of gravity coincide in an aircraft there is no external moment produced by the gravity about the center of gravity. Hence, for the body frame, gravity contributes only to the external force vector.

The gravity vector, $m\vec{g}$, is directed along the Z axis of the Earth frame. We can express the gravity vector as the sum of its components in the Body frame.

The Euler angles presented earlier can be used to pass from the Earth frame to the Body frame, and thus express the gravity force in the body frame as follows:

$$\delta X = -m\vec{g} \sin \Theta \quad (3.26)$$

$$\delta Y = m\vec{g} \cos \Theta \sin \Phi \quad (3.27)$$

$$\delta Z = m\vec{g} \cos \Theta \cos \Phi \quad (3.28)$$

3.2.4 Linearization of the Inertial and gravitational terms

Inertial forces and the contribution of the gravitational forces on the aircraft may be conveniently combined into components to represent the accelerations and forces along the Body frame axes.

The external forces \vec{F} acting on the aircraft can be re-expressed as:

$$\vec{F} = \vec{i}X + \vec{j}Y + \vec{k}Z \quad (3.29)$$

with

$$\begin{aligned} X &= \Delta X + \delta X \\ Y &= \Delta Y + \delta Y \\ Z &= \Delta Z + \delta Z \end{aligned} \quad (3.30)$$

The first term on the right hand side represent the aerodynamic and thrust forces while the second term is the gravitational term.

For notational convenience $\Delta L, \Delta M$ and ΔN are now denoted L, M and N . Thus the equations of motion of the rigid body, for its six degree of freedom, may be expressed as:

$$X = ma_{x_{cg}} = m(\dot{U} + QW - VR + g \sin \Theta) \quad (3.31)$$

$$Y = ma_{y_{cg}} = m(\dot{V} + UR - PW - g \cos \Theta \sin \Phi) \quad (3.32)$$

$$Z = ma_{z_{cg}} = m(\dot{W} + VP - UQ - g \cos \Theta \cos \Phi) \quad (3.33)$$

$$L = I_{xx}\dot{P} - I_{xz}(\dot{R} + PQ) + (I_{zz} - I_{yy})QR \quad (3.34)$$

$$M = I_{yy}\dot{Q} - I_{xz}(P^2 - R^2) + (I_{xz} - I_{zz})PR \quad (3.35)$$

$$N = I_{zz}\dot{R} - I_{xz}\dot{P} + PQ(I_{yy} - I_{xx}) + I_{xz}QR \quad (3.36)$$

where $a_{x_{cg}}, a_{y_{cg}}, a_{z_{cg}}$ are the components of the acceleration along the Body frame axes.

The presented equations are non-linear. Some simplification is possible, and the linearization of these equations can be done using the Taylor series. We consider the aircraft to comprise two components: a mean motion which represents the equilibrium, or trim, and a dynamic motion which accounts for the perturbations about the mean motion. We assume that the perturbations are small. Thus every motion variable is considered to have two components, for example:

$$U = U_0 + u \quad (3.37)$$

The trim values are denoted by a subscript 0 and the small perturbation values of a variable are denoted by the lower case letter.

At trim conditions there is no acceleration, and thus we may write:

$$\begin{aligned} X_0 &= m(Q_0 W_0 - V_0 R_0 + g \sin \Theta_0) \\ Y_0 &= m(U_0 R_0 - P_0 W_0 - g \cos \Theta_0 \sin \Phi_0) \\ Z_0 &= m(V_0 P_0 - U_0 Q_0 - g \cos \Theta_0 \cos \Phi_0) \\ L_0 &= (I_{zz} - I_{yy})Q_0 R_0 - I_{xz}(P_0 Q_0) \\ M_0 &= I_{xz}(P_0^2 - R_0^2) + (I_{xx} - I_{zz})P_0 R_0 \\ N_0 &= P_0 Q_0(I_{yy} - I_{xx}) + I_{xz}Q_0 R_0 \end{aligned} \quad (3.38)$$

The perturbed equations can be found by substituting eq. 3.37 into equations. 3.31 to 3.36, expanding the terms then subtracting the set of equations 3.38. Considering the perturbations from the mean conditions are small, the sines and cosines can be approximated to the angles themselves and the value unity, respectively. Moreover, the products and squares of the perturbed quantities are negligible.

Other assumptions and approximations may be applied. We may consider flight cases with simpler trim conditions, a case of great interest being, for example, when an aircraft has been trimmed to fly straight in steady, symmetric flight, with its wings level. This leads to the following assumptions:

- Straight flight implies $\dot{\Psi}_0 = \Theta_0 = 0$
- Symmetric flight implies $\Psi_0 = V_0 = 0$

- Flying with wings level implies $\Phi_0 = 0$
- Within these cruise flight conditions, we may also add the assumption $Q_0, P_0, R_0, V_0=0$

Therefore, the simplified equations of motion can now be written as follows:

$$\begin{aligned}
 x &= m(\dot{u} + qW_0 - g\theta \cos \Theta_0) \\
 z &= m(\dot{w} - qU_0 + g\theta \sin \Theta_0) \\
 m_1 &= I_{yy}\dot{q}
 \end{aligned} \tag{3.39}$$

$$\begin{aligned}
 y &= m(\dot{v} + rU_0 - pW_0 - g\phi \cos \Theta_0) \\
 l &= I_{xx}\dot{p} - I_{xz}\dot{r} \\
 n &= I_{zz}\dot{r} - I_{xz}\dot{p}
 \end{aligned} \tag{3.40}$$

The equations not only have been simplified, but they can also be separated into two distinct groups. The first group depends on u, w, q and θ and these are confined to the longitudinal plane. Similarly, the second group depends on the variables confined to the lateral plane r, v, p and ϕ .

3.3 Linearization of the longitudinal equations of motion

To expand the left hand side of the equations of motion, a series of Taylor is used about the trimmed flight conditions. Thus for example:

$$z = \frac{\partial Z}{\partial u}u + \frac{\partial Z}{\partial \dot{u}}\dot{u} + \frac{\partial Z}{\partial w}w + \frac{\partial Z}{\partial \dot{w}}\dot{w} + \frac{\partial Z}{\partial q}q + \frac{\partial Z}{\partial \dot{q}}\dot{q} + \frac{\partial Z}{\partial \delta_E}\delta_E + \frac{\partial Z}{\partial \dot{\delta}_E}\dot{\delta}_E + \frac{\partial Z}{\partial \delta_T}\delta_T + \frac{\partial Z}{\partial \dot{\delta}_T}\dot{\delta}_T + \dots \tag{3.41}$$

In the presented equation, the perturbed force z has a contribution from two control surfaces, the elevator, δ_E , and thrust variation, δ_T . However, if any other control surface on the aircraft being considered were involved, additional terms would be used.

As an example, applying the previous decomposition to the following equation:

$$x = m(\dot{u} + qW_0 - g\theta \cos\Theta_0) \quad (3.42)$$

Results in:

$$\begin{aligned} x &= \frac{\partial Z}{\partial u} u + \frac{\partial Z}{\partial \dot{u}} \dot{u} + \frac{\partial Z}{\partial w} w + \frac{\partial Z}{\partial \dot{w}} \dot{w} + \frac{\partial Z}{\partial q} q + \frac{\partial Z}{\partial \dot{q}} \dot{q} + \frac{\partial Z}{\partial \delta_E} \delta_E + \frac{\partial Z}{\partial \dot{\delta}_E} \dot{\delta}_E + \frac{\partial Z}{\partial \delta_T} \delta_T + \frac{\partial Z}{\partial \dot{\delta}_T} \dot{\delta}_T \\ &= m(\dot{u} + qW_0 - g\theta \cos\Theta_0) \end{aligned} \quad (3.43)$$

To simplify the notation it is customary to make the following substitutions:

$$X_x = \frac{1}{m} \frac{\partial X}{\partial z} \quad (3.44)$$

$$Z_x = \frac{1}{m} \frac{\partial Z}{\partial z} \quad (3.45)$$

$$M_x = \frac{1}{I_{yy}} \frac{\partial M}{\partial z} \quad (3.46)$$

When this substitution is made, these coefficients are referred to as the stability derivatives. From studying the aerodynamic data of a large number of aircrafts, it becomes evident that not every stability derivative is significant and, frequently, a number can be neglected. Without loss of generality it can be assumed that the following stability derivatives are often insignificant, and may be ignored: $X_{\dot{u}}, X_{\dot{w}}, X_q, X_{\dot{q}}, Z_q, Z_{\dot{w}}, X_{\dot{\delta}_E}, X_{\dot{\delta}_T}, Z_{\dot{\delta}_E}, Z_{\dot{\delta}_T}, M_{\dot{\delta}_E}, M_{\dot{\delta}_T}$

We can now rewrite the equations in the following form:

$$\begin{aligned}
 \dot{u} &= -W_0 q - \theta g \cos \Theta_0 + X_u u + X_w w + X_\alpha \delta e + X_\delta \delta t \\
 \dot{w} &= U_0 q - \theta g \sin \Theta_0 + Z_u u + Z_w w + Z_\alpha \delta e + Z_\delta \delta t \\
 \dot{q} &= M_u u + M_w w + M_\omega \dot{\omega} + M_q q + M_\alpha \delta e + M_\delta \delta t \\
 \dot{\theta} &= q
 \end{aligned} \tag{3.47}$$

In the stability frame, the orientation causes W_0 to be zero. Also, $\Theta_0 = \gamma_0 + \alpha_0$, where α_0 is zero cause this frame is oriented such as the X axis is in the wind direction. We can therefore write:

$$\begin{aligned}
 \dot{u} &= -\theta g \cos \gamma_0 + X_u u + X_w w + X_\alpha \delta e + X_\delta \delta t \\
 \dot{w} &= U_0 q - \theta g \sin \gamma_0 + Z_u u + Z_w w + Z_\alpha \delta e + Z_\delta \delta t \\
 \dot{q} &= M_u u + M_w w + M_\omega \dot{\omega} + M_q q + M_\alpha \delta e + M_\delta \delta t \\
 \dot{\theta} &= q
 \end{aligned} \tag{3.48}$$

3.4 Longitudinal state equations

A state equation is a first order, vector differential equation. It is a standard form in which to represent the equation of motion of an aircraft. Its most general expression is:

$$\dot{\vec{x}} = A\vec{x} + B\vec{u} \tag{3.49}$$

where \vec{x} is the state vector and \vec{u} the control vector. A is the state coefficient matrix and B the driving matrix.

If the state vector is defined as:

$$\bar{x} = \begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} \quad (3.50)$$

and if the aircraft is being controlled by the elevators and thrust, then:

$$\bar{u} = [\delta e \quad \delta r] \quad (3.51)$$

The linear equations of longitudinal motion found previously in equation 3.48 may easily be written under the state form, however the equation for \dot{q} is written as:

$$\dot{q} = M_u u + M_w w + M_{\dot{w}} \dot{w} + M_q q + M_{\delta e} \delta e + M_{\delta r} \delta r$$

It is obvious that a term in \dot{w} exists on the right hand side of the equation. The state equation does not admit on its right hand side terms involving the first derivatives of any of the state control variables. Fortunately, \dot{w} , itself, depends only upon \bar{x} and \bar{u} and therefore an easy substitution is possible.

Earlier in equation 3.47 we have written:

$$\dot{w} = U_0 q - \theta g \sin \gamma_0 + Z_u u + Z_w w + Z_{\delta e} \delta e + Z_{\delta r} \delta r$$

Substituting for w in the equation for \dot{q} yields:

$$\begin{aligned} \dot{q} = & (M_u + M_w Z_u) u + (M_w + M_w Z_w) w + (M_q + M_w U_0) q - (g M_w \sin \gamma_0) \theta \\ & + (M_{\delta e} + M_w Z_{\delta e}) \delta e + (M_{\delta r} + M_w Z_{\delta r}) \delta r \end{aligned} \quad (3.52)$$

which can now be rewritten as:

$$\dot{q} = \tilde{M}_u u + \tilde{M}_w w + \tilde{M}_q q + \tilde{M}_\theta \theta + \tilde{M}_\delta \delta e + \tilde{M}_\delta \delta \dot{e} \quad (3.53)$$

where $\tilde{M}_u = M_u + M_\psi Z_u$, $\tilde{M}_w = M_w + M_\psi Z_w$, $\tilde{M}_q = M_q + M_\psi U_0$, $\tilde{M}_\theta = -gM_\psi \sin \gamma_0$,

$\tilde{M}_\delta = M_\delta + M_\psi Z_\delta$ and $\tilde{M}_\delta = M_\delta + M_\psi Z_\delta$

And now, putting the equations in the state form, it is possible to write:

$$A = \begin{bmatrix} X_u & X_w & 0 & -g \cos \gamma_0 \\ Z_u & Z_w & U_0 & -g \sin \gamma_0 \\ \tilde{M}_u & \tilde{M}_w & \tilde{M}_q & \tilde{M}_\theta \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad (3.54)$$

And

$$B = \begin{bmatrix} X_\delta & X_\delta \\ Z_\delta & Z_\delta \\ \tilde{M}_\delta & \tilde{M}_\delta \\ 0 & 0 \end{bmatrix} \quad (3.55)$$

3.5 Longitudinal stability

The dynamic stability of the perturbed longitudinal model is most effectively determined by the eigenvalues of coefficient matrix A . They can be found by solving the linear equation:

$$|\lambda I - A| = 0 \quad (3.56)$$

In fact, as mentioned by Mclean (1990), an aircraft may be said to be dynamically stable if all its eigenvalues, which are composed of real and imaginary parts, have their real part negative. Zero or positive real parts will be translated into the fact the aircraft will be dynamically unstable.

It has been observed that for the majority of aircraft types, the quadratic equation used to determine the eigenvalues, invariably factorizes into two quadratic factors in the following manner:

$$(\lambda^2 + 2\zeta_{ph}\omega_{ph}\lambda + \omega_{ph}^2)(\lambda^2 + 2\zeta_{sp}\omega_{sp}\lambda + \omega_{sp}^2) \quad (3.57)$$

The first factor corresponds to a mode which is characterized by an oscillation of long period. The damping of this mode is usually very low, and may be negative, so that the mode is unstable and the oscillations grow with time. The low frequency associated with the long period motion is defined as the natural frequency ω_{ph} ; the damping ratio has been denoted ζ_{ph} . The mode is referred to as the phugoid mode. The second factor corresponds to a rapid, relatively well-damped motion associated with the short period mode whose frequency is ω_{sp} and damping is ζ_{sp} .

3.6 Linear state space model

We shall now determine, in the same state-space model form presented earlier, the linear model of the aircraft to which our controller will be applied. We shall also study the characteristics of that specific linear model.

The Dynamic Test Bed used for the test and implementation of our control designs uses FlightSim (refer to section 7.1 for further details) as the platform for the aircraft model and simulation. Within that environment, a Boeing 747-400 aircraft has been modeled and is used as the aircraft model for the simulation and integration of the DTB. The “FlsimLinearization” script program provided with Flightsim provides an algorithm that generates the linear equations model of the simulation at a desired trim position. This file is provided to help design the aircraft control system. The FlsimLinearization program considers that the state-space can be broken down into two independent modes, Longitudinal and Lateral. For both of these, FlsimLinearization generates the A and B matrices.

We are only interested in the longitudinal model of the aircraft. Linearizing the model at cruise conditions, an altitude of 12000 meters and a True Airspeed of 466 knots, we obtain the following A and B matrices for the longitudinal state space model:

$$A = \begin{bmatrix} -0.306613 & -0.000529565 & -0.00330512 & -0.000621396 \\ -0.531057 & -0.00674625 & 0.0195349 & -9.83466 \\ 237.123 & -0.12062 & -0.664316 & -0.0195343 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.120458 & 0.0131793 \\ -0.0178595 & 2.09909 \\ 1.11696 & -0.0916484 \\ 0 & 0 \end{bmatrix}$$

Matrices and other linear models at various other conditions have been generated and are presented in Annex I.

It is important to note that in the state space model equation 3.49: $\dot{\bar{x}} = A\bar{x} + B\bar{u}$, \bar{x} defines the vector of states and \bar{u} the controls. In the Flsimlinearization algorithm, the definition of \bar{x} differs from the definition presented in equation 3.50, where now \bar{x} is defined as:

$$\bar{x} = \begin{bmatrix} q \\ u \\ w \\ \theta \end{bmatrix}$$

Matrices A and B are defined accordingly.

3.7 Open loop characteristics

The study of the open loop characteristics helps us further understand the dynamic behaviour of our system. The open loop model characteristics for the longitudinal aircraft model are presented. In the next sections, the dynamic stability of the system with respect to the speed or altitude of the aircraft is studied. We shall also study the effect of thrust variation on the static stability.

3.7.1 Eigenvalues of the longitudinal movement

Considering the linear model presented earlier, given by matrices A and B, a linearization at trim point of 12000 meters of altitude and airspeed of 466 knots gives us matrices:

$$A = \begin{bmatrix} -0.306613 & -0.000529565 & -0.00330512 & -0.000621396 \\ -0.531057 & -0.00674625 & 0.0195349 & -9.83466 \\ 237.123 & -0.12062 & -0.664316 & -0.0195343 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.120458 & 0.0131793 \\ -0.0178595 & 2.09909 \\ 1.11696 & -0.0916484 \\ 0 & 0 \end{bmatrix}$$

The natural longitudinal movement can be analysed using the system eigenvalues. This is achieved using equation 3.58:

$$|\lambda I - A| = 0 \quad (3.58)$$

Using Matlab, the eigenvalues of matrix A are calculated:

Table 3.1
System Eigenvalues

Eigenvalue	Damping	Freq. (rad/s)
$-2.29e-003 + 2.15e-002i$	1.06e-001	2.16e-002
$-2.29e-003 - 2.15e-002i$	1.06e-001	2.16e-002
$-4.87e-001 + 8.69e-001i$	4.89e-001	9.96e-001
$-4.87e-001 - 8.69e-001i$	4.89e-001	9.96e-001

In general, all civil aircrafts are statically stable, and the longitudinal mode can be represented by two conjugate modes: the short-period mode and the phugoid mode. The first 2 eigenvalues correspond to the phugoid mode, while the last two correspond to the short-period mode. Looking at the eigenvalues we notice that all the real components are negative. The system can therefore be considered stable in open loop. However, the phugoid mode roots are very close to the origin of the root map, and leads us to expect a very slow dynamic. Using the eigenvalues found, it is possible to determine the following parameters: half-time response t_{half} , oscillation period T , natural frequency w_n , and damping coefficient ζ . These parameters have an important role in characterizing the dynamic behaviour of our system. They can be calculated using the equations presented by Mclean (1990) :

$$\lambda_i = n_i + jw_i \quad (3.59)$$

$$t_{half} = \frac{0,693}{|n_i|} \quad (3.60)$$

$$T = \frac{2\pi}{w_i} \quad (3.61)$$

$$w_n = \sqrt{n_i^2 + w_i^2} \quad (3.62)$$

$$\zeta = \frac{-n_i}{\sqrt{n_i^2 + w_i^2}} \quad (3.63)$$

Results are presented in Table 3.2:

Table 3.2
System Eigenvalues and Parameters

Eigenvalue	t_{half} (sec)	T (sec)	w_n (rad/sec)	ζ
$-0.00229 \pm j 0.0215$	302.62	290.88	0.0216	0.106
$-0.487 \pm j 0.869$	1.423	6.3084	0.996	0.489

Looking at Table 3.2, we notice the following two modes and their characteristics:

- Short-period mode: corresponding to the eigenvalues $-0.487 \pm j 0.869$, this mode presents a small t_{half} , which indicates that the mode is fast, however presents some oscillations that are quickly damped.
- Phugoid mode: corresponding to the eigenvalues $-0.00229 \pm j 0.0215$, this mode presents a very important t_{half} , and a very small damping ratio. The result is an important slow oscillation that is poorly damped.

We are mainly interested by the speed response in respect to the thrust control. From the state space model, we can write the following transfer function of $\frac{u}{\delta}$:

$$\frac{u}{\delta} = \frac{2.099s^3 + 2.029s^2 + 2s - 0.08836}{s^4 + 0.9777s^3 + 0.9967s^2 + 0.005006s + 0.0004645} \quad (3.64)$$

The root locus graphic is presented in Figure 3.6:

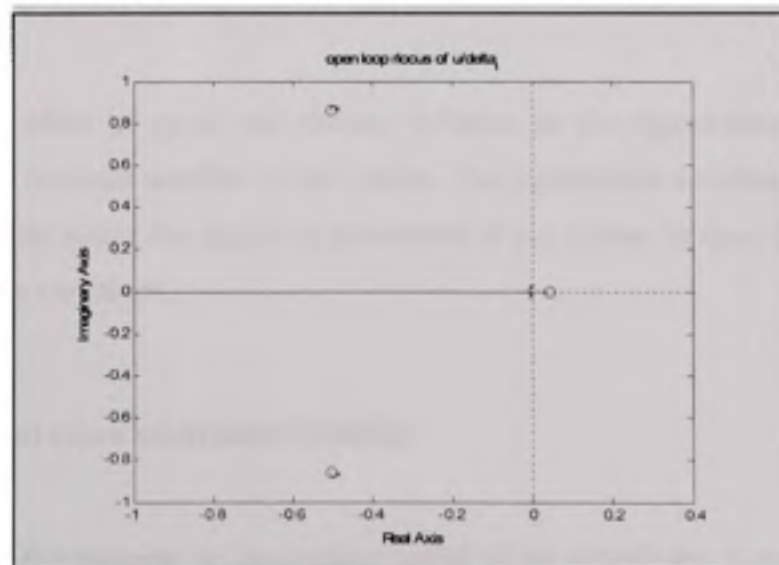


Figure 3.6 System root map.

The open loop response in the time domain shows a settling time of around 2500 seconds, as presented in Figure 3.7:

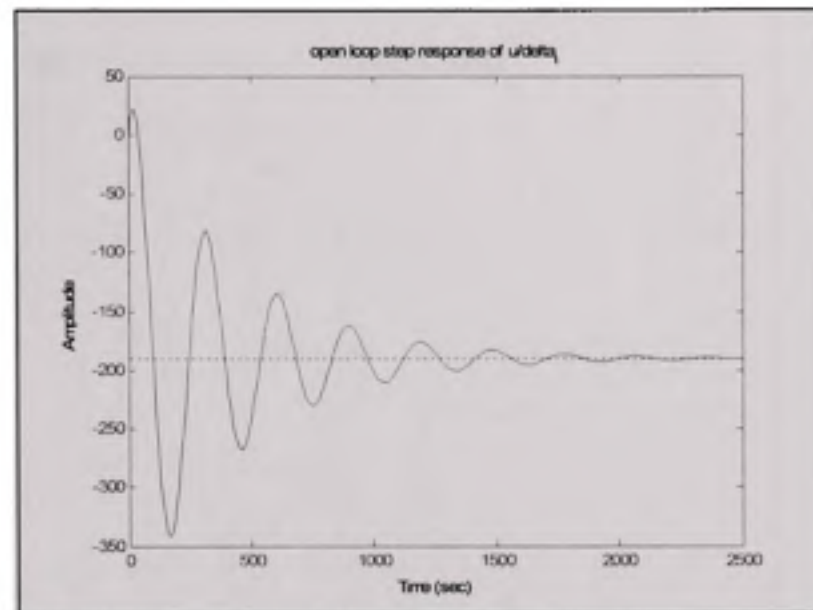


Figure 3.7 Open loop time response.

3.7.2 Effect of speed and altitude variation on dynamic stability

By studying the effect of speed and altitude variation on the eigenvalues, we will have a better idea of the dynamic stability of our system. The eigenvalues variation in the open loop system will help us notice the degree of robustness of our system in open loop during small speed and altitude variations.

3.7.2.1 Effect of speed on dynamic stability

The variation of the airspeed or longitudinal speed of an aircraft has a major effect on the time response of the longitudinal movement. Figure 3.8 presents the effect of speed variation on the eigenvalues of our system:

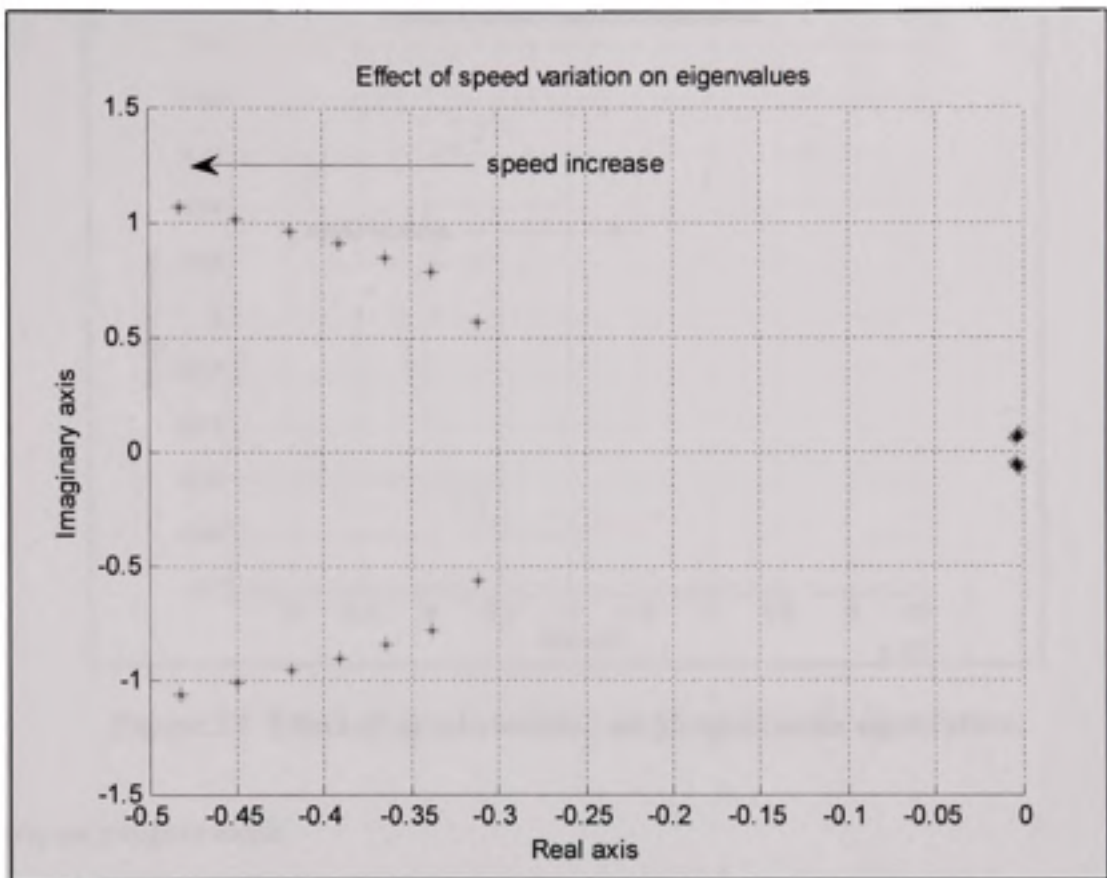


Figure 3.8 Effect of speed variation on eigenvalues.

Effects on short-period mode

As speed increases over the envelope of flight, the eigenvalues move to the negative part of the Real axis, hence into the stability area. These eigenvalues are always stable. The imaginary part of the eigenvalues is also more important as speed increases. The oscillation frequency will be more important as speed increases, and the period shorter.

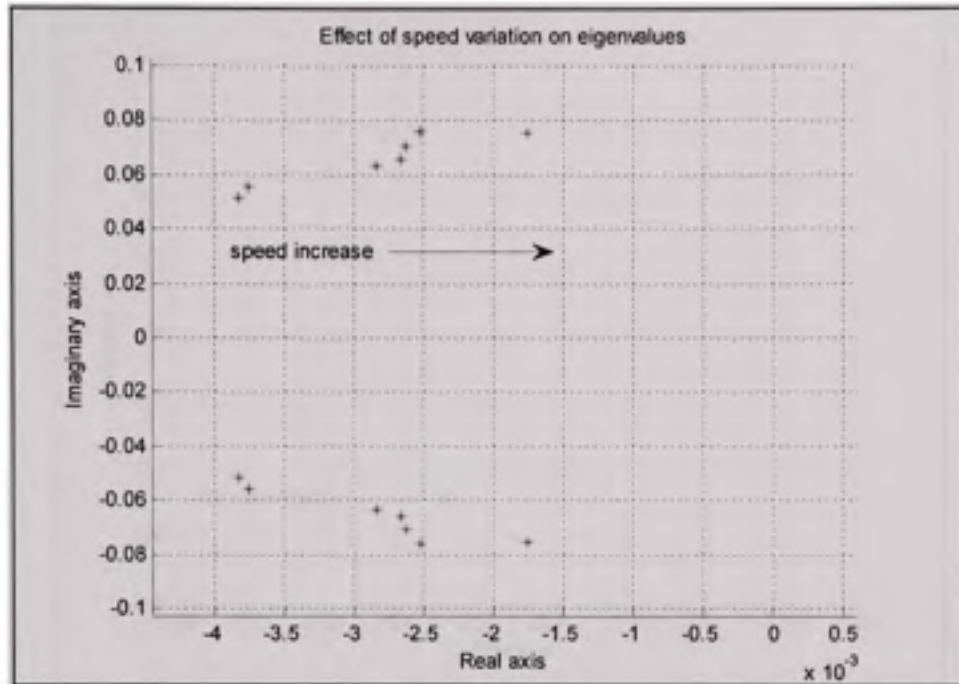


Figure 3.9 Effect of speed variation on phugoid mode eigenvalues.

Effects on phugoid mode

Figure 3.9 shows us a closer look at the eigenvalues corresponding to the phugoid mode. We notice the eigenvalues moving towards the positive side of the real axis; however they never reach instability while we are in the flight envelope. The effect of speed variation is very small on eigenvalues, but we still notice the imaginary part increasing as the real part of the eigenvalues decreases, which will result into a smaller damping ratio as speed increases.

3.7.2.2 Effect of altitude on dynamic stability

The variation of the altitude also has an important effect on the time response of the longitudinal movement. Figure 3.10 presents the effect of altitude variation on the eigenvalues of our system:

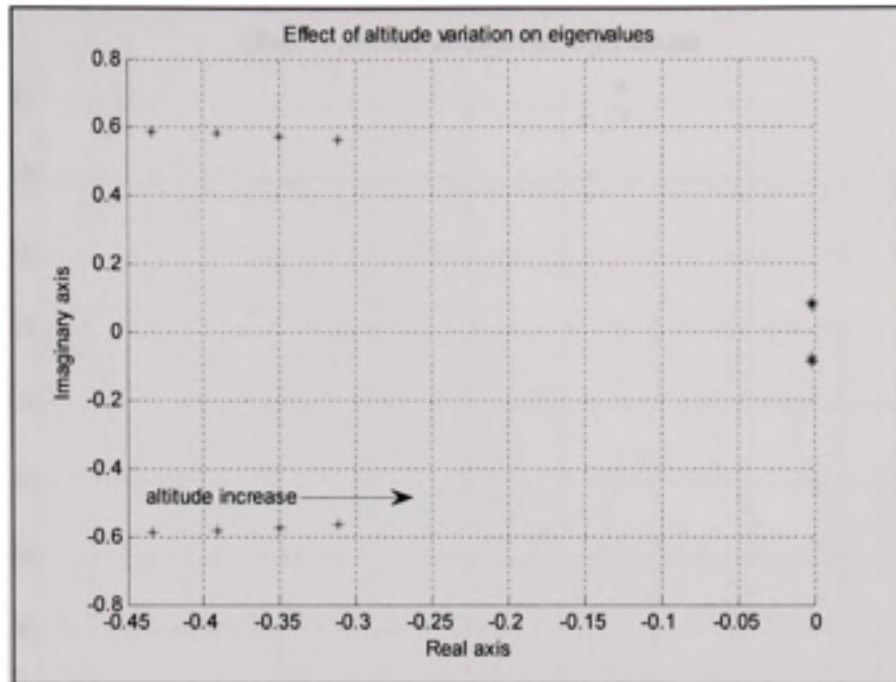


Figure 3.10 Effect of altitude variation on eigenvalues.

Effects on short-period mode:

As altitude increases over the envelope of flight, the eigenvalues move in the positive direction of the Real axis, however they remain in the negative part, the stability area. These eigenvalues are always stable. The imaginary part of the eigenvalues doesn't vary much, while the real part decreases. The oscillation frequency will be more important as altitude increases.

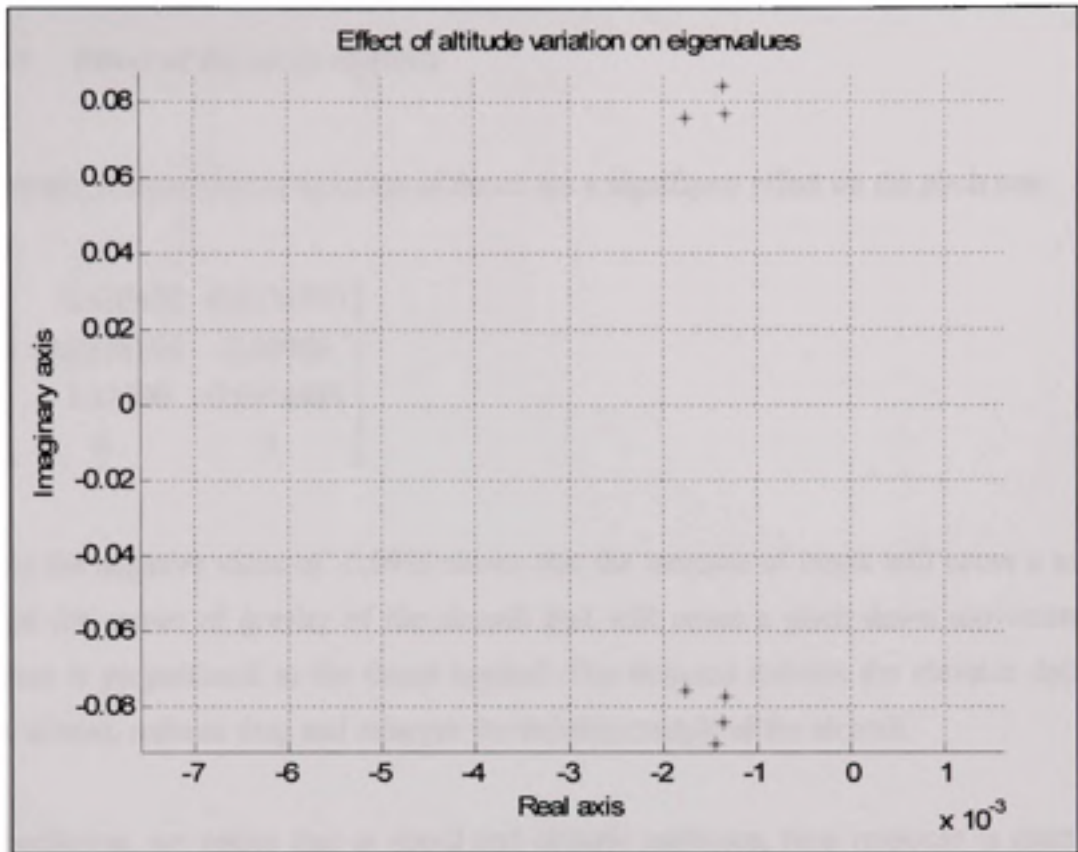


Figure 3.11 Effect of altitude variation on phugoid mode eigenvalues.

Effects on phugoid mode:

Figure 3.11 shows us a closer look at the eigenvalues corresponding to the phugoid mode. The effect of altitude variation is very small on eigenvalues. It is important to note that eigenvalues remain stable over the flight envelope.

3.7.2.3 Effect of thrust on stability

The matrix B shows that a variation of thrust has a significant effect on the pitch rate:

$$B = \begin{bmatrix} 0.120458 & 0.0131793 \\ -0.0178595 & 2.09909 \\ 1.11696 & -0.0916484 \\ 0 & 0 \end{bmatrix}$$

In fact the negative value of -0.0916 shows that the increase of thrust will cause a moment around the center of gravity of the aircraft that will cause a pitch down movement. The moment is proportional to the thrust applied. The moment reduces the elevator deflection angle at trim, reduces drag and enlarges the stability margin of the aircraft.

In conclusion, we notice that as speed and altitude increases, time response is shorter and oscillations are more important. We will therefore use the linearized model at high speed and high altitude as a reference for our study to present the overall benefits of the proposed controller. The controller will then be further developed for the entire flight envelope. We also notice the effect of thrust on pitch variation, and the importance of coupling the pitch and thrust channels to eliminate any unwanted altitude variation due to thrust activity. This is further elaborated in the next section and considered in chapters 4 and 5 in the controller design.

3.7.3 Thrust and pitch coupling at trim conditions

It is possible to determine a relation between pitch and thrust around a trim point, and thus predicting the effects a pitch or thrust variation would have on the other control output

Oshe'a J. (2000) shows that it is possible to evaluate the thrust variation effect on pitch by studying the system around a certain trim point.

In fact, two outputs are of interest, the airspeed and the climb rate. Inputs are δt (thrust) and δe (elevator's angle).

At the trim conditions we have:

$$\bar{x}' = \bar{0} = A\bar{x}_e + B\bar{u}_e \quad (3.65)$$

$$\bar{y}_e = C\bar{x}_e + D\bar{u}_e \quad (3.66)$$

We can write:

$$\begin{bmatrix} \bar{0} \\ \bar{y}_e \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \bar{x}_e \\ \bar{u}_e \end{bmatrix} \quad (3.67)$$

If a solution exists, then it must satisfy:

$$\begin{bmatrix} \bar{x}_e \\ \bar{u}_e \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} \begin{bmatrix} \bar{0} \\ I \end{bmatrix} \bar{y}_e \quad (3.68)$$

We use the notation:

$$\begin{bmatrix} \bar{x}_e \\ \bar{u}_e \end{bmatrix} = \begin{bmatrix} N_{xy} \\ N_{wy} \end{bmatrix} \bar{y}_e \quad (3.69)$$

With

$$\begin{bmatrix} N_{xy} \\ N_{wy} \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} \begin{bmatrix} \bar{0} \\ I \end{bmatrix} \quad (3.70)$$

For example, if :

$$\bar{x} = \begin{bmatrix} u \\ w \\ q \\ \theta \end{bmatrix} \quad (3.71)$$

And :

$$\bar{u} = \begin{bmatrix} \delta_t \\ \delta_e \end{bmatrix} \quad (3.72)$$

And we are interested in the airspeed u and climb rate h' (which is a function of w and θ) as outputs, then we have

$$\bar{y} = \begin{bmatrix} u \\ h' \end{bmatrix} = C\bar{x} \quad (3.73)$$

with C chosen accordingly.

We then calculate N_{xy} and N_{uy} , and we may now establish the relation given by:

$$\begin{bmatrix} \delta_t \\ \delta_e \end{bmatrix} = N_{xy} \begin{bmatrix} u_e \\ h_e' \end{bmatrix} \quad (3.74)$$

Using equation 3.74, we can observe and evaluate the influence of δt variation on the climb rate around this point of stability. We can also, inversely, evaluate for a certain climb rate, the commands needed. Since the thrust command is known, we can then evaluate the elevator deflection needed to bring the climb rate to zero.

Such calculations must be done prior to the implementation of the control loop. A linearized model of the aircraft will be obtained for several regions of the envelope. Once this model is obtained, Matrices A, B, C and D will be available for each value of the speed and altitude of the aircraft. We can then obtain using a simple computation the elevator's predicted estimated variation, using speed and thrust measures. This predicted elevator's deflection will then be sent to the pitch loop to compensate the effect of speed variation.

A prediction term will only be sent if the aircraft is in cruise mode, since it is essentially the main mode in which the autothrottle controls airspeed, and in which the altitude variation is unwanted.

Another mean of pitch-thrust coupling can be achieved through coupled multi input multi output control, which will be presented in chapter 5.

3.8 Engine modelization

In the previous sections, we have presented the aircraft model that shall be used throughout this study. The outputs of the aircraft model are speed, pitch, pitch rate and vertical speed while the inputs are elevator deviation command and thrust command. Thrust input can be used to control the outputs of the model, however this model presents limitations to our application since no Thrust output is available for thrust control. A model of the engines should be available for thrust control design. Thrust control design shall be presented in the next chapters.

3.8.1 Turbofan multi-spool engines

Most modern jet engines are actually turbofans, where the low pressure compressor acts as a fan, supplying supercharged air not only to the engine core, but to a bypass duct. The bypass airflow either passes to a separate 'cold nozzle' or mixes with low pressure turbine exhaust gases, before expanding through a 'mixed flow nozzle'.

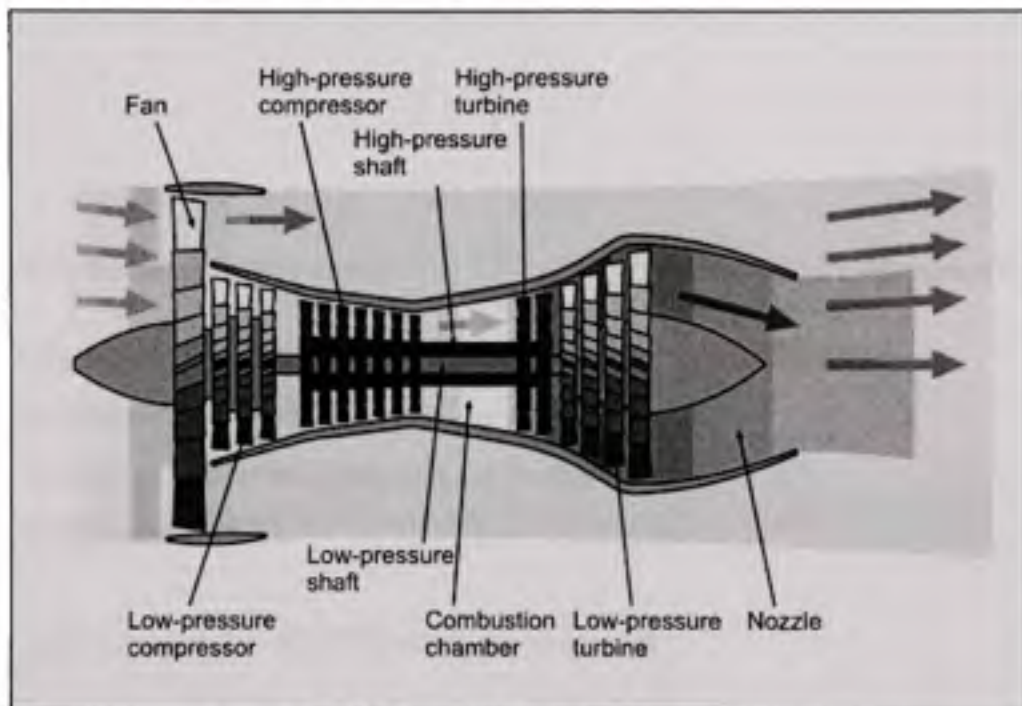


Figure 3.12 Turbofan multi-spool engine [25].

Jet engines have pressure sensors mounted at various places inside. As air travels through the compressor stages of an engine, its pressure increases. There are two probes, the front one called PT2 and the back one is PT7. PT is Total pressure and the number is the engine station (2 being the front of the first stage compressor/fan and 7 being just behind the last turbine stage). These pressures are brought into the EPR gauge and compared to produce a reading. The engine pressure ratio (EPR) is a parameter which provides an indication of total thrust

output during both steady state and non steady state operation which is sufficiently accurate for most purposes.

3.8.2 Engine Model

In steady flight, the thrust of an engine is given by:

$$T = W \left(\frac{D}{L} \right) \quad (3.75)$$

Where W is the weight of the aircraft and $\left(\frac{L}{D} \right)$ is the Lift-Drag ratio of the aircraft.

Considering the Boeing 747 case presented by Hanke & Nordwall (1970):

- Maximum Thrust=60000 pounds per engine
- Weight=875000 pound (gross take-off weight)
- Lift/Drag ratio=17.5

Since 1 pound force = 4.4482216 Newton, we may write:

- $T_{MAX} = 4 \times 6000 = 240000 \text{ pound} = 1067573.184 \text{ Newton}$
- $W = 875000 \text{ pound} = 3892193.9 \text{ Newton}$

And the excess Thrust can be expressed by (Mclean D., (1990)) :

$$T_e = T_{MAX} - W \left(\frac{D}{L} \right) \quad (3.76)$$

And so, calculating the excess thrust we get: $T_e = T_{MAX} - W\left(\frac{D}{L}\right) = 845345.92 \text{ Newton}$

If authority of engines is 10%, then allowed thrust to be changed is the excess thrust.

McLean D., (1990) cites that for large aircrafts, the time response of the engine varies between 0.3 and 1.5 seconds. Considering that the engine time response is 0.5 seconds, the engine model becomes:

$$\frac{T}{\delta T} = \frac{845345.92}{0.5s + 1} \quad (3.77)$$

With T in Newton, and input thrust command δT normalized between 0 and 1.

3.8.3 Engine Pressure Ratio

In controlling the thrust of a multi-spool turbofan engine and especially in causing selected changes in the thrust output of such an engine, it is important that a thrust control system provide both accuracy and stability. Providing acceptable accuracy and stability in such controls has been a problem of long standing in the art.

The accuracy and stability of thrust control is determined in large part by feedback signals from the engine and by treatment of these signals by the thrust control system. One approach to stabilizing the operation of a two spool turbofan engine has been to monitor the difference between the actual rotational speed of the high-speed spool (N1) and a selected rotational speed for the high-speed spool and then control fuel flow to the engine in response to this difference. However, it is known that certain engine pressure-related parameters are more directly related to this difference, and are much better indicators of engine thrust than is the speed of the high-speed spool. It is therefore more advantageous, in terms of accuracy, to

utilize engine parameters such as engine pressure ratio (EPR) to control engine operation. EPR is the ratio of the total pressure of the exhaust flow (static + dynamic) to the same pressure in the engine inlet for a gas turbine. It is proportional to the thrust and therefore a good indicator of engine output. 1.0 is essentially zero thrust. 1.8 is fairly high thrust, depending on the engine.

Unfortunately, Larson & Stephan (1980) cite that in a multi-spool turbofan engine there is more inertia associated with engine pressure parameters such as EPR than there is with the speed of the high speed spool, so that although accuracy is improved with the use of parameters such as EPR there is also some loss of stability. In addition the pressure signals used to determine values of the relevant pressure related parameters usually include an element of noise which must be filtered out, and the filtering step further delays response time and have a negative impact on thrust control stability.

In order to modelize the EPR sensor, we shall elaborate the relation between Thrust and EPR. Hanke & Nordwall (1970) as well as Robbins & Simpson (1991) cite that the relationship between EPR and an equivalent throttle rate command is a function of temperature as presented earlier in Figure 1.3.

The curve between two points A and B for a selected temperature can be approximated by a straight line. From this approximation, the slope of the line may be computed, and a linear relation between EPR and Thrust variation can be found, as presented earlier in equation 1.1, in chapter 1.

The variation of the slope compared to the change in temperature is small (Robbins & Simpson (1991)), and thus a constant value may be chosen for use at all reasonable temperatures. The control achieves then the required EPR level by commanding a throttle rate proportional to the EPR error.

According to Hanke & Nordwall (1970) the value of K can be considered linear in two zones for the Boeing 747. At low thrust, K is given by approximately 90000 pound of Thrust per EPR and at high thrust 45000 pound of Thrust per EPR. Using the international system, K will be approximated to 400000 Newton per EPR at low thrust operation and 200000 Newton per EPR at high thrust operation. However since the Thrust modeled is the thrust of four engines, the EPR has to be measured accordingly. In fact every engine has its own EPR sensor, so if the thrust used to measure the EPR is the total thrust, the EPR measure should be divided by 4 to evaluate the desired EPR. Also, the thrust used in the model is only the excess thrust (10% of the thrust). EPR is used to measure total thrust and hence the EPR sensor gain has to be adjusted accordingly:

$$K_{EPR} = \frac{10}{4 \times K} \text{ EPR / Newton} \quad (3.78)$$

Since EPR sensors are basically pressure and temperature sensors, the time response is considerably slow and can be approximated to be around 2.5 seconds.

Hence, EPR sensor at high thrust is modeled by:

$$\left(\frac{EPR}{T} \right)_{HIGH} = \frac{1}{(80000)2.5s + 1} \quad (3.79)$$

And at low thrust modeled by:

$$\left(\frac{EPR}{T} \right)_{LOW} = \frac{1}{(160000)2.5s + 1} \quad (3.80)$$

It has to be noted that the Engine and EPR modelization are not based on the Flightsim aircraft model since no engine data is available from this model. Therefore any design using the engine and EPR sensor model will have to be fine-tuned upon integration within the Flightsim environment.

In order to elaborate the relation between EPR and the thrust lever angle, we add the engine model to the EPR sensor model to find the relation between EPR and δT :

$$\frac{EPR}{\delta T} = \frac{T}{\delta T} \times \frac{EPR}{T} = \frac{K_E}{T_E s + 1} \times \frac{K_{PRE}}{T_{EPR} s + 1} = \frac{845345.92}{0.5s + 1} \times \frac{1}{(80000)2.5s + 1} \quad (3.81)$$

Or at low thrust

$$\frac{EPR}{\delta T} = \frac{845345.92}{0.5s + 1} \times \frac{1}{(160000)2.5s + 1} \quad (3.82)$$

It has been noted that the engine model used within the Flightsim modelization of a Boeing 747 is not available. However, using a normal engine power up at ground conditions, it is possible to output the EPR and Throttle percentage during the engine power up. Using the 2 outputs, it is possible to present the EPR variation of the simulation model with respect to the throttle position percentage, as shown in Figure 3.13:

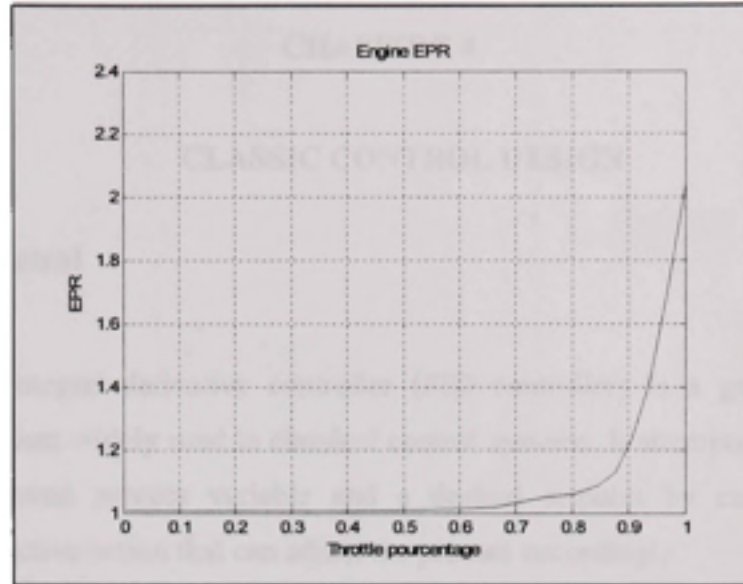


Figure 3.13 EPR-Throttle position relation.

The data collected from the simulation is used to build a lookup table used in the modelization of the EPR sensor model. The EPR sensor model is then used during the design process to determine the appropriate design for EPR control. The EPR control design is presented in the next chapter, section 4.2.

CHAPITRE 4

CLASSIC CONTROL DESIGN

4.1 PID control

A proportional–integral–derivative controller (PID controller) is a generic control loop feedback mechanism widely used in classical control systems. It attempts to correct the error between a measured process variable and a desired setpoint by calculating and then outputting a corrective action that can adjust the process accordingly.

The PID controller calculation (algorithm) involves three separate parameters; the Proportional, the Integral and Derivative values. The Proportional value determines the reaction to the current error, the Integral determines the reaction based on the sum of recent errors and the Derivative determines the reaction to the rate at which the error has been changing. The weighted sum of these three actions is used to adjust the process.

By "tuning" the three constants in the PID controller algorithm the PID can provide control action designed for specific process requirements. The response of the controller can be described in terms of the responsiveness of the controller to an error, the degree to which the controller overshoots the setpoint and the degree of system oscillation. Note that the use of the PID algorithm for control does not guarantee optimal control of the system or system stability.

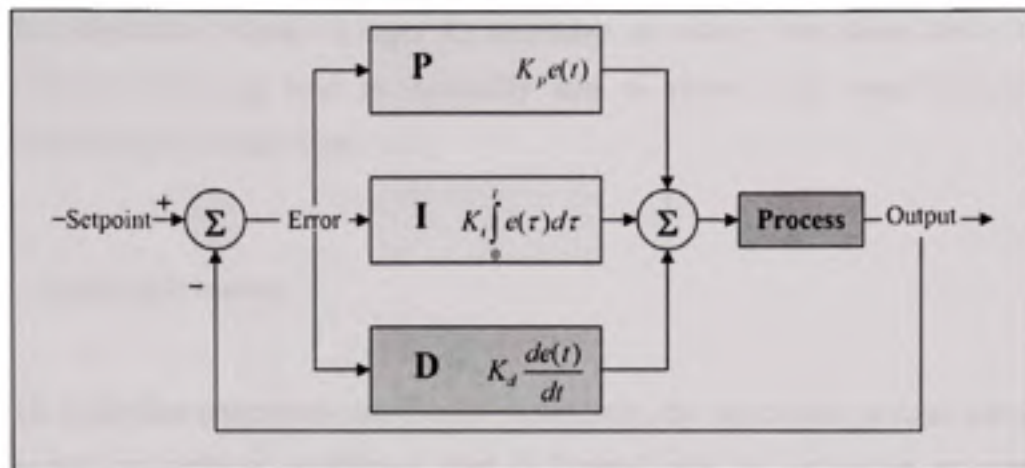


Figure 4.1 PID controller [34].

The output from the three terms, the proportional, the integral and the derivative terms are summed to calculate the output of the PID controller. Defining $u(t)$ as the controller output, the final form of the PID algorithm is:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de}{dt} \quad (4.1)$$

and the tuning parameters are:

- **K_p : Proportional Gain** - Larger K_p typically means faster response since the larger the error, the larger the Proportional term compensation. An excessively large proportional gain will lead to process instability and oscillation.
- **K_i : Integral Gain** - Larger K_i implies steady state errors are eliminated quicker. The trade-off is larger overshoot: any negative error integrated during transient response must be integrated away by positive error before we reach steady state.

- K_d : **Derivative Gain** - Larger K_d decreases overshoot, but slows down transient response and may lead to instability due to signal noise amplification in the differentiation of the error.

4.1.1 Loop gain tuning

If the PID controller parameters are chosen incorrectly, the controlled process input can be unstable, with or without oscillation, and is limited only by saturation or mechanical breakage. Tuning a control loop is the adjustment of its control parameters (gain/proportional band, integral gain/reset, derivative gain/rate) to the optimum values for the desired control response.

There are several methods for tuning a PID loop, as presented by Van& Doren (2003). The most effective methods generally involve the development of some form of process model, and then choosing K_p , K_d and K_i based on the dynamic model parameters. Manual tuning methods can be relatively inefficient. The best tuning method often involves subjecting the system to a step change in input, measuring the output as a function of time, and using this response to determine the control parameters. Numerical methods can be used to control the gains value selection to modify the output in order to match pre-selected criterias. For example, if gains are calculated to ensure a system response within predefined limits, a constraint envelope is defined to fix the upper and lower bounds of the controlled variable. Together with a number of tunable variables (K_i and K_p), a cost function is generated to consist of a weighted maximum constraint violation. At each iteration, each tunable variable is perturbed in turn and the resulting constraint values and cost function are evaluated. A gradient search direction is determined from these results and a line search along the gradient is performed in order to minimise the cost function while satisfying the constraint envelope criteria simultaneously. Once the system response is within the selected limits and the control system requirements are met, the gains are held and used within the control loop.

In case real-time application tuning is required, one tuning method is to first set the K_i and K_d gain values to zero. Increase the proportional gain until the output of the loop oscillates, and then the gain should be left set to be approximately half of that value for a "quarter amplitude decay" type response. Then increase derivative gain until any offset is correct in sufficient time for the process. However, too much derivative gain will cause instability. Finally, increase the integral term gain, if required, until the loop is acceptably quick to reach its reference after a load disturbance. However, too much integral gain will cause excessive response and overshoot. A fast PID loop tuning usually overshoots slightly to reach the setpoint more quickly; however, some systems cannot accept overshoot, in which case an "over-damped" closed-loop system is required, which will require a proportional gain setting significantly less than half that of the setting causing oscillation.

Table 4.1
Effects of PID parameters variation

Effects of <i>increasing</i> parameters				
Parameter	Rise Time	Overshoot	Settling Time	S.S. Error
K_p	Decrease	Increase	Small Change	Decrease
K_i	Decrease	Increase	Increase	Eliminate
K_d	Small Decrease	Decrease	Decrease	None

The algorithm used in this study will consist of two main steps:

- 1- Offline design: Using the numerical method presented earlier, and the linearized system model, the gains are calculated first in order to insure that the linearized model output behaviour is as desired.

- 2- Real-time tuning: After integration, the gains will be adjusted manually to overcome any desired effect introduced by the non-linear application of the system. The adjustment of the gains will take into account the effects presented in Table 4.1 and the real-time tuning method presented earlier.

4.2 Thrust control

The EPR control law achieves the commanded EPR level by commanding a throttle position proportional to the EPR error. The throttle position is given by multiplying the EPR error by a constant gain. An integral term will be added to remove steady state error. The output of the Proportional-Integral compensator is the desired Throttle position.

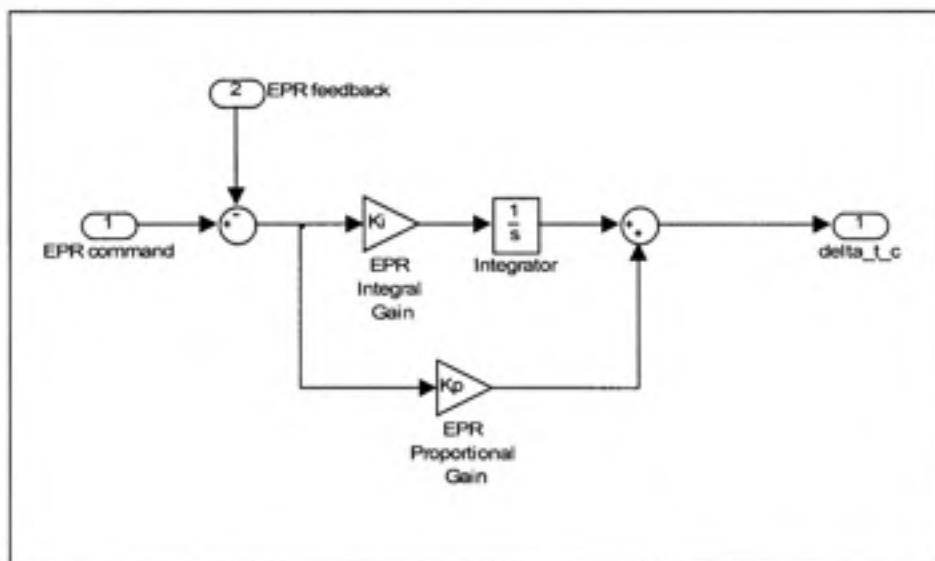


Figure 4.2 EPR Capture control channel.

Thus, the thrust command in EPR capture mode is given by the following equation:

$$dt_c = K_p (EPR_{ref} - EPR) + K_i \int (EPR_{ref} - EPR) dt \quad (4.2)$$

The selection of the gains can consider different regions of the non-linear EPR envelop. Using the Flightsim aircraft and engine model, the relationship between EPR and an equivalent throttle command can be found, as presented earlier in section 3.8.3, Figure 3.13.

Using the model built with the look up table from the data presented in Figure 3.13, the EPR control gains will be evaluated using a numerical optimisation method. Gains are calculated to ensure a system response within predefined limits. A constraint envelope is defined to fix the upper and lower bounds of the controlled variable. Together with a number of tunable variables (K_I and K_P), a cost function is generated to consist of a weighted maximum constraint violation. At each iteration, each tunable variable is perturbed in turn and the resulting constraint values and cost function are evaluated. A gradient search direction is determined from these results and a line search along the gradient is performed in order to minimise the cost function while satisfying the constraint envelope criteria simultaneously.

The EPR control law gains that ensure proper stability and control are thus found during tests and design procedures. They are presented in the following table:

Table 4.2
Gain values for EPR Control law

K_P	K_I
0.33	0.23

At the output, the throttle position will be limited in rate to the throttle rate limit (10% of maximum value per second) and in position to the defined speed limits.

Figure 4.3 presents the EPR controller simulation model used in Simulink.

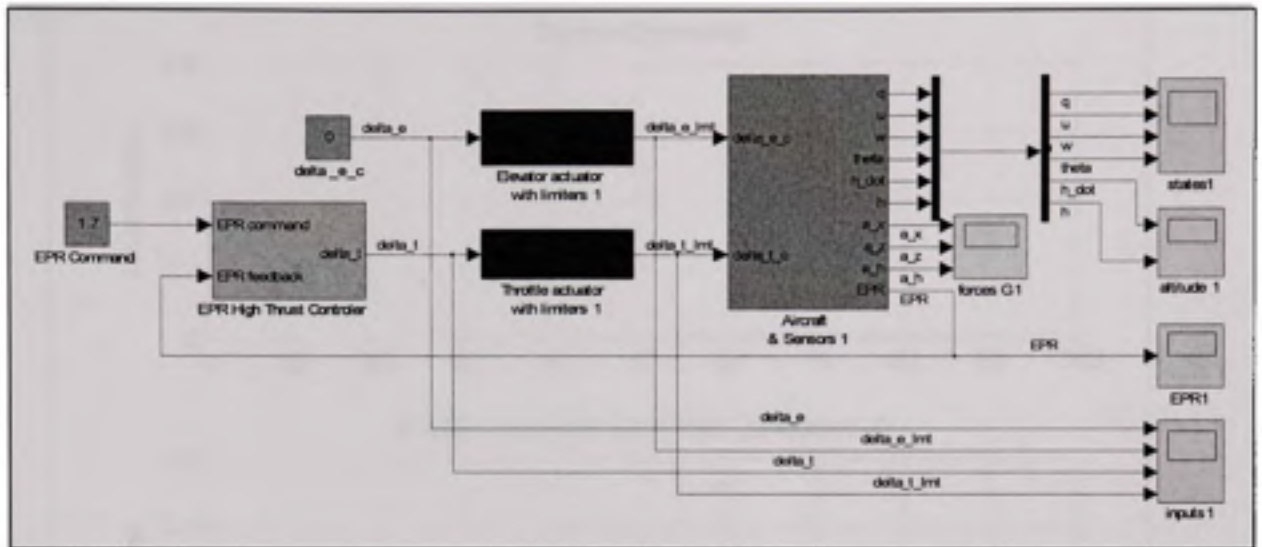


Figure 4.3 EPR controller Simulink simulation model.

Using the linear simulation model in Simulink, and the controller designed in the previous section, the controller's outputs are computed and presented in Figure 4.4.

Comparing a limited throttle command to a general throttle command, we notice that the thrust limitation does not affect the EPR controller to capture the desired EPR (step response input). The EPR settles after appreciatively 20 seconds, and no steady state error is observed. The controller respects well the system requirements as presented in chapter 2.

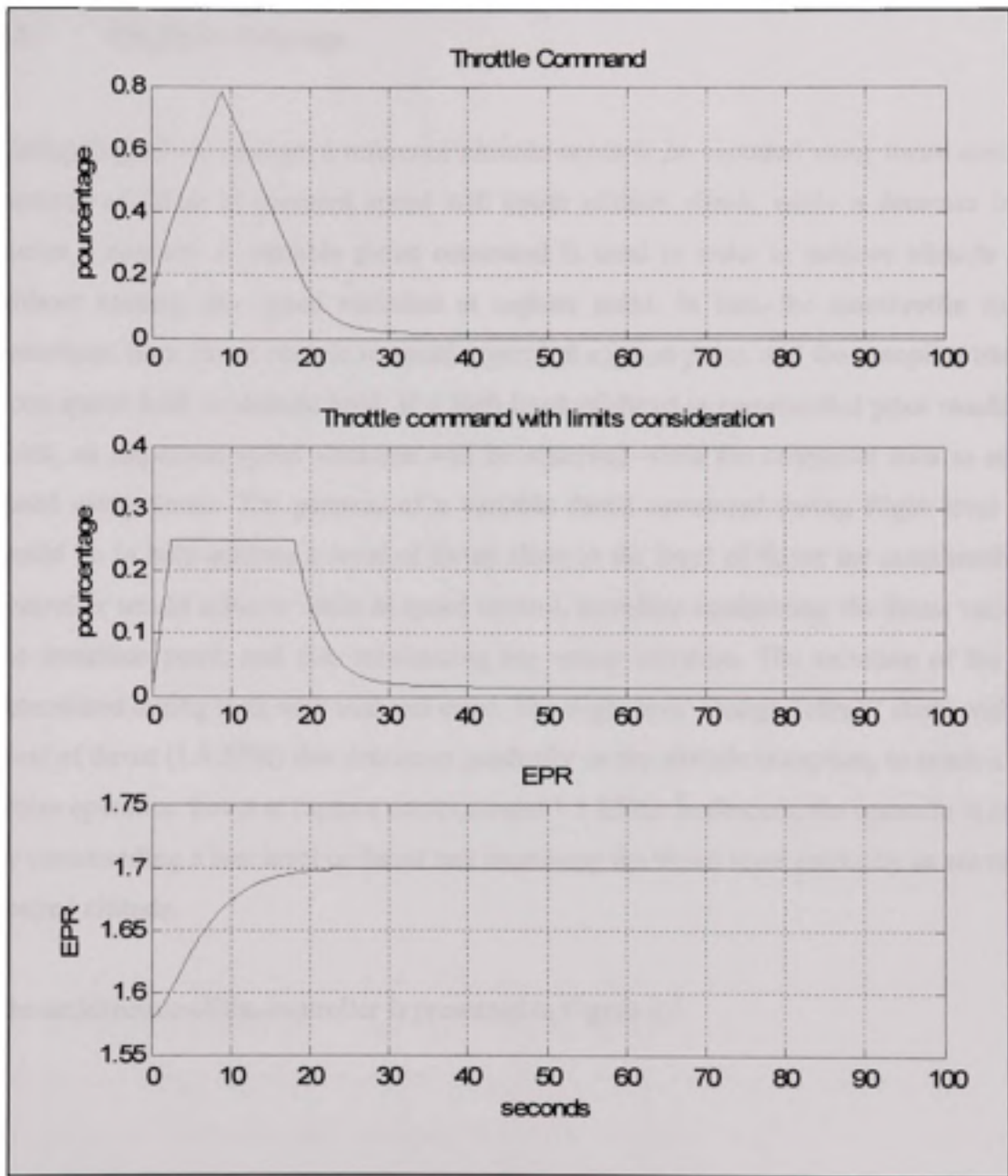


Figure 4.4 EPR controller outputs and EPR capture response.

4.3 Flight level change

During flight level change, a reference altitude needs to be captured using thrust control. An increase of thrust at constant speed will cause altitude climb, while a decrease in thrust causes a descent. A variable thrust command is used in order to achieve altitude capture without causing any speed variation at capture point. In fact, the autothrottle controller transitions from thrust control to speed control at capture point, and the autopilot transitions from speed hold to altitude hold. If a high level of thrust is commanded prior reaching that point, an important speed variation will be observed while the controller tries to settle the speed using thrust. The purpose of a variable thrust command during flight level change would be to help achieve a level of thrust close to the level of thrust the autothrottle speed controller would achieve while in speed control, therefore minimizing the thrust variation at the transition point, and thus minimizing any speed variation. The variation of the gain is determined during tests with trial and error. The flight level change “climb” starts with a high level of thrust (1.4 EPR) that decreases gradually as the altitude is capture, to reach a normal cruise operation thrust at capture point (around 1.1 EPR). In descent, the opposite is achieved by commanding a low level of thrust and increasing the thrust level gradually as we reach the desired altitude.

The architecture of the controller is presented in Figure 4.5:

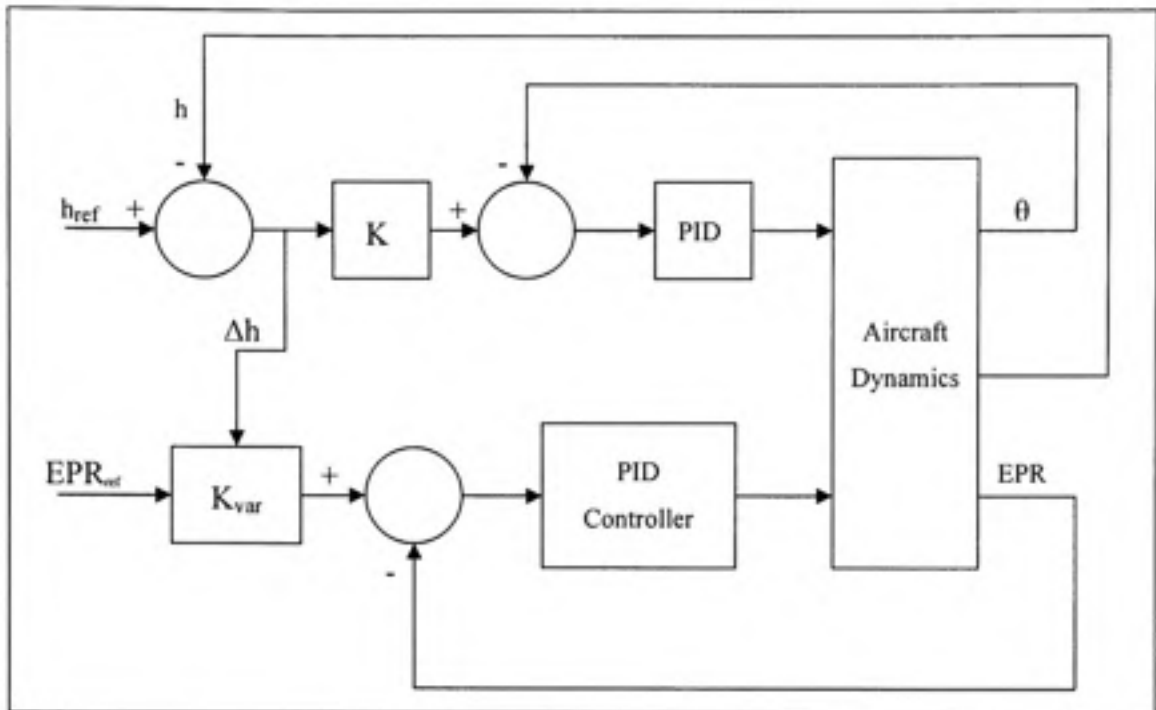


Figure 4.5 Controller architecture in Flight level change mode.

The EPR control channel has the same architecture as the normal EPR capture channel presented in the previous section. The altitude hold channel is the normal altitude hold channel used by the Autopilot integrated and used with the Flightsim aircraft model. The EPR command is adjusted using the difference between the altitude command and the actual altitude. During climb, the bigger the difference is between the command and the actual altitude, the more important the EPR command is. During descent, the more important the altitude is, the lower the EPR command is. Performance and results of the present controller are presented later on in chapter 8.

4.4 Speed control

During cruise, speed is maintained using a PID controller as presented in Figure 4.5.

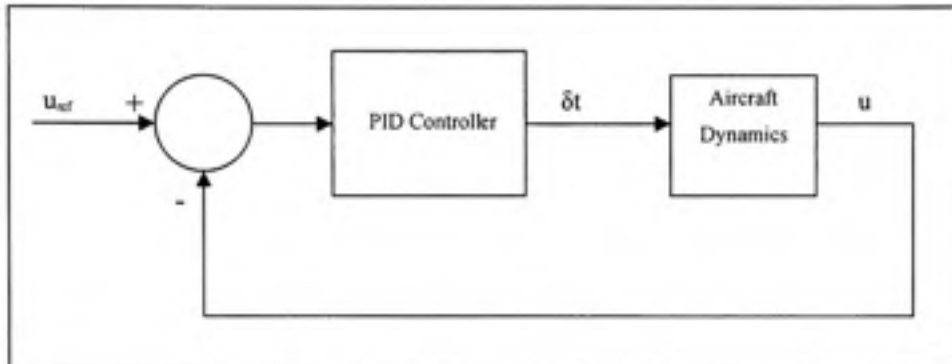


Figure 4.6 Speed Controller architecture.

In fact, the DTB already presented a classical speed controller loop. The same loop is maintained and integrated within the new autothrottle system. The parameters of the controller may be further tuned during simulation in real-time, using the same procedure presented in section 4.1.1, through the Flightsim menus.

At the output, the throttle position will be limited in rate to the throttle rate limit (10% of maximum value per second) and in position to the defined speed limits, similar to the EPR control channel.

During tuning, the controller was tested at several points of the flight envelope to guarantee performance over the full envelope. A single controller has been tuned and validated over the envelope. The controller's performance and capture results are further presented in chapter 8.

CHAPITRE 5

MODERN CONTROL DESIGN

In the previous chapter, the design of autopilot and autothrottle was a single input single output control system. In that case cross-coupling errors in speed and altitude will occur when manoeuvring. In order to perform speed variations at constant pitch attitude without acting on the stick, a multi-input pitch movement approach is used to coordinate the two control channels by means of pitch and thrust control coupling. Such algorithms provide full-time thrust and elevator control coordination for all modes and flight conditions, thus eliminating flight path deviations due to speed manoeuvres. The problem of multi-input multi-output control in this case is solved by using the eigenstructure assignment technique using state feedback. In this chapter we will present the application of eigenstructure assignment to our thrust control application.

5.1 Pole placement and Eigenstructure method

The state-feedback pole assignment in control system design is an eigenvalue determination problem. In the case of a multi-input multi-output (MIMO) system, the solution is, in general, undetermined, with many degrees of freedom. A desirable property of any system design is that the poles should be insensitive to perturbations in the coefficient matrices of the system equations. This criterion may be used to restrict the degrees of freedom in the assignment problem, and to produce a well-conditioned or robust solution.

A number of constructive methods for pole assignment by state feedback are described in the literature, Mayne and Murdoch (1970), Maki and Van de Vegte (1974), Barnett (1975), Gourishankar and Ramar (1976), Moore (1976), Klein and Moore (1977, 1982), Porter and D'Azzo (1978), Munro (1979), Wonham (1979), Flamm (1980), Varga (1981), Fahmy and O'Reilly (1982), Minmis and Paige (1982).

In the single-input case, where at most one solution to the pole assignment problem exists, these methods accurately compute the required feedback. In the multi-input case, however, where the feedback is undetermined, these methods do not generally lead to a robust solution to the problem.

5.1.1 Pole assignment

We consider the time-invariant, linear, multivariable system with dynamic state equation

$$\dot{\vec{x}} = A\vec{x} + B\vec{u} \quad (5.1)$$

where \vec{x} is the state vector and \vec{u} the control vector. A is the state coefficient matrix and B the driving matrix. \vec{x} and \vec{u} are n - and m -dimensional vectors, respectively. A , B are real, constant matrices of compatible orders. Matrix B is assumed (without loss of generality) to be of full rank. The behaviour of system is governed by the poles of the system, that is, by the eigenvalues of matrix A . It is often desirable to modify the poles of the system in order to obtain certain properties, such as stability. This may be achieved by using a state-feedback control:

$$\vec{u} = -K\vec{x} \quad (5.2)$$

where K , the feedback or gain matrix, is chosen such that the modified dynamic system

$$\dot{\vec{x}} = (A - BK)\vec{x} \quad (5.3)$$

has the desired poles.

The state-feedback pole assignment problem for system (5.1) is formulated precisely as follows:

Given real matrices (A, B) , of orders $n \times n$ and $n \times m$ respectively, and a set of n complex numbers $\{\lambda_1, \lambda_2, \dots, \lambda_n\}$ closed under complex conjugation, find a real $m \times n$ matrix K such that the eigenvalues of $A - BK$ are $\lambda_j, j = 1, 2, \dots, n$.

According to the Theorem of Wonham (Wonham 1979), a solution K to the pole assignment problem exists for every set of self-conjugate complex numbers, $\{\lambda_1, \lambda_2, \dots, \lambda_n\}$, if and only if the pair (A, B) is completely controllable.

In the case of a single input single output system, the solution to the pole assignment problem can be found using Ackerman's formula [11] which is based on the Cayley-Hamilton Theorem which states that the matrix $(A - BK)$ satisfies its own characteristic equation.

However in the multi-input multi-output case, the solution is, undetermined, with many degrees of freedom. Other criteria may be considered to restrict the solution of matrix K . A desirable property of any system design is that the poles should be insensitive to perturbations in the coefficient matrices of the system equations. This criterion may be used to restrict the degrees of freedom in the assignment problem, and to produce a well-conditioned or robust solution. This criterion is achieved through a proper eigenvector assignment.

5.1.2 Robust eigenstructure assignment

Typically, for a MIMO controllable system, a given set of eigenvalues can be assigned by an infinity of gain matrices. This fact can be exploited to exercise the freedom "beyond"

placement of the eigenvalues to achieve robustness or to extremize some given performance measure.

Considering the time-invariant, linear, multivariable system with dynamic state equation presented in equation (5.1), and considering the set of n complex numbers $\{\lambda_1, \lambda_2, \dots, \lambda_n\}$ the robust pole assignment problem for the system (5.1) can be formulated as follows:

Given real matrices (A, B) , of orders $n \times n$ and $n \times m$ respectively, find real matrix K and non singular matrix X satisfying $(A - BK)X = X\Lambda$ where $\Lambda = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_n\}$ and X is the matrix of eigenvectors.

Our objective is to choose eigenvectors, given by X , satisfying $(A - BK)X = X\Lambda$ and such that the conditioning of the eigenproblem is minimized.

A number of eigenstructure assignment formulations and algorithms have been developed and have been shown to be useful tools for state and output feedback system designs. These design approaches allow the designer to directly choose eigenvalues and to explore the arbitrariness of admissible eigenvectors.

Differences arise among the several algorithms because of choices of criteria that generate the closed loop eigenvectors, as well as the implementation details. The success of these approaches depends mainly on the selection of criteria or parameter sets that explicitly determine the closed-loop eigenvectors.

It is well known that near orthogonality of the closed-loop eigenvectors is very desirable to minimize sensitivity of eigenvalue placement to model errors. Arbitrary selection of feasible eigenvectors may cause the eigenvector modal matrix to be ill conditioned. If this is the case, the associated gain matrix may not be accurately calculated, and, further, the stability and response of the closed-loop system may be highly sensitive to parameter variations. Therefore,

generation of well-conditioned eigenvectors is a key issue for this family of pole placement algorithms with regard to achieving both insensitive feedback design and numerical stability.

The Matlab function ‘‘place’’ uses the algorithm developed by Kautsky J., Nichols & Dooren (1985) which, for multi-input systems, optimizes the choice of eigenvectors for a robust solution. This algorithm uses the extra degrees of freedom to find a solution that minimizes the sensitivity of the closed-loop poles to perturbations in A or B. In this algorithm, the eigenvectors are chosen by an iterative process to minimize the measure of conditioning.

5.2 Controller design

In order to perform speed variations at constant pitch attitude without acting on the stick, a multi-input pitch movement approach is used to coordinate the two control channels by means of pitch and thrust control coupling. Such algorithms provide full-time thrust and elevator control coordination for all modes and flight conditions, thus eliminating flight path deviations due to speed manoeuvres. Two channels (pitch and speed) will be coupled using the state feedback control method. The pitch command will be used to maintain altitude at actual altitude (cruise mode), while the speed channel will be used to capture the reference speed.

The implementation of the corresponding control law is done as the feedback gains are calculated by using the eigenvalues and eigenvectors method (pole placement and eigenstructure selection). The algorithm calculates the feedback gains which are given by the matrix **K**:

$$K = \begin{bmatrix} K_{1,1} & K_{1,2} & K_{1,3} & K_{1,4} \\ K_{2,1} & K_{2,2} & K_{2,3} & K_{2,4} \end{bmatrix} \quad (5.4)$$

The commands are therefore given by:

$$de_c = K_{pitch}\theta_c - (K_{1q}q + K_{1u}u + K_{1w}w + K_{1\theta}\theta) \quad (5.5)$$

$$dt_c = K_V V_c - (K_{2q}q + K_{2u}u + K_{2w}w + K_{2\theta}\theta) \quad (5.6)$$

The feedback loop on each channel is the sum of each measured aircraft parameter (pitch rate, pitch angle, true airspeed and vertical speed) multiplied by a gain, as shown in the state feedback block in Figure 5.1.

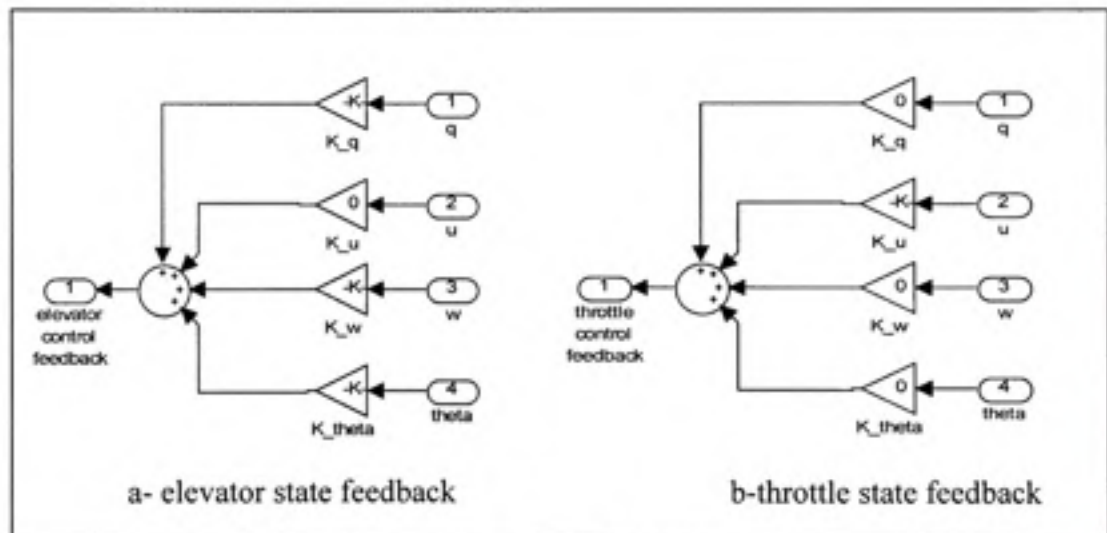


Figure 5.1 State feedback blocks.

Without loss of generality, we shall apply the controller theory on the linearized model presented in section 3.7. We will then extend our study to other equilibrium points of the flight envelope and extend the controller design using gain scheduling techniques.

The linear model is represented by matrices A and B:

$$A = \begin{bmatrix} -0.306613 & -0.000529565 & -0.00330512 & -0.000621396 \\ -0.531057 & -0.00674625 & 0.0195349 & -9.83466 \\ 237.123 & -0.12062 & -0.664316 & -0.0195343 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 0.120458 & 0.0131793 \\ -0.0178595 & 2.09909 \\ 1.11696 & -0.0916484 \\ 0 & 0 \end{bmatrix}$$

5.2.1 Controllability

The best way to determine the controllability of our system is by determining the rank of the controllability matrix. The rank has to be equal to the number of states in order that the system is fully controllable. Using Matlab and matrices A and B, the determination of the rank of the controllability matrix is straightforward. We find the following results:

- per input δ_e : rank=4 hence controllable
- per input δ_i : rank=4 hence controllable

The condition to the application of the pole assignment technique is therefore met.

5.2.2 State feedback

We choose to place poles at the following positions:

$$P = [-0.6+j*0.6 \ -0.6-j*0.6 \ -0.2 \ -0.2] \quad (5.7)$$

This choice will guarantee that the rapid mode is fast and well damped and that the phugoid mode doesn't present any oscillation or overshooting.

The matrix of gain K found is the following:

$$K = \begin{bmatrix} 4.9105 & -0.0134 & -0.0237 & 2.1875 \\ -11.2786 & 0.1000 & 0.0452 & -2.8095 \end{bmatrix} \quad (5.8)$$

We shall now observe the time response of our system using matrix K and the state feedback.

5.2.2.1 System time response to initial conditions (theta =0.5 rad)

We observe a time response of 30 seconds on the speed response, as shown in Figure 5.2.

By observing the input command in Figure 5.3, we notice that the input is very important, since it goes beyond the value of 1. The input is normalized between 0 and 1. However this is normal since the initial condition of 0.5 radians is far too important.

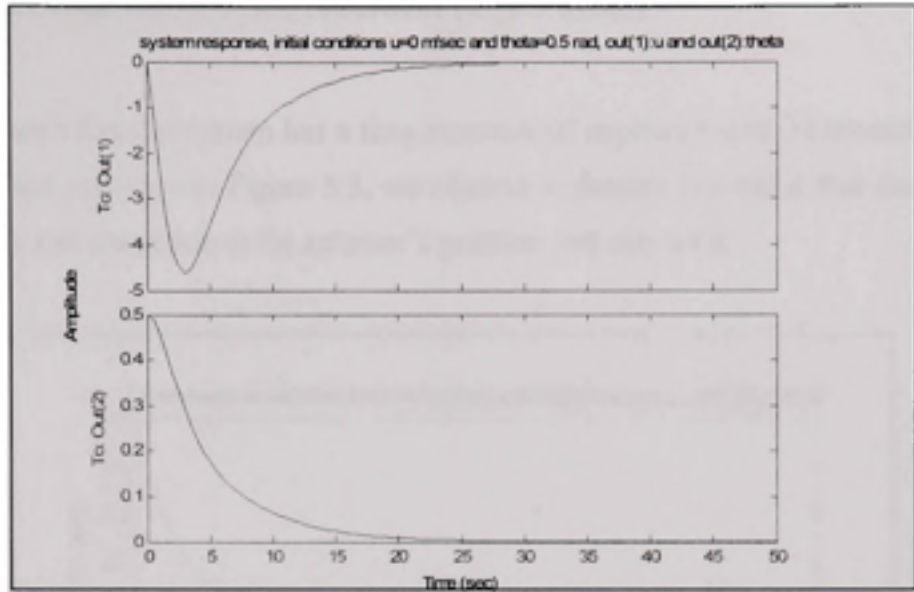


Figure 5.2 System response for initial conditions $u=0$ m/sec and $\theta=0.5$ rad.

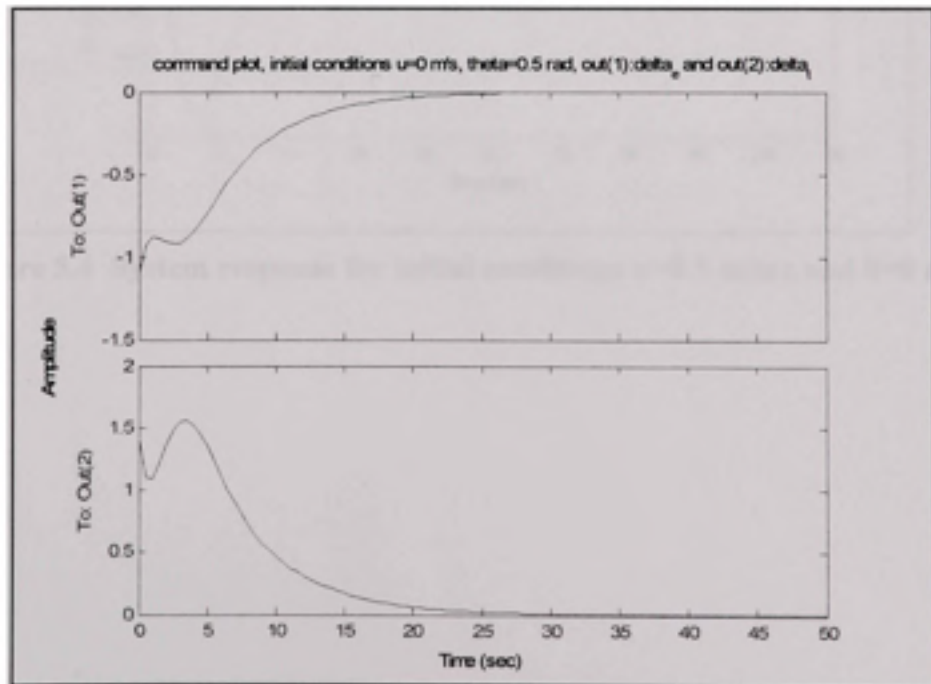


Figure 5.3 Command plot for initial conditions $u=0$ m/sec and $\theta=0.5$ rad.

5.2.2.2 Time response to initial conditions ($u = 0.5$ m/sec)

Figure 5.4 shows that the system has a time response of approximately 30 seconds. Looking at the command response in Figure 5.5, we observe a smooth command that doesn't cause any instability and excitation at the actuator's position and rate limits.

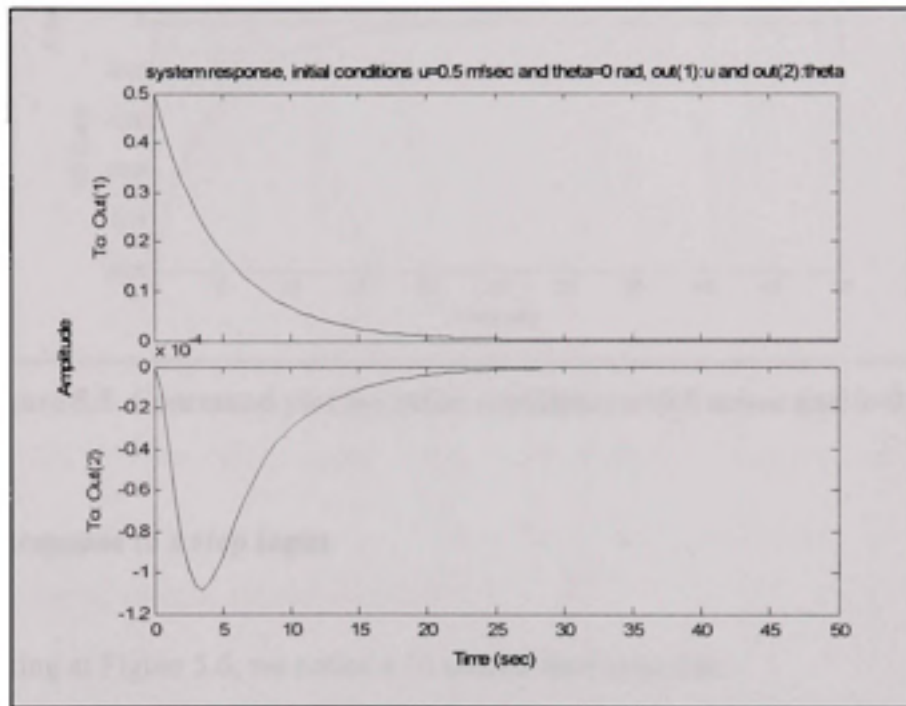


Figure 5.4 System response for initial conditions $u=0.5$ m/sec and $\theta=0$ rad.

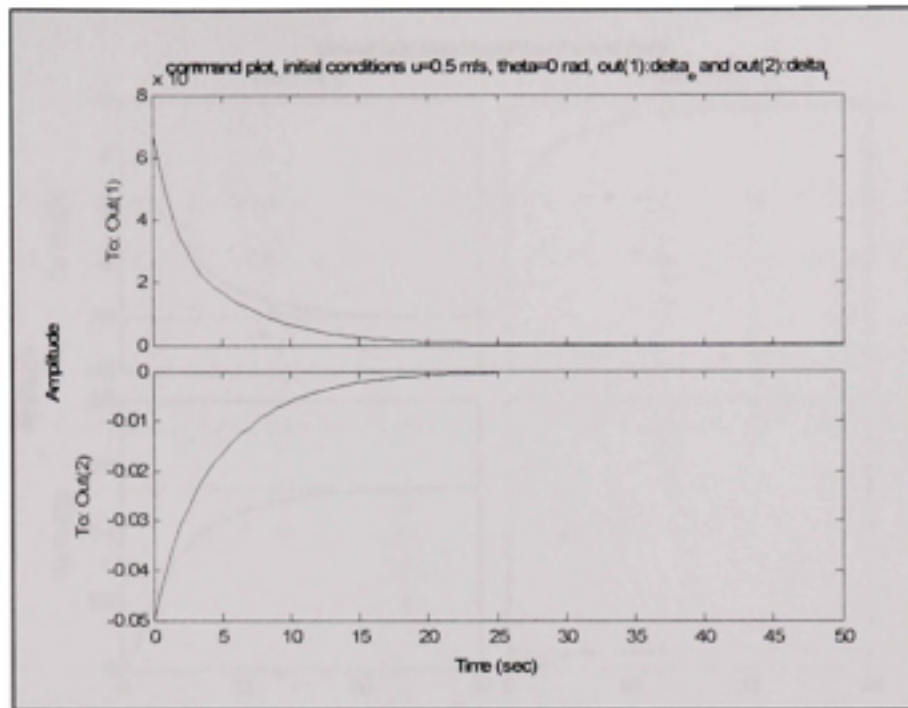


Figure 5.5 Command plot for initial conditions $u=0.5$ m/sec and $\theta=0$ rad.

5.2.2.3 Response to a step input

Again, looking at Figure 5.6, we notice a 30 second time response:

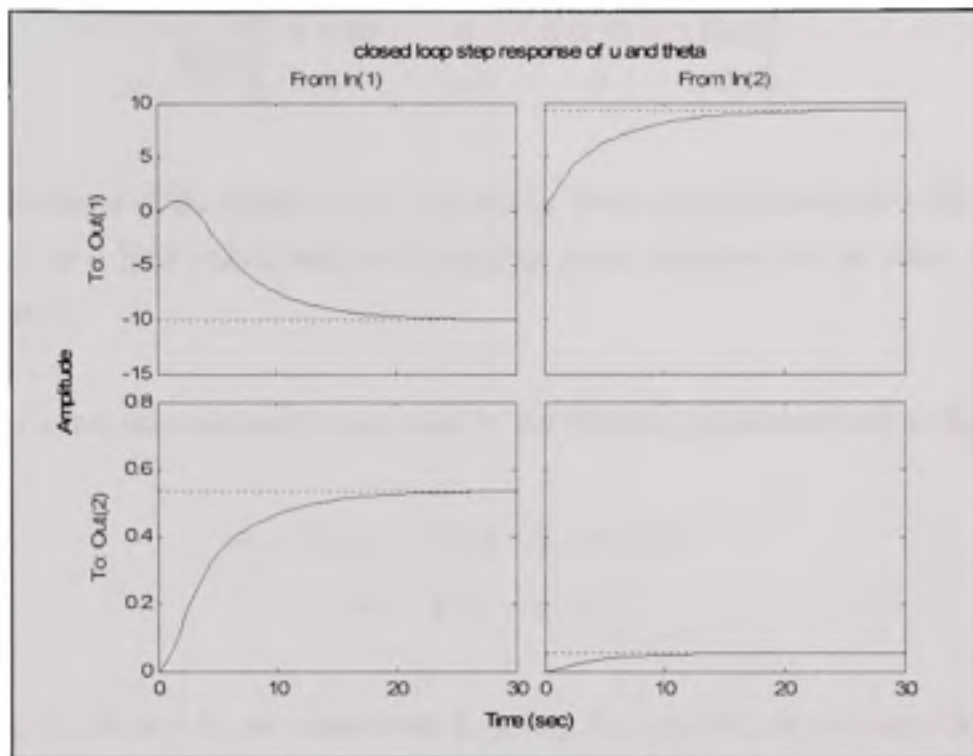


Figure 5.6 Closed loop step response of speed and pitch, with state feedback matrix K .

5.2.3 Effect of various gains on feedback

To further simplify the control law, certain state feedbacks may be neglected because they have little effect on each channel. During various tests, it is observed that the exclusion of certain feedbacks do not affect the stability of the control law. On the pitch channel, the speed feedback is neglected while the pitch angle, vertical speed and pitch rate feedbacks are kept. On the speed channel, only the speed feedback is used while all the other feedbacks are neglected. Observing matrix K in the previous section, we can see that the effect of the "u" feedback on the elevators input is minimal compared to other feedbacks. We cancel these feedbacks and found the modified gain matrix K_m :

$$K_m = \begin{bmatrix} 4.9105 & 0 & -0.0237 & 2.1875 \\ 0 & 0.1 & 0 & 0 \end{bmatrix} \quad (5.9)$$

The performance of the system is not affected by these simplifications since the neglected feedbacks have little effect and small damping gains compared to the states taken into consideration.

Thus the control laws can now be expressed by the following equations (refer to figure 4.3):

$$de_c = K_{pitch}\theta_c - (K_q q + K_w w + K_\theta \theta) \quad (5.10)$$

$$dt_c = K_v V_c - K_u u \quad (5.11)$$

where K_q , K_w , K_θ and K_u are respectively $K_{1,1}$, $K_{1,3}$, $K_{1,4}$, and $K_{2,2}$ in the matrix \mathbf{K} presented in equation 5.4. K_{pitch} and K_v are respectively the pitch scale elevator gain and the speed scale throttle gain as presented in Figure 5.9.

Let's start by observing the pole position in the closed loop to study the system stability, the poles are:

$$-0.6109 + 0.4562i$$

$$-0.6109 - 0.4562i$$

$$-0.1034$$

$$-0.4273$$

The system is stable. The phugoid mode is a little slower using K_m compared to using gain matrix \mathbf{K} , but does not present any imaginary term. The fast mode is well damped.

Now let us observe the effect of the modification to matrix K by observing the system time response to the same step input in closed loop, and using matrix K_m :

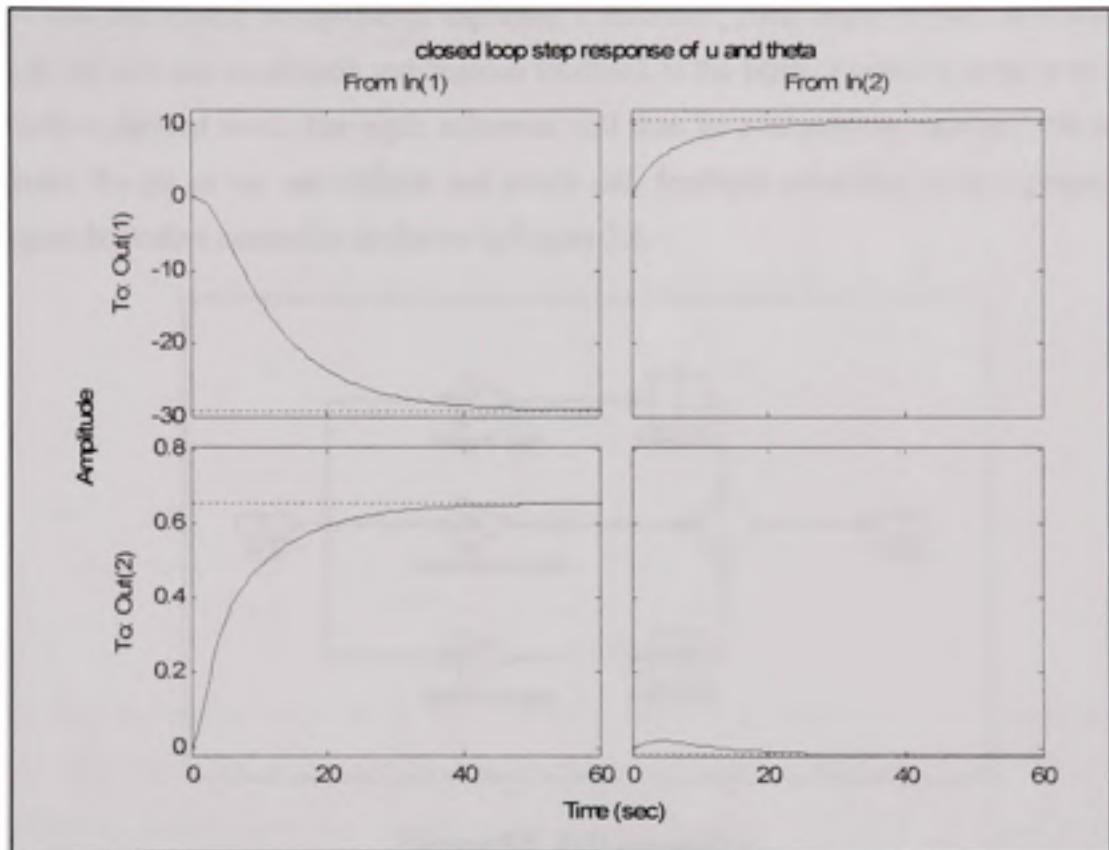


Figure 5.7 Closed loop step response with state feedback gain matrix K_m .

We observe the fact that the time response nearly doubled and is almost of 60 seconds, compared to 30 seconds, as shown in Figure 5.7. The modified controller, using the simplified gain matrix, will have to be validated upon integration to respect the system requirements presented in chapter 2.

5.2.4 Altitude hold

Now that the system is capable of capturing a reference pitch angle as well as a reference speed, we will add an altitude stabilisation feedback to the pitch channel in order to maintain altitude at desired level. The pitch reference will then be evaluated to maintain this desired altitude. To do so we use altitude and climb rate feedback combined with a proportional integral derivative controller as shown in Figure 5.8.

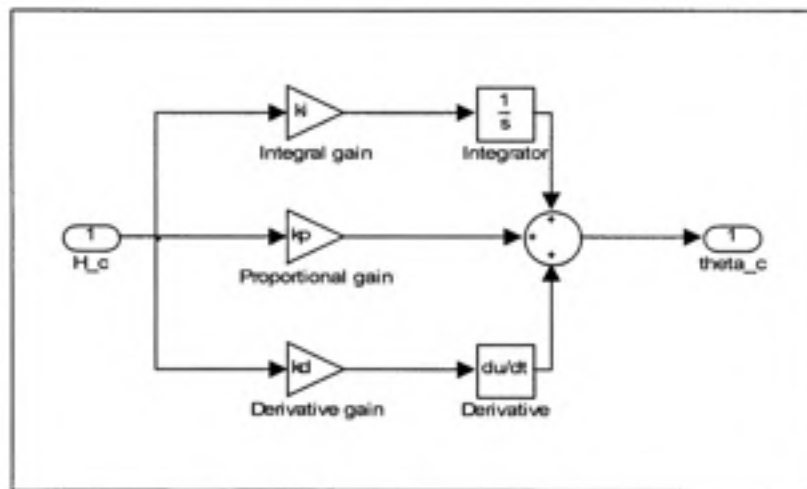


Figure 5.8 PID controller.

We can now express the control law by the following equation:

$$\theta_c = K_p (K_H H_{error} - \dot{H}) + K_I \int (K_H H_{error} - \dot{H}) dt + K_D \frac{d(K_H H_{error} - \dot{H})}{dt} \quad (5.12)$$

Using equations 5.9 and 5.11, we can express the global elevator channel control law by:

$$de_c = K_{pitch} \left[K_p (K_H H_{error} - \dot{H}) + K_I \int (K_H H_{error} - \dot{H}) dt + K_D \frac{d(K_H H_{error} - \dot{H})}{dt} \right] - (K_q q + K_w w + K_\theta \theta) \quad (5.13)$$

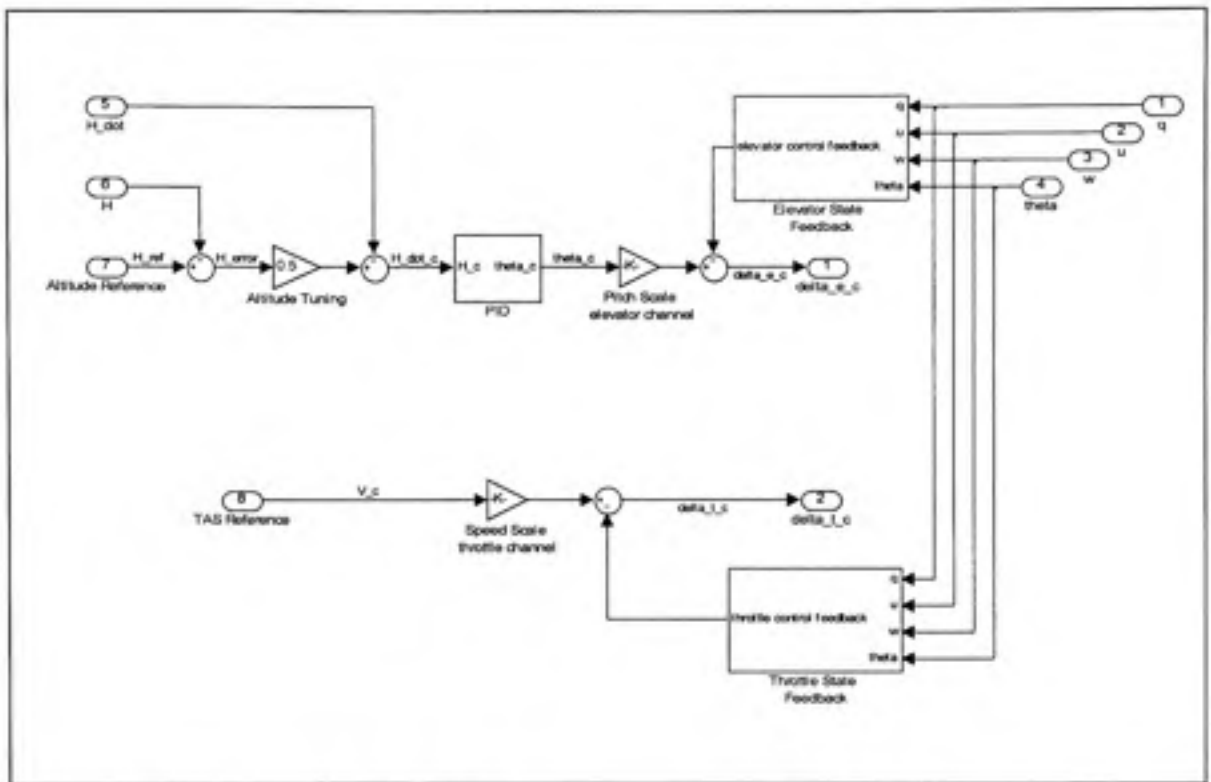


Figure 5.9 Speed Capture control law with altitude hold.

The various tunable gains (other than the state feedback gains and the Static scaling gains) were calculated for the selected trim point using numerical optimisation methods and algorithms during tests and design process. Gains were calculated to ensure a system response within predefined limits. A constraint envelope is defined that fixes upper and lower bounds of the controlled variable, together with a number of tunable variables (the gains of the controller). A cost function is generated consisting of a weighted maximum constraint violation. At every iteration, each tuneable variable is perturbed in turn and the resulting constraint values and cost function are evaluated. A gradient search direction is determined from these results and a line search along the gradient is performed in order to minimise the cost function while simultaneously satisfying the constraint envelope criteria.

5.2.5 Simulation results

The simulation model of the state-feedback controller, with limitations and EPR protection is presented in Figure 5.10. EPR protection will be presented in detail in the next chapter.

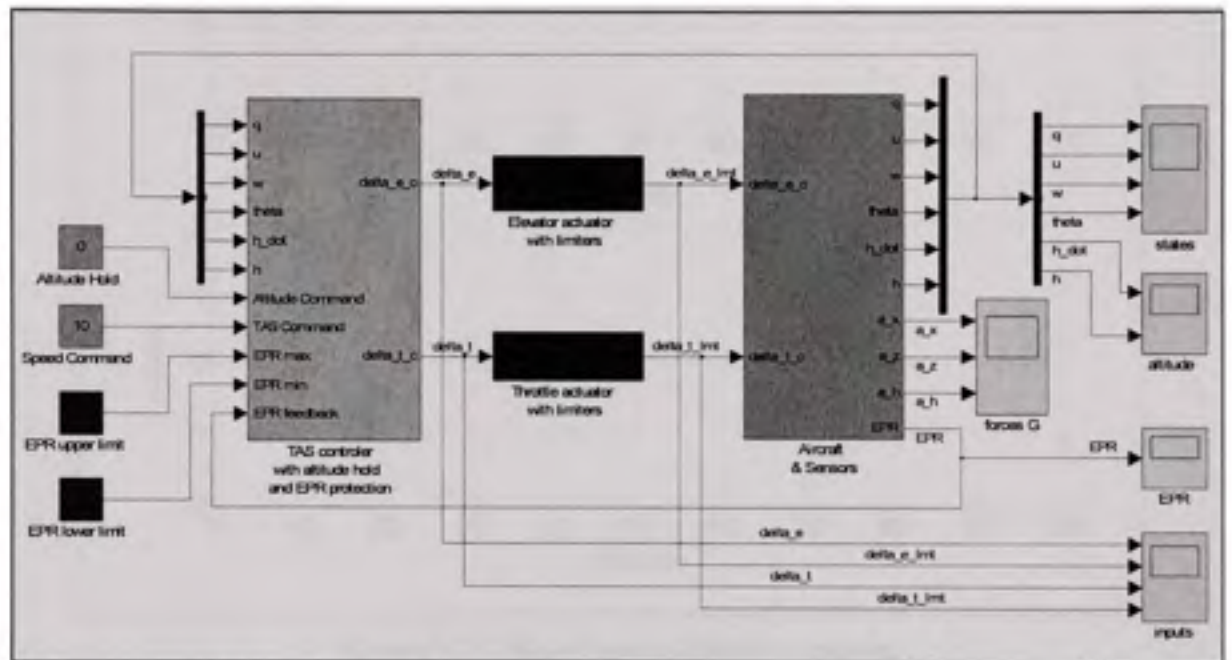


Figure 5.10 Pitch-Speed state-feedback controller architecture.

5.2.5.1 Speed u and altitude h

For an input command of 10 m/sec (a step of 10 m/sec with respect to the trim speed of 237 m/sec), the time response is 30 sec which is well within acceptable limits, as seen in Figure 5.11.

The altitude varies between -2 and +2 metres, a variation of around 4 meters peak to peak, before stabilizing. We will later on observe the G force variation due to this variation of altitude to make sure this variation is acceptable.

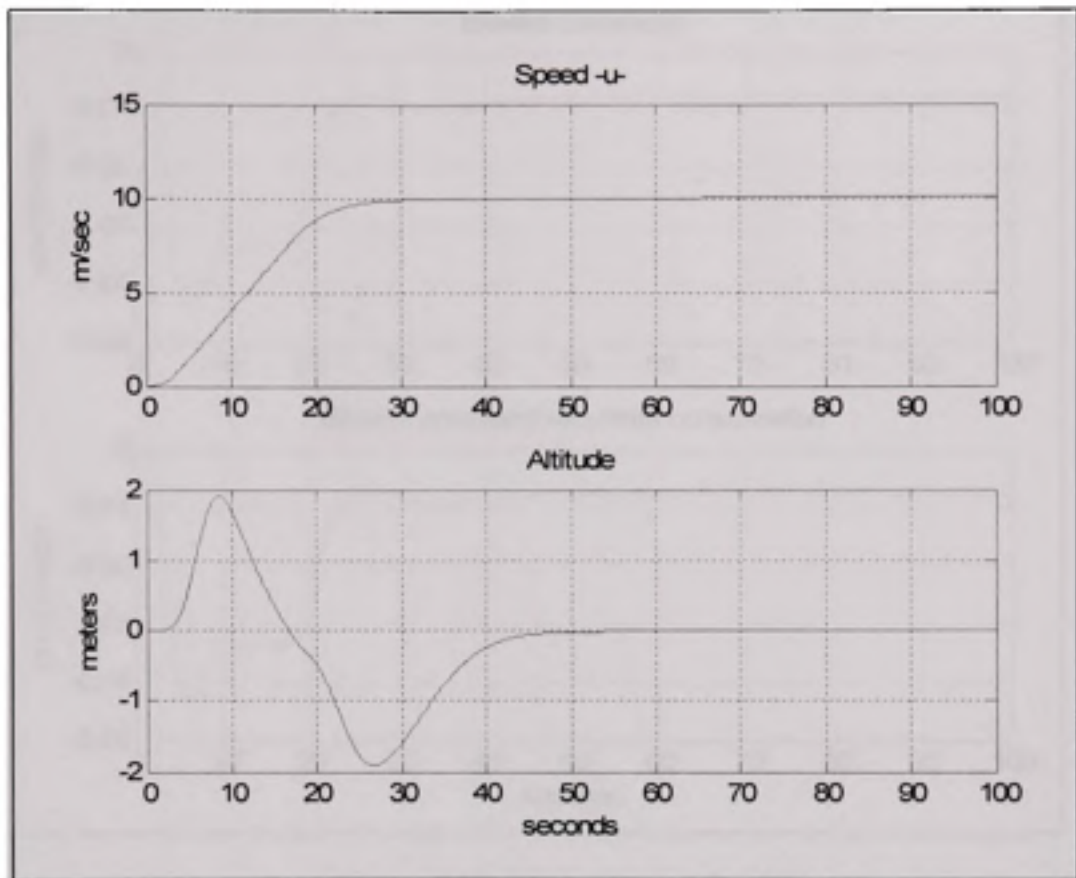


Figure 5.11 Speed and altitude response.

5.2.5.2 Input command on the elevators with and without limiting

The input is normalised between 0 and 1. We have limited the rate variation to 10% per second, and limited the actuator position to +0,5 and -0,5, knowing that the linearization was done at a trim point of 0. We notice that there is no saturation and hence the limiters do not affect the command input, as seen in Figure 5.12.

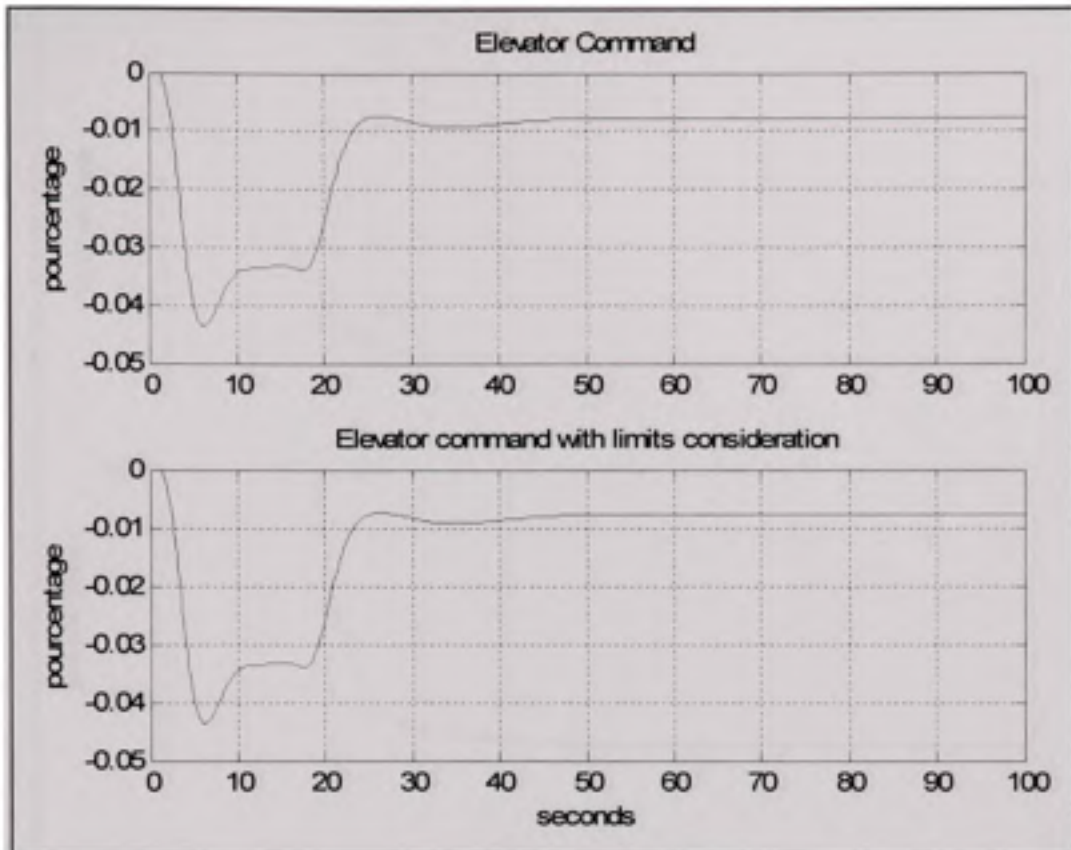


Figure 5.12 Elevator command.

5.2.5.3 Input command on the throttles with and without limiting

The input is normalized between 0 and 1. However, at our trim position we have a trim thrust of 75%. Hence our maximum variation from trim position is 25%. The speed variation is limited to 0,1/second.

In Figure 5.13, we notice the saturation during almost 10 seconds. The system still reaches stability after almost 40 seconds.

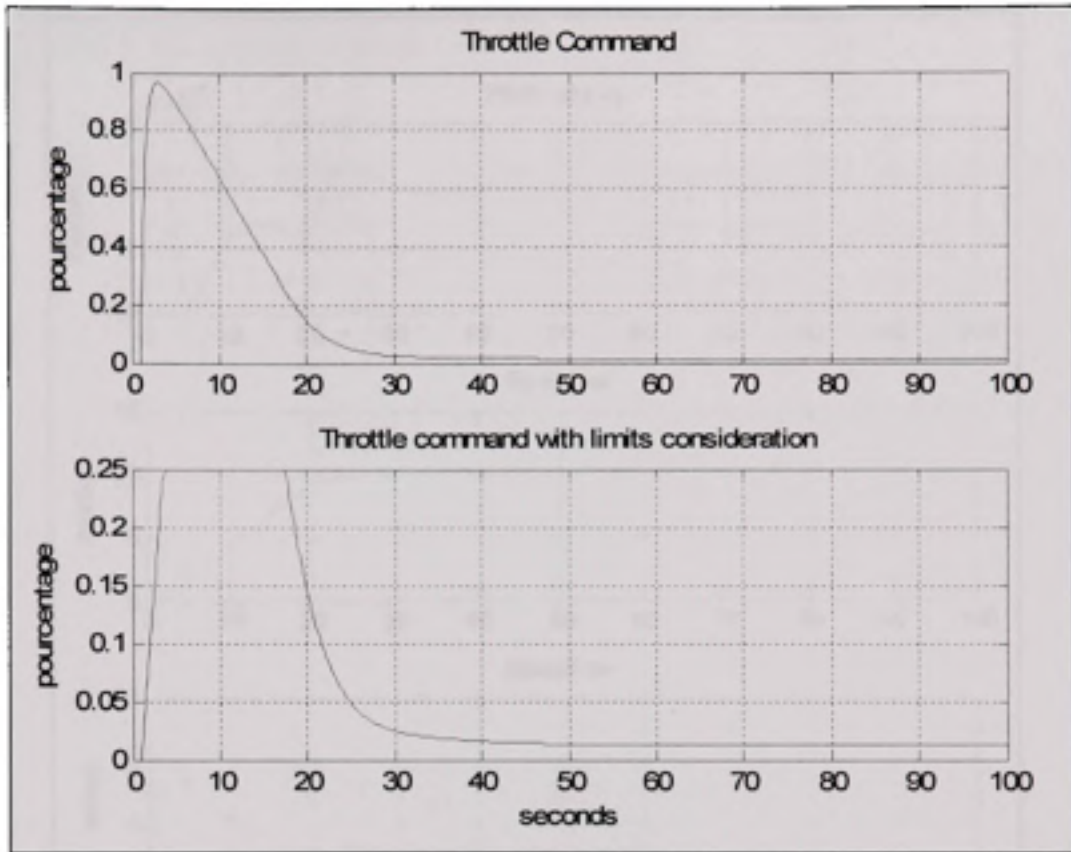


Figure 5.13 Throttle command.

5.2.5.4 System states

The system states variation, used as the controller feedback, is presented in Figure 5.14. Pitch rate, pitch, aircraft X axis speed, aircraft Z axis speed are all read and returned to the controller's input as feedback.

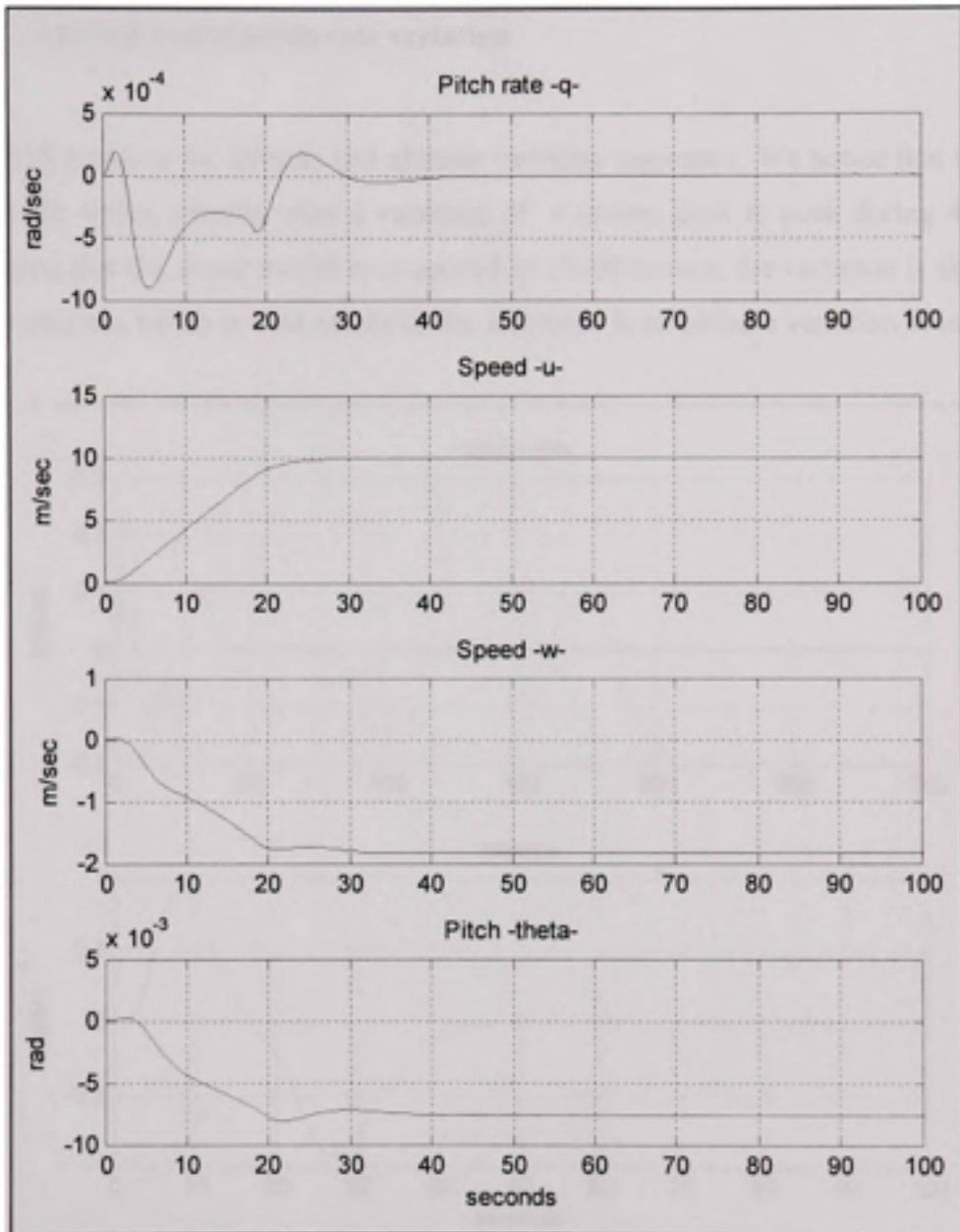


Figure 5.14 Aircraft model states response.

5.2.5.5 Altitude and altitude rate variation

Figure 5.15 presents the altitude and altitude variation responses. We notice that the altitude settles at the initial altitude, after a variation of 4 meters peak to peak during 40 seconds, Considering that the linear model is computed at 10000 meters, the variation is about 0.04% during transition, which is well within limits, and there is no altitude variation at settle time.

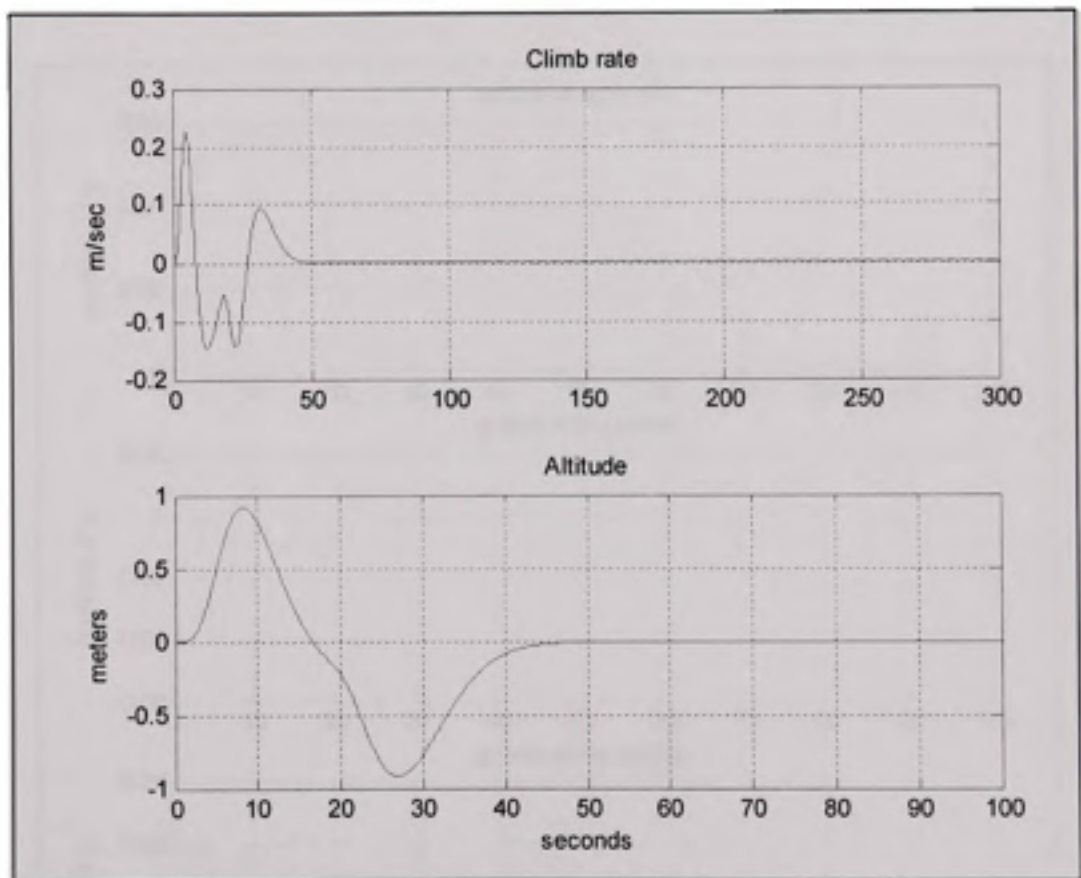


Figure 5.15 Altitude and climb rate response.

5.2.5.6 G forces on the different axis

Figure 5.16 presents the G loads on the different body axes. Our system design requirement indicates that the system shall keep the G force within a 0.1G limit. The most important acceleration occurs on the X axis due to the variation of u . We reach almost 0,05g which is well within limits.

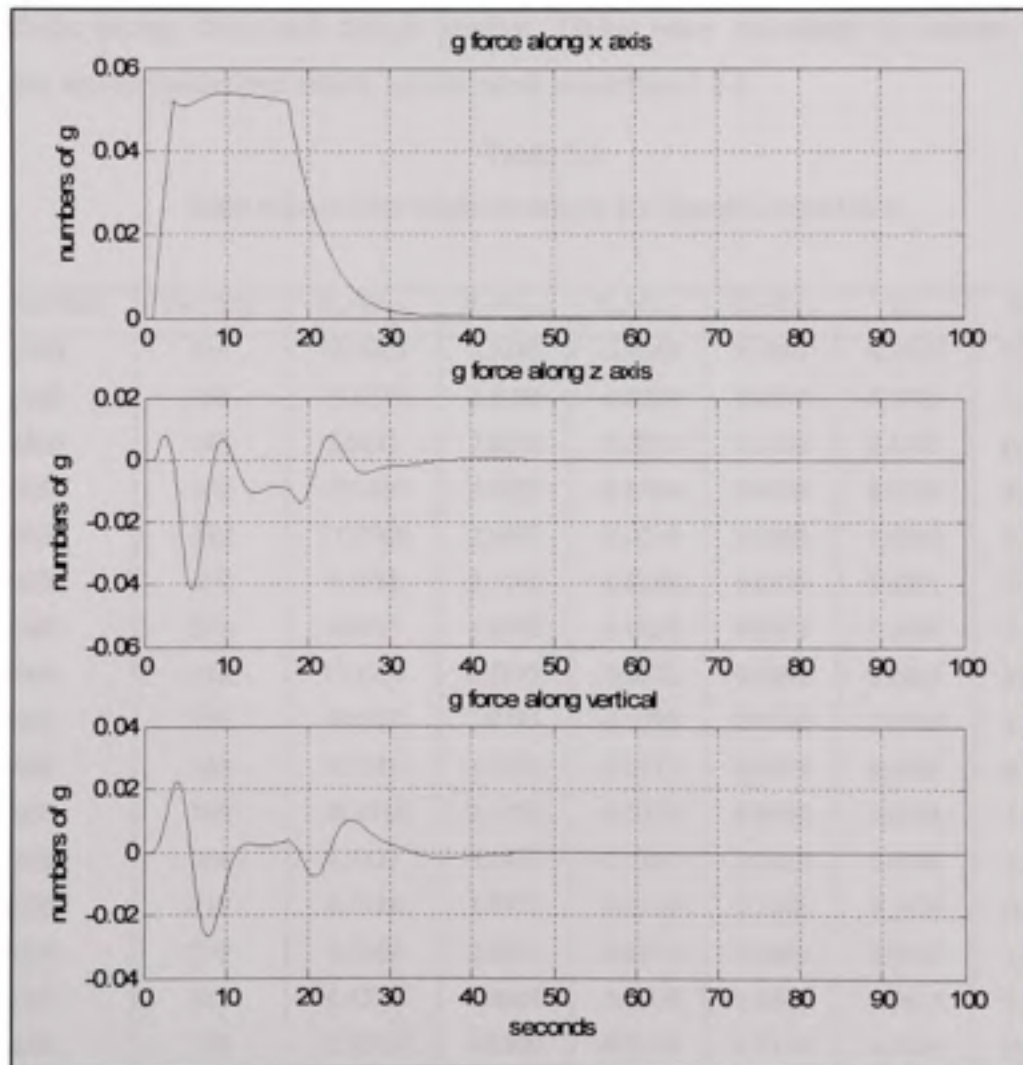


Figure 5.16 G-load on aircraft body axes.

5.3 Gain scheduling

The selection of the gains will consider different regions of the flight envelope and the values of these gains will be selected from a table.

The following table presents the calculated gains for different points of the flight envelope. The various tuneable gains (other than the state feedback gains and the Static scaling gains) were calculated for the selected trim points using numerical optimisation methods and algorithms during tests and design process. Gains were calculated to ensure a system response within predefined limits, as discussed in section 5.2.4

Table 5.1
Gain values over flight envelope for Speed Control law

Altitude trim	TAS trim	$K_q=K_{1,1}$	$K_\theta=K_{1,4}$	$K_w=K_{1,3}$	$K_a=K_{2,2}$	K_V	K_{pitch}
12000	415	13.0406	3.2485	-0.0283	0.1063	0.1078	2.6457
12000	449	10.8770	2.5364	-0.0230	0.0996	0.1005	1.9585
12000	483	8.8341	1.8213	-0.0269	0.1036	0.1076	0.9358
10000	335	15.1485	3.9725	-0.0356	0.0939	0.0966	3.2988
10000	382	11.3768	2.6871	-0.0314	0.0864	0.0884	2.1076
10000	412	9.4738	2.1140	-0.0266	0.0796	0.0811	1.6055
8000	271	4.5017	1.2505	-0.0106	0.0857	0.0895	1.0752
8000	314	13.1956	3.2558	-0.0372	0.0842	0.0884	2.6544
8000	371	9.0157	2.0195	-0.0298	0.0740	0.0759	1.5295
6000	243	4.3185	1.1531	-0.0117	0.0818	0.0868	0.9933
6000	308	10.2510	2.3735	-0.0374	0.0730	0.0768	1.8798
6000	360	6.9627	1.5454	-0.0267	0.0629	0.0654	1.1174
4000	219	4.1438	1.0771	-0.0126	0.1706	0.1820	0.9408
4000	279	9.5249	2.2297	-0.0371	0.0660	0.0703	1.7497
4000	326	6.4326	1.4681	-0.0264	0.0603	0.0635	1.0855
2000	198	3.9075	0.9989	-0.0128	0.1310	0.1418	0.8874
2000	253	8.7388	2.0936	-0.0341	0.1282	0.1359	1.6811
2000	296	-1.8460	1.3769	-0.0247	0.0527	0.0562	1.0082

The altitude hold channel gains and parameters are found to not vary with the variation of altitude and speed. In fact, the controller with gains presented in table 5.2 was tested at all the trim points considered in table 5.1, and the controller's performance was validated.

Table 5.2
Non varying gain values for Speed Control law

K_P	K_I	K_D	Altitude Tuning
0.0146	0.0022	0.0285	0.5

Initially, the parameters were computed for an initial trim point using an optimization algorithm to respect system requirements. Using the same algorithm over several trim points, we noticed little if no variation to the parameters. We then tested the initially computed parameters over the envelope and validated the controller's performance.

CHAPITRE 6

CRITICAL LIMITS AND PROTECTIONS

6.1 EPR and engines limit protection

The speed control law is coupled with EPR control laws to respect engine limitations. To be able to achieve EPR limit protection, the EPR control laws similar to the control laws presented in chapter 4 are used with EPR limits as the reference EPR to generate a throttle angle position command. The throttle angle commanded by the EPR control laws is compared to the throttle angle commanded by the speed control law as shown in Figure 6.1 on the next page.

Two EPR control laws are used, one with maximum EPR (EPR_{max}) as the reference and the second with EPR_{min} as the reference. The first control law will generate a throttle maximum position. This position is compared to the throttle position command given by the speed control law and the minimal value is selected to guarantee operation within engine limits. Similarly, the second EPR control law commands a minimum throttle position which is compared to the throttle position command given by the speed command law and the maximal value is selected. This logic will protect the engine from overboost and stall.

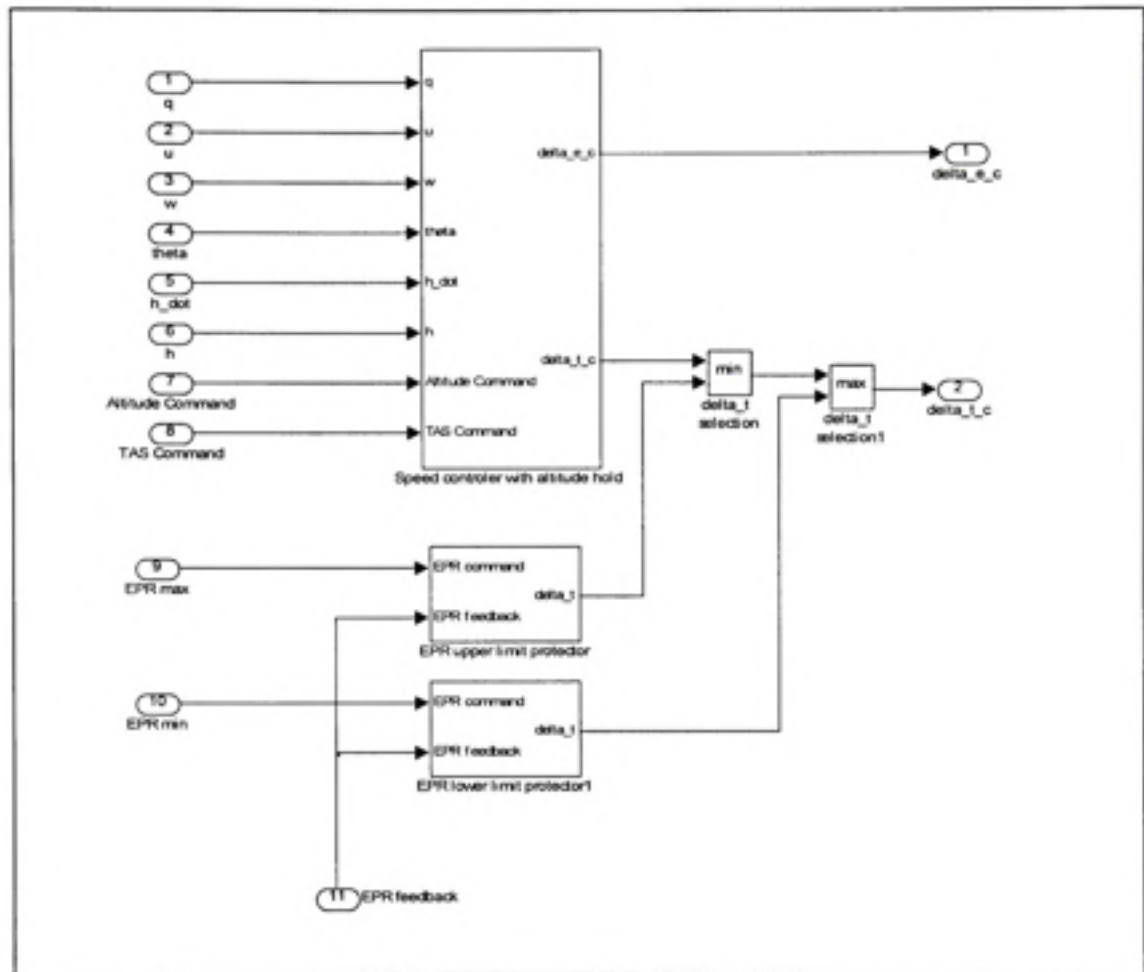


Figure 6.1 Controller with EPR limit protection.

The EPR limit protection controller behavior is presented in Figures 6.2 and 6.3. Using EPR limits of $EPR_{min}=1$ and $EPR_{max}=2$, we notice that the speed command is reached within 20 seconds, with no EPR saturation at any of the limits. The EPR reaches 1.74 and then settles around 1.59. This is shown in Figure 6.2.

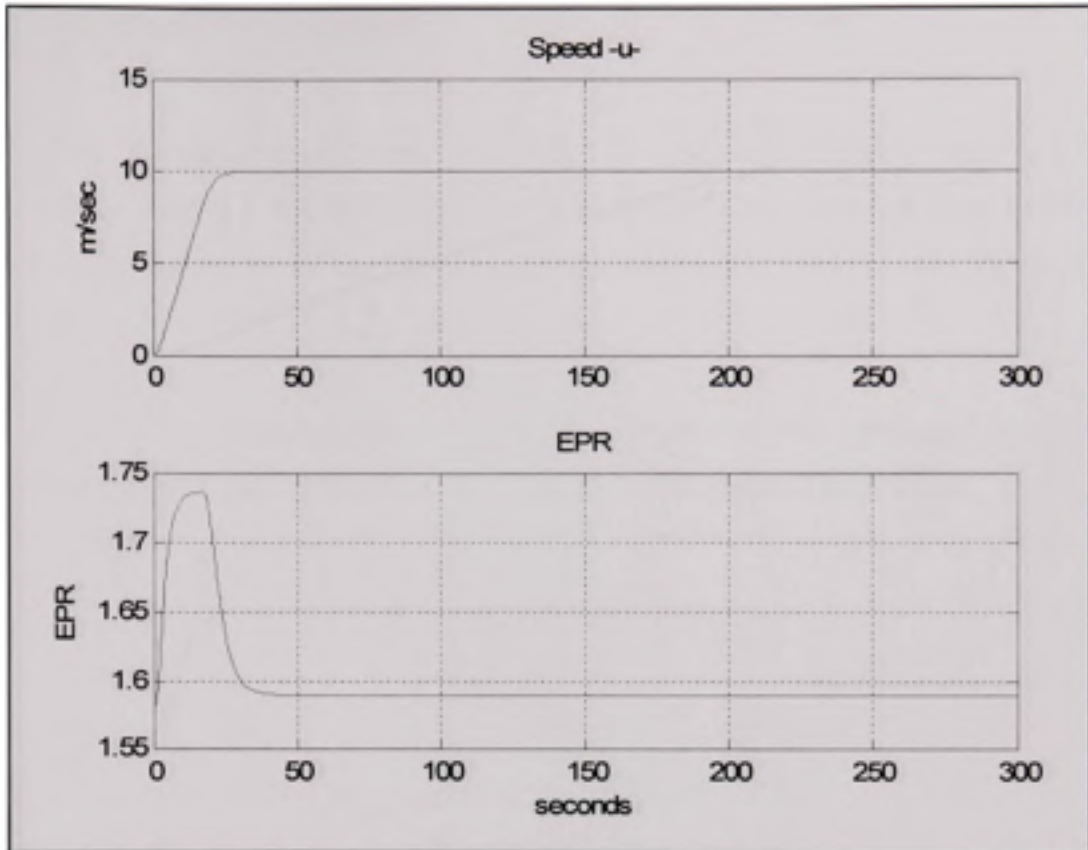


Figure 6.2 EPR and Speed responses with no EPR limit saturation.

We now try to achieve the same speed command, but we have modified the EPR limits to $EPR_{\min}=1$ and $EPR_{\max}=1.6$ to highlight the behavior of the protection control loop. The results are presented in Figure 6.3. We notice that the EPR saturates at 1.6 with no overshoot and maintains that value till the speed command is reached 200 seconds later, and then settles at the same level of 1.59.

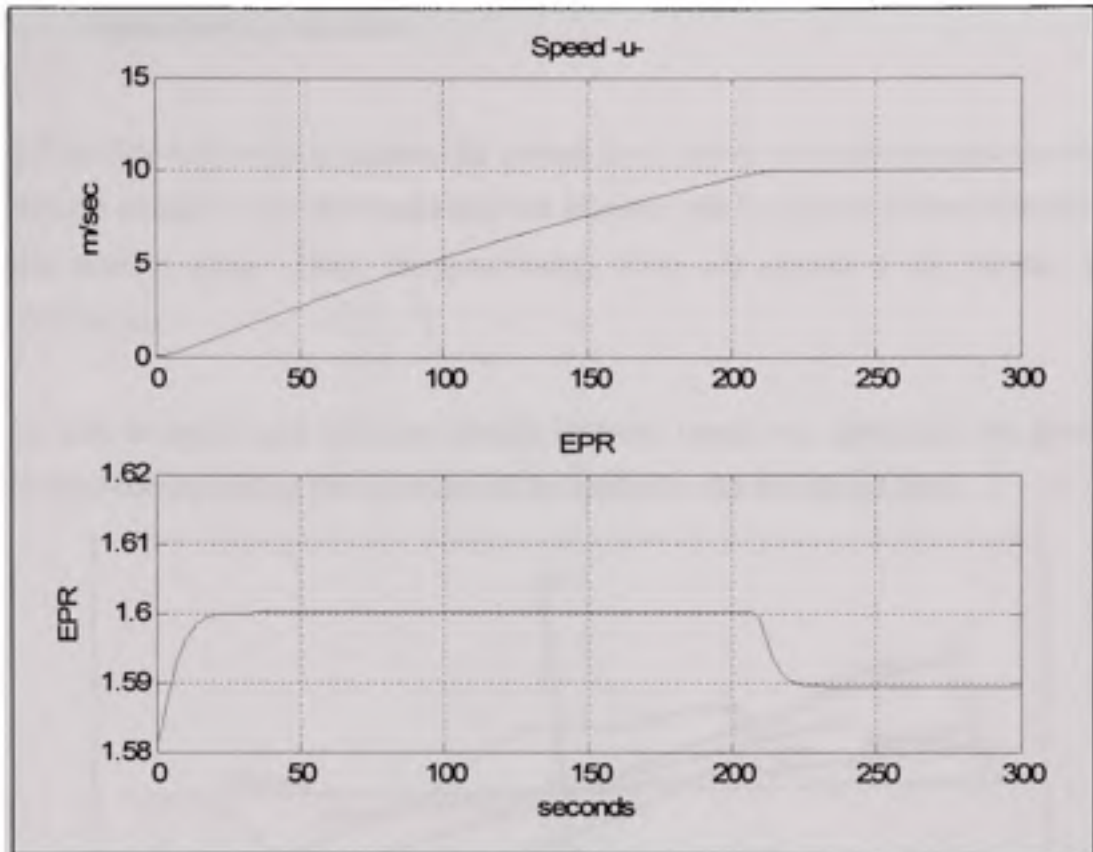


Figure 6.3 EPR and Speed responses with EPR limit saturation.

6.2 Throttle position and rate limits

At the output, the throttle position will also be limited in rate and position. The system presents two types of limits: physical or pre-defined limits that are characteristic of the throttle actuator, and a calculated or processed limit that reflects the speed limits to guarantee operation of the TC within the speed limits. For the first type of limitation, rate is limited to 10% of maximum position per second.

For the second type of limitation, the speed limits are used to calculate throttle position limits and then applied to the throttle position command. This is further elaborated in the next section.

6.3 Speed limits protection

As indicated in the previous chapter, the control laws provide a throttle position to command the throttle actuator. First, the maximum rate allowed will be applied to limit the rate of the throttle position signal. Then, the speed safety limits are applied to the throttle actuator position signal.

To be able to apply such limits, a relation between thrust and speed must be given. The normalised corresponding thrust limits will be applied to the throttle position.

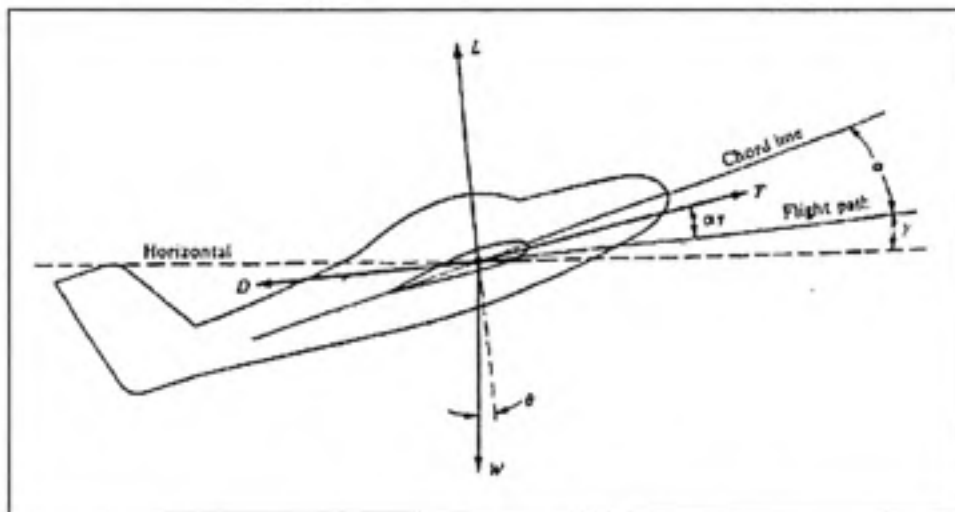


Figure 6.4 Forces applied to aircraft [20].

The required relation is obtained using the second law of Newton applied to the airspeed (flight path) direction axis as seen in Figure 6.4 considering the longitudinal movement in the vertical plane only. Since the aircraft is considered to be in steady flight cruise altitude, no sideslip or bank angle are present.

Therefore:

$$m.a_x = T \cos(\alpha_r) - D - mg \sin(\gamma) \quad (6.1)$$

where T is the thrust, and D is the drag and a_x the longitudinal acceleration.

Using the relations given by:

$$T = \eta T_M \quad (6.2)$$

where T_M is the maximum thrust, and η is the normalized throttle position.

The drag expression can be elaborated as a function of the drag coefficient C_D , air density ρ , the wing surface area S (5,600 square feet or 524.9 m² for a B747), and True Airspeed (TAS) V .

$$D = \frac{1}{2} \rho S V^2 C_D \quad (6.3)$$

Thus,

$$m.a = \eta T_M \cos(\alpha_T) - \frac{1}{2} \rho S V^2 C_D - mg \sin(\gamma) \quad (6.4)$$

Finally:

$$\eta = \frac{m.a + \frac{1}{2} \rho S V^2 C_D + mg \sin(\gamma)}{T_M \cos(\alpha_T)} \quad (6.5)$$

Furthermore,

$$\alpha_T = \alpha - \varepsilon \quad (6.6)$$

with α the angle of attack and ε the angle between the engines axis and the aircraft chord line. $\varepsilon = 2.5^\circ$ in this specific case.

The speed limits are used to find the corresponding throttle position limits to be applied. The actual acceleration given by the sensors is used. For speed, VMO and MMO will be used and compared. MMO will be used to calculate the corresponding indicated airspeed. The lower limit between the two maximum speeds will be used as the maximum position allowed.

The minimal speed will be used to obtain the minimal position allowed. These limits will then be applied to the position signal.

However, during real-time application of this method, we noticed important variation and instability in the speed limits. The fast variation observed in the speed limits caused the command to oscillate and introduced important instability.

An alternative similar to the control scheme used in the EPR limit control can be designed and tested. In fact, we may use the speed controller designed previously in chapter 4, with speed limits as the reference input, as a parallel reference controller that replaces the main controller whenever the throttle command exceeds the command outputted by the speed limit control loop. The design and selection logic is similar to the design presented in section 6.1. Results of such an application are further presented and discussed in section 8.2.5.

CHAPITRE 7

INTEGRATION AND SYSTEM INTERFACE

7.1 The Dynamic Test Bed

7.1.1 System Overview

The DTB is used to stimulate the FMS with real aircraft environment and conditions. The main objectives are to reduce the number of flight tests by getting flight test credit by the certification authority; prepare and run the entire engineering and certification flight-test before the real flight-test; and to reproduce customer problems with the same conditions under the DTB. Systems testing can be performed under the DTB, e.g.: evaluation and test of the dynamic performance of the Flight Management System (FMS); act as a Vertical Navigation development test bed; demonstrate operation of all FMS modes for customers or other interested parties.

7.1.2 Dynamic Test Bed Overview

The DTB is a sophisticated flight simulation system, which is able to simulate the experience of flying an aircraft as closely and realistically as possible. The DTB can be used to not only stimulate an individual aircraft component (such as FMS) for design/testing purposes, but also perform various flights and engineering tests.

The DTB can be used:

- To stimulate the FMS with a real aircraft environment and conditions

- To perform systems tests, e.g. evaluation and test of the dynamic performance of systems.
- To reproduce customer problems using real-life environmental conditions
- To prepare and run the entire engineering and Transport Canada Civil Aviation (TCCA) flight-test before the real flight-test
- To reduce the number of flight tests by getting flight-test credit from certification authority.

The DTB is divided into three sections: Aircraft Simulation (Flightsim), Hardware or Hardware Stimulation Models (Part Task Trainer – PTT), and an Instructor Operational Software (IOS) or Graphical User Interface (GUI). A High-level block diagram H/W and S/W are shown in Figure 7.1 and 7.2 respectively:

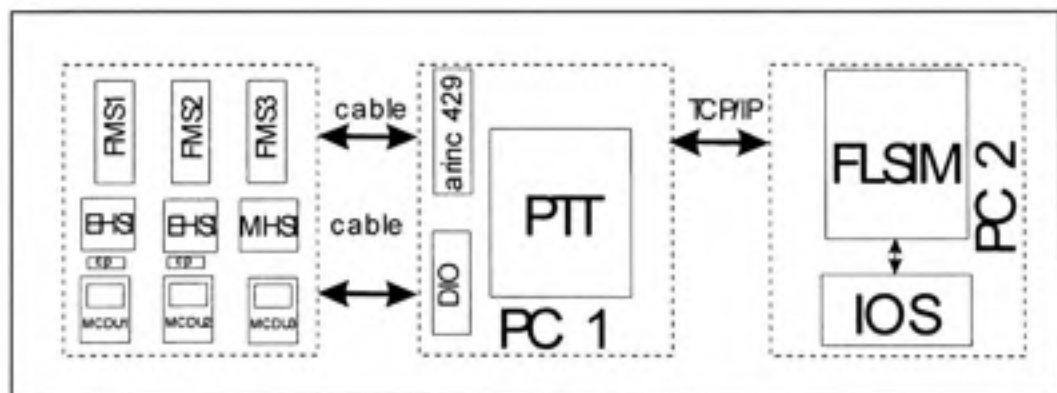


Figure 7.1 High Level H/W Block Diagram [7].

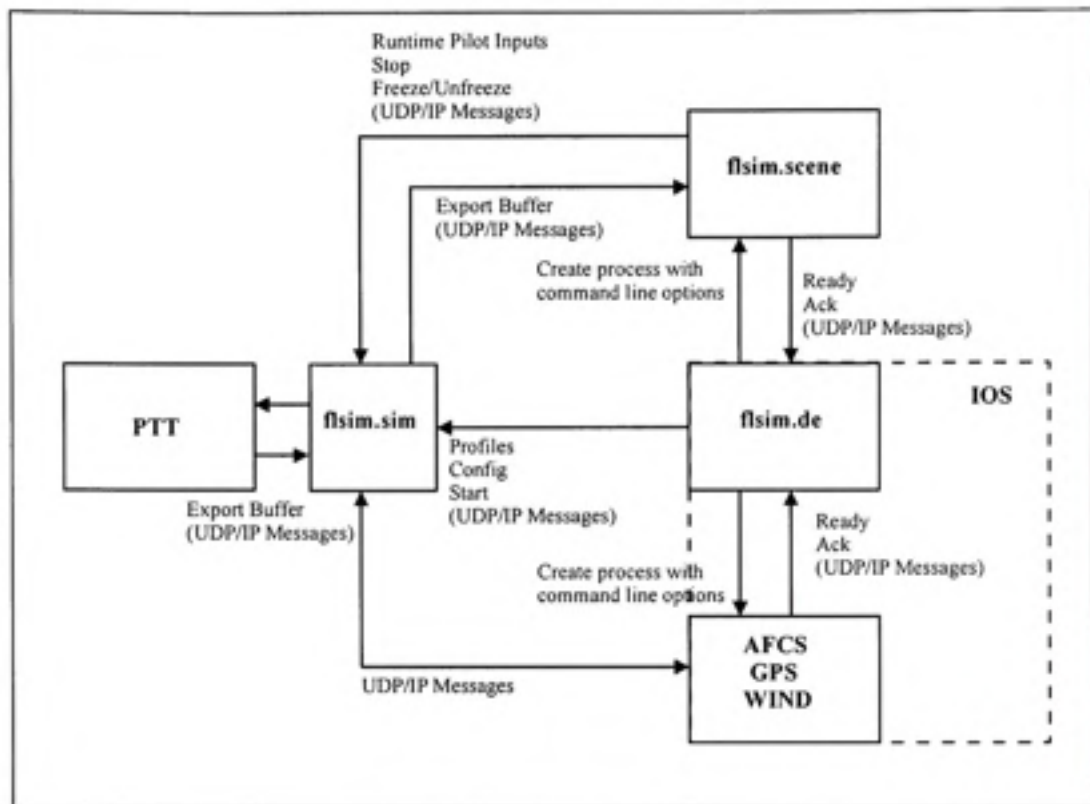


Figure 7.2 S/W Block Diagram [7].

7.1.2.1 Aircraft Simulation

The DTB configuration presents a Boeing 747-400 aircraft configuration. It is the aircraft model used throughout our study.

The aircraft simulation is provided by Flightsim. Flightsim is a software package that lets you enter parameters describing the performance of a fixed-wing aircraft, and then use those parameters to test how that aircraft will perform.

Flightsim's purpose is to act as a high-fidelity flight performance model. It is a highly flexible tool. Not only can it be used for flight training, but also to test aircraft designs, or to test how existing aircrafts might perform under simulated conditions that can be controlled. The main strength of Flightsim's performance is its flexibility. The flight model can be quickly changed through dialog boxes or through modifying data curves.

The aircraft simulation includes the earth, ambient air conditions, winds system, Nav aids, AFCS, flight control, undercarriage, engines, weight and balance system, equations of motion and aerodynamics models. The flight model provided presents the particularity of being expandable and modifiable. System models can be individually developed, modified and added to the simulation model.

The flight model also communicates with external processes and systems, in our case the PTT and IOS or GUI via the interface cards with UDP/IP protocol or shared memory, as shown in Figure 7.2 and 7.3.

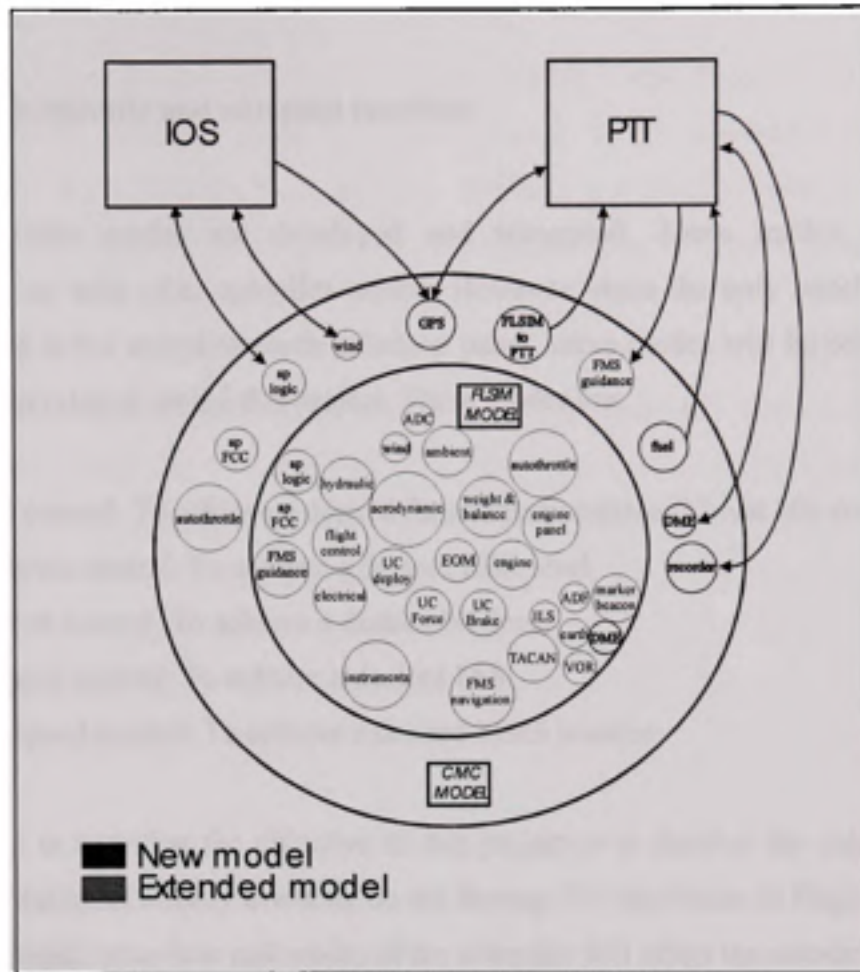


Figure 7.3 Flightsim Simulation Diagram [7].

7.1.2.2 The Thrust Control System

The development of the thrust controller will include modifications and additional developments on the default Flightsim model. It will expand and add to the functionalities and properties of the autopilot logic and system integrated in the flight model.

7.2 Authothrottle and autopilot interface

Many autothrottle modes are developed and integrated. These modes can be used independently or with other autopilot modes. However, since the only panel available for mode selection is the autopilot mode selection panel, these modes will be selected through the new GUI developed within this project. These modes are:

1. Thrust control: To achieve a desired thrust lever position (Thrust idle command)
2. EPR thrust control: To achieve a desired EPR level
3. N1 thrust control: To achieve a desired N1 level
4. IAS speed control: To achieve a desired IAS
5. Mach speed control: To achieve a desired Mach number

It is important to note that the objective of this project is to develop the autothrottle to be used with the autopilot already available on the Boeing 747 simulation in Flightsim. It is then clear that the logic, behaviour and modes of the autopilot will affect the autothrottle and vice-versa. The autopilot modes are well defined and integrated, and certain modes will require the coupling of the autothrottle with the autopilot. These modes could not be achieved unless the autothrottle and autopilot are coupled. They are presented hereafter:

1. Flight level change: This mode uses the autothrottle to increase or decrease thrust while maintaining certain airspeed. The Thrust variation will cause an altitude variation. If the flight level change mode is activated, then the Autothrottle needs to capture a certain EPR level (or N1 level), and the autopilot maintains a certain airspeed using the elevators. This mode couples the autothrottle thrust mode with the autopilot airspeed hold mode. Once the desired Altitude level is achieved, the autopilot switches to Altitude hold mode and the autothrottle to speed mode.

2. **Airspeed capture and hold, using thrust:** During airspeed capture, the autothrottle increases or decreases thrust to capture a certain airspeed. Meanwhile the autopilot needs to be in altitude hold mode to maintain a desired altitude using the Pitch command. This mode couples the autothrottle Airspeed mode with the autopilot Altitude Hold mode.

While integrating the autothrottle, it is important to integrate the new modes within the functioning structure of the Autopilot logic. Therefore, the new autothrottle modes are added to the logic to be able to select them independently. Also, in case one of the coupled modes is selected, the autothrottle modes are called and integrated within the respective modes, and once the level achieved a selection and transition is done to fall back on the required autopilot or autothrottle mode. All the required integration and modifications are done in the autopilot logic processor.

7.3 System inputs

7.3.1 Flightsim systems

Flightsim continuously provides all required aircraft and flight information to the TC model during the simulation. Such data are IAS measurement, EPR measurement, Air data such as pressure and speed of sound for MACH number calculations and TAS corrections. Also all data of other inputs such as the PTT (FMS) or the GUI are received through Flightsim Flight Model.

Table 7.1 indicates all the variables needed to perform the Thrust Controller computations.

Table 7.1
Thrust Controller Input Variables

Variable Label	Variables description	Data Structure	Units
autopilot.ats	Alimentation status	Flsim_Electrical_Output	N/A
data.ats_mode	Autothrottle selected mode	CMC_AFCS_Logic_Output	N/A
engine[],pla	Thrust position	Flsim_Engine_Panel_Output	%
epr	Engine pressure Ratio	Flsim_Engine_Output	N/A
thrust	Total engine thrust	Flsim_Engine_Output	N
alpha	Angle of attack	Flsim_Adc_Output	rad
cos_alpha	Cosine of angle of attack	Flsim_Adc_Output	N/A
gamma	Flight path angle	Flsim_Adc_Output	rad
sin_gamma	Sine of Flight path angle	Flsim_Adc_Output	N/A
speed_true	True Airspeed	Flsim_Adc_Output	m/sec
speed_true_dot	True airspeed rate	Flsim_Adc_Output	m/sec ²
speed_ind	Indicated Airspeed	Flsim_Adc_Output	m/sec
mach_num	Mach number	Flsim_Adc_Output	N/A
ambient_temp	Air temperature	Flsim_Ambient_Output	K
ambient_pres	Barometric pressure	Flsim_Ambient_Output	Pa
ambient_dens	Air density	Flsim_Ambient_Output	Kg/m ³
underlying_area.t heta	Pitch	Flsim_Eom_Output	rad
underlying_area. alt	Altitude	Flsim_Eom_Output	m
body.a_abs_cm	Accelerations vector	Flsim_Eom_Output	m/sec ²
body.v_abs_cm	Velocity vector	Flsim_Eom_Output	m/sec
ned.euler	Angular vector	Flsim_Eom_Output	rad
ned.euler_vel	Angular rates vector	Flsim_Eom_Output	rad/sec

7.3.2 PTT (FMS)

The TC model receives from the PTT (FMS) the envelope of flight, including upper and lower speed limits, ceilings, and engine limits (EPR limits).

Engine limits are foolproof safeguards to prevent stall and overboost of the engine for all modes and flight conditions. It limits the forward and aft throttle motion. The forward position is set to avoid exceeding the maximum allowable engine pressure ratio or temperature. The aft position closely corresponds to the flight IDLE thrust value. These values are a function of altitude and temperature. The performance and optimization computer calculates these values and makes them available for the TC via the PTT (FMS).

Upper speed limits include VMO (Max Operating Speed IAS) and MMO (Mach Max Operating Speed). Also, the TC will maintain the speed over the Minimum Operating Speed to avoid stall. VMO, MMO and Minimum Operating Speed will all be received for different flight conditions by the PTT (FMS).

Also, when the AFCS is in FMS Guidance mode, the Autopilot and Thrust Controller are asked to capture a trajectory calculated by the FMS. The TC will then receive from the PTT (FMS) the reference speed to be captured. The inputs are not received directly from the PTT (FMS), but through Flightsim. The PTT (FMS) sends the information to Flightsim which then makes them available for the Thrust Controller.

7.3.3 GUI

The GUI represents the AFCS and mode selector panel. The GUI allows the pilot to activate the desired Speed mode (IAS or MACH), which is needed to initiate the capture logic. It also allows the pilot to input the reference speed that is desired to attain. The GUI used to trigger the different AFCS modes and speed inputs is shown in Figure 7.4.



Figure 7.4 CMC AFCS GUI.

The inputs are not received directly from the GUI, but through the FLSIM. The GUI sends the information to Flightsim which then makes them available for the Thrust Controller.

7.3.4 Additional Inputs

The actual AFCS Control panel does not support EPR requests from the pilot for the Thrust modes. A new interface process was developed to meet such needs. The TC will receive the requested EPR from the process. It will also be possible to select EPR upper and lower limits manually via the interface.

As cited earlier, the thrust controller will receive loop gain values that are manually selected or from a file. This will facilitate tests and design modifications as well as gain tuning and thus make the TC system easily adaptable to other environments.

7.4 System outputs

7.4.1 Outputs to Flightsim

The TC sends a Throttle position to the flight model. This command will act on the engines throttles servo.

Table 7.2
Thrust Controller Output Variables

Variable Label	Variables description	Data Structure	Units
req_throttle[i]	Requested Thrust	Flsim_Autothrottle_Output	%
theta_c	Pitch demand	Flsim_Ap_Fcc_Output	rad

7.4.2 GUI

The GUI will receive any error message or mode annunciation from the TC. If the TC can not achieve the desired airspeed due to limits and constraints, the system will annunciate it as discussed in the previous sections.

The messages output on the GUI are presented in Table 7.2.

Table 7.3
GUI Output messages

Scenario	Message
Desired EPR is above limits	EPR reference above limit, EPR reference adjusted to maximum value
Desired EPR is below limits	EPR reference below limit, EPR reference adjusted to minimum value
Desired speed is above VMO limit	Speed reference above VMO limit, speed reference adjusted to VMO value
Desired speed is above MMO limit	Speed reference above MMO limit, speed reference adjusted to MMO value
Desired speed is below minimum limit	Speed reference below minimum operating speed, speed reference adjusted to minimum value
EPR mode armed	EPR armed
EPR mode captured	EPR captured
IAS mode armed	IAS armed
IAS mode captured	IAS captured
Mach mode armed	MACH armed
Mach mode captured	MACH captured

7.5 System code and development

The system was initially designed in Matlab for a linear system. The design presented the linear controller architecture, but not the selection logic or output messages.

To integrate the controller within the non linear aircraft model, the code development was done in several steps:

- 1- Controller coding:
 - a. PID generic code
 - b. Different controllers gains and loops (EPR, N1, IAS, Mach...)
 - c. Limits protections code
- 2- Interface coding
 - a. Interfacing with the inputs and outputs of Flightsim
- 3- Mode selection logic coding
 - a. Adding the different modes to the autopilot logic
 - b. Modifying the autopilot logic code to incorporate new coupled modes
 - c. Coding a new GUI for new modes selection and definition of the communication messages.
- 4- Messages, alerts and warnings output function coding

Respecting these steps during the coding process permitted the testing and integration of each feature independently and effectively. The system was coded using C++ language to be integrated within the Flightsim environment. Once all the code integrated, system tests could be run within the simulation, by selecting the appropriate modes and command values, monitoring the messages outputs and the controller outputs and performance.

7.6 Integration issues & limitations

During the integration process of the proposed controller, many technical issues arose which limited the functionality and final outcome of the controller. In this section we discuss further the limitations and the way they affected the outcome of the proposed design and development.

7.6.1 Pitch inner loop

The initial proposed controller was a multi-input state feedback controller. The inputs of the controller are speed and altitude while the outputs are elevator deflection and thrust lever angle. During the integration, it has been noticed that the elevator deflection could not be controlled directly in Flightsim, and that the pitch channel could only be controlled through the pitch inner loop, and hence through a pitch command. This caused a problem to the integration of our controller. Two solutions were available. The first solution would see us eliminate the pitch inner loop and replace it with the pitch channel of the state feedback modern controller. However, if this solution was to be chosen, the whole autopilot would require modification, a new design approach and the required integration. This approach goes against one of our requirements, which is to be able to add new development, with easy integration, that doesn't affect the functioning of the existing features. It would also require extra development that would require time and resources not available for this project.

The second possible solution would integrate the Thrust channel without affecting the Pitch channel since it is already stabilized using the inner loop. However, we would lose the multi-input controller features and pitch-thrust coupling. The integration of the Thrust channel controller as developed in the state feedback controller also has no added value if integrated alone without the pitch channel, since it would represent a proportional controller using airspeed as feedback on the thrust channel, nothing more. If this solution is to be chosen, then a more complex controller on the Thrust channel should be used such as a Proportional-Integral-Derivative controller, to guarantee a fast and stable response that would not affect the Pitch channel.

The controller we developed for integration within Flightsim supports the two possible solutions. However if the first solution is to be used, then further development is required. The previous autopilot design has to be redesigned using the modern state feedback controller theory. The developed and integrated controller does provide the elevator and

thrust commands, but these commands are not yet integrated and used by Flightsim. This would be a potential topic for further studies or developments.

The second solution is also integrated, and a Thrust channel PID controller is used to guarantee a fast and stable response. Similarly to the Engine Pressure Ratio controller, the Thrust channel controller controls a Speed input using a thrust command. The PID controller gains are tuned during the real time simulation.

7.6.2 PID controller integral term

During the simulation, the first order approximation of the integral function in the PID controller caused unstable behaviour, and the controller output diverged whenever an important Integrator gain was chosen. It would have been possible to explore higher orders of integral approximation; however simulation results proved that to be unnecessary. Satisfactory results were obtained using only the proportional and derivative terms in the controller only. Stability and system response were satisfactory and the integral term was eliminated from the controllers.

CHAPITRE 8

SYSTEM TEST AND VALIDATION

8.1 Qualification standards and guidelines

DO-178B: “Software Consideration in Airborne Systems and Equipment Certification” [27] will be used as a standard and guideline in the qualification process of the Thrust Controller Model. The DO-178B standards cover the complete product development lifecycle from project and product planning to final testing and system validation.

DO-178B classifies Airborne Software into two categories: Development Tools, and Verification Tools. DO-178B defines development tools as “tools whose output is part of airborne software and thus can introduce errors” while it defines verification tools as “tools that cannot introduce errors, but may fail to detect them.”

According to the definition, the Thrust Controller Model implemented on the DTB is classified as a verification tool since its outputs won't modify nor introduce errors into the airborne software operating the DTB (i.e. Flightsim 9.1).

8.1.1 The need for Tool Qualification

According to Section 12.2 of the DO-178B, “qualification of a tool is needed only when processes described in DO-178B are eliminated, reduced, or automated by the use of that tool without its output being verified as specified in DO-178B Section 6”.

In other words, if the results of the TC Model tool are being relied on to supply the sole evidence that one or more objectives are satisfied, the tool is required to be qualified. If the output of the tool is verified by some other means, then there is no need to qualify the tool.

Notice N8110.49 “Software Approval Guidelines” [27] released by the FAA on June 2nd 2003 provides three questions to be asked in determining whether a tool has to be qualified or not.

1. Can the Model tool allow an existing error to remain undetected?
2. Will the Model’s output not be verified per section 6 of DO-178B?
3. Are processes of DO-178B eliminated, reduced, or automated?

In the case of the TC Model, the answer of all three questions is “yes”. Therefore qualification is required.

8.1.2 Qualification Requirements

Section 12.2.2 of DO-178B states that verification tools should be qualified by “demonstration that the tool complies with its Tool Operation Requirements under normal operation conditions.”

We shall then prove that the TC model meets all requirements cited in chapter 4.1. The test plans, procedures and results will be needed for qualification data and documentation. Data showing that all of the requirements in the Tool Operational Requirements have been verified should be documented and available for certification review. Sufficient verification data are needed to demonstrate normal operation only.

8.1.3 Software Integration Tests

Once the development of the TC Model is completed it will be integrated into the DTB software system. This integration will be tested to ensure that the model operates without any errors with the DTB software system, mainly Flightsim. Those tests include:

- Input from the system to the model
- Output from the model to the system

Those test deal with the TC Model as a “black box” they don’t check how the input or the output is received or released, they only check if an input is actually received and if an output is actually released.

8.1.4 Requirements Tests

The operational requirements of the TC will be tested under normal conditions. These tests will go into all the details of the model’s architecture and coding. Requirements to be tested are:

- Thrust capture
- Speed capture
- EPR constraints and engine limits
- Speed and acceleration limits and placards
- Flight envelope validity
- Pitch coupling

The explanation of those requirements is given in section 4.1.

8.1.5 Code Testing

Since the TC Model is classified as a Verification Tool, unit code testing is not required for validation or qualification.

8.1.6 Documentation

The required qualification documents aren't the subject of this study and therefore won't be discussed. However, we should know that in order to produce the required documentation, test reports including the following are required:

- Operational requirement tested
- Test case/scenario
- Expected results
- Obtained results
- Corrective actions (if any)

8.2 TC Model Functionality Test Procedures

This section will verify that all the functionalities of the TC model have been met according to the Software Requirements Document for the TC Model, presented in Annex II.

The autothrottle has requirements that were set forth in the Software Requirements Document for the TC Model. These requirements include EPR capture and EPR limits protection, speed capture with speed limits protection, as well as pitch coupling.

In order to assure that the testing is performed properly and all requirements are met, different test sections will be set up.

8.2.1 Thrust Idle

This test case will perform requirement check of the Thrust idle Mode's ability to decrease thrust until idle position is attained. Figure 8.1 shows that Thrust idle (5%) is achieved within 20 seconds. Thrust decreases from around 68% to 5%, a variation of 63%, in just above 15 seconds.

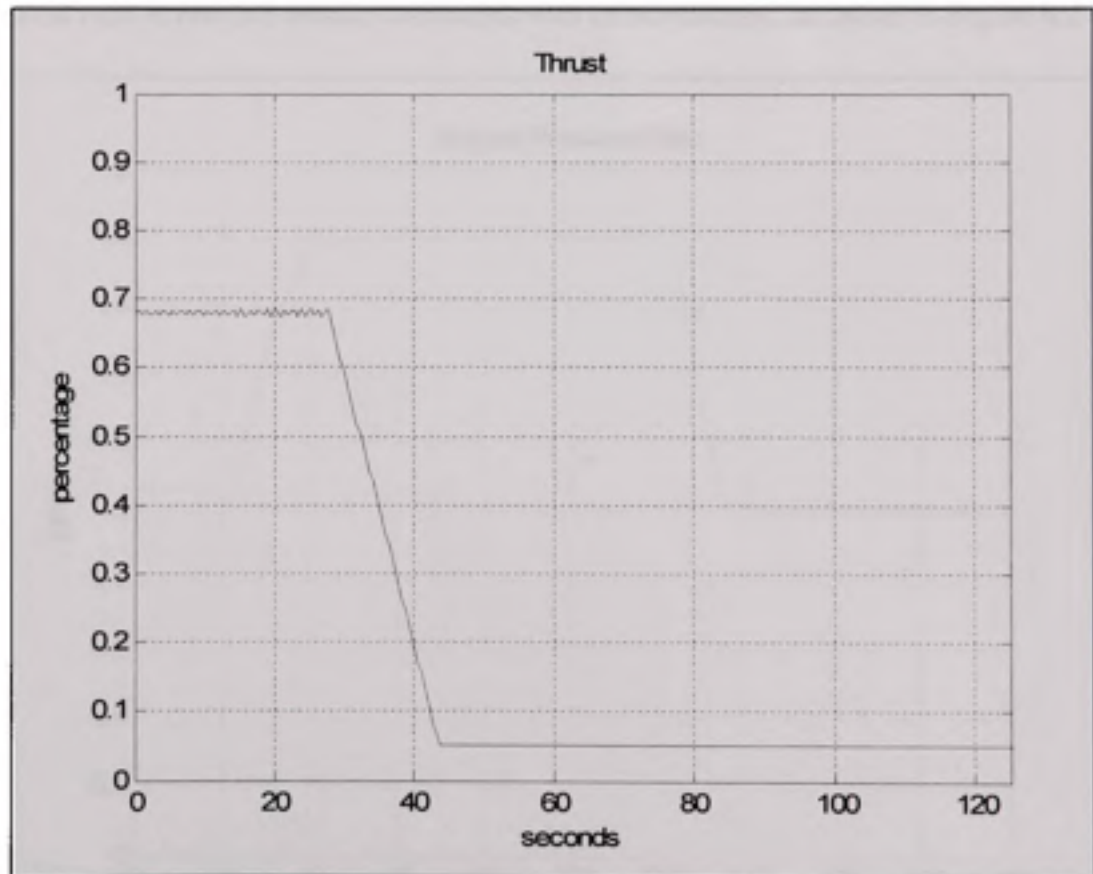


Figure 8.1 Thrust variation for a Thrust idle command.

8.2.2 EPR Arm and Reference Limit Protection and EPR Capture

This test case will perform requirement check of the EPR Arm Mode's ability to respect the EPR limits for EPR reference inputs and the functioning of EPR Arm mode logic. During this test, EPR maximum is set to 1.8 and minimum value is set to 1. The initial EPR request is set to 3. We have noticed the system output message during the simulation "EPR request over maximum allowed. EPR request adjusted to maximum EPR value". Figure 8.2 shows the EPR capture. The EPR command is adjusted to the maximum value of 1.8, and the EPR reaches that value within 30 seconds, with no overshoot. At 120 seconds, we then request an EPR of 0.5, and again the EPR request is adjusted to acceptable limits (EPR=1) and the requested EPR is reached within 20 seconds with no downshoot, as shown in Figure 8.2.

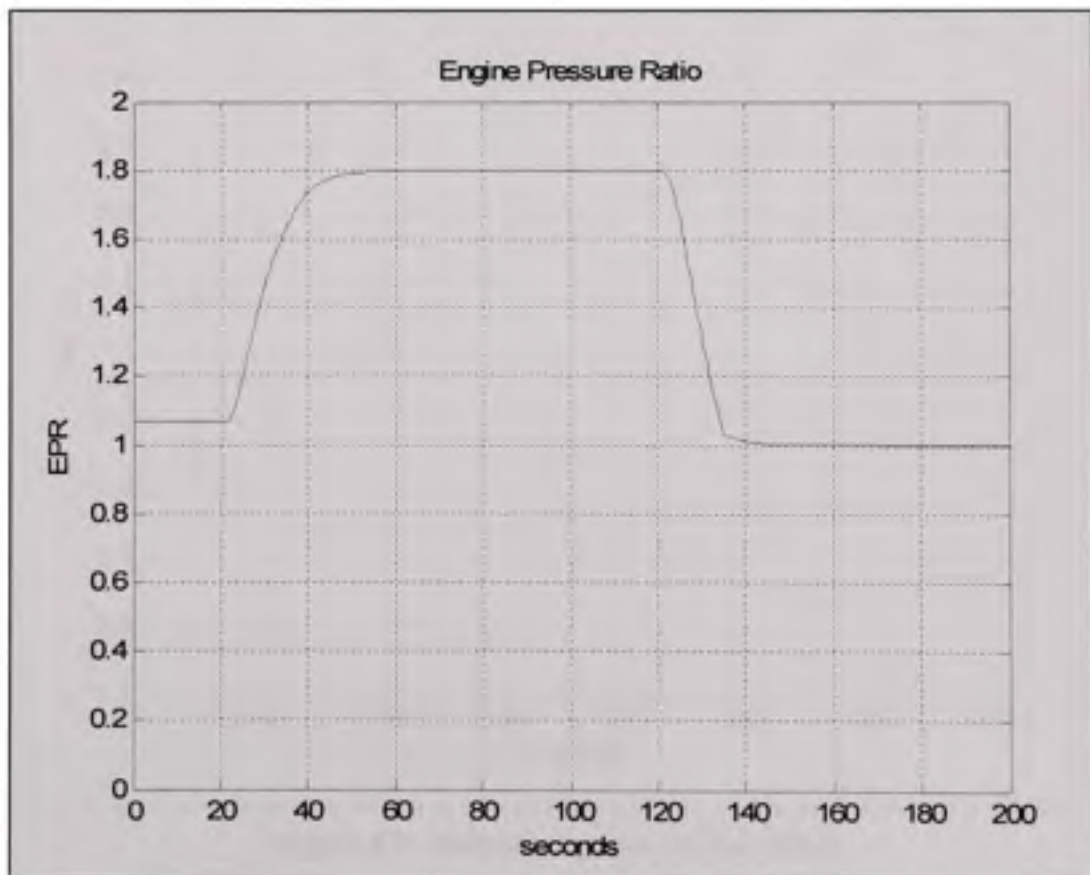


Figure 8.2 EPR capture during an EPR thrust command.

8.2.3 EPR Flight Level Change Climb

This test case will perform requirement check of the EPR Capture Mode's ability to increase thrust until aircraft captures a desired altitude. The increase of thrust will cause climb and level change and will be done at constant speed using the Autopilot IAS pitch control to maintain speed at desired level. Figure 8.3 shows the altitude capture, with no overshoot. The requested altitude is achieved within 250 seconds. Initially positioned at 28000 feet (8534.4 meters), the requested altitude is 33000 feet (10058.4 meters). The controller achieves 5000 feet within 250 seconds, a vertical speed of 20 feet per second (1200 feet per minute).

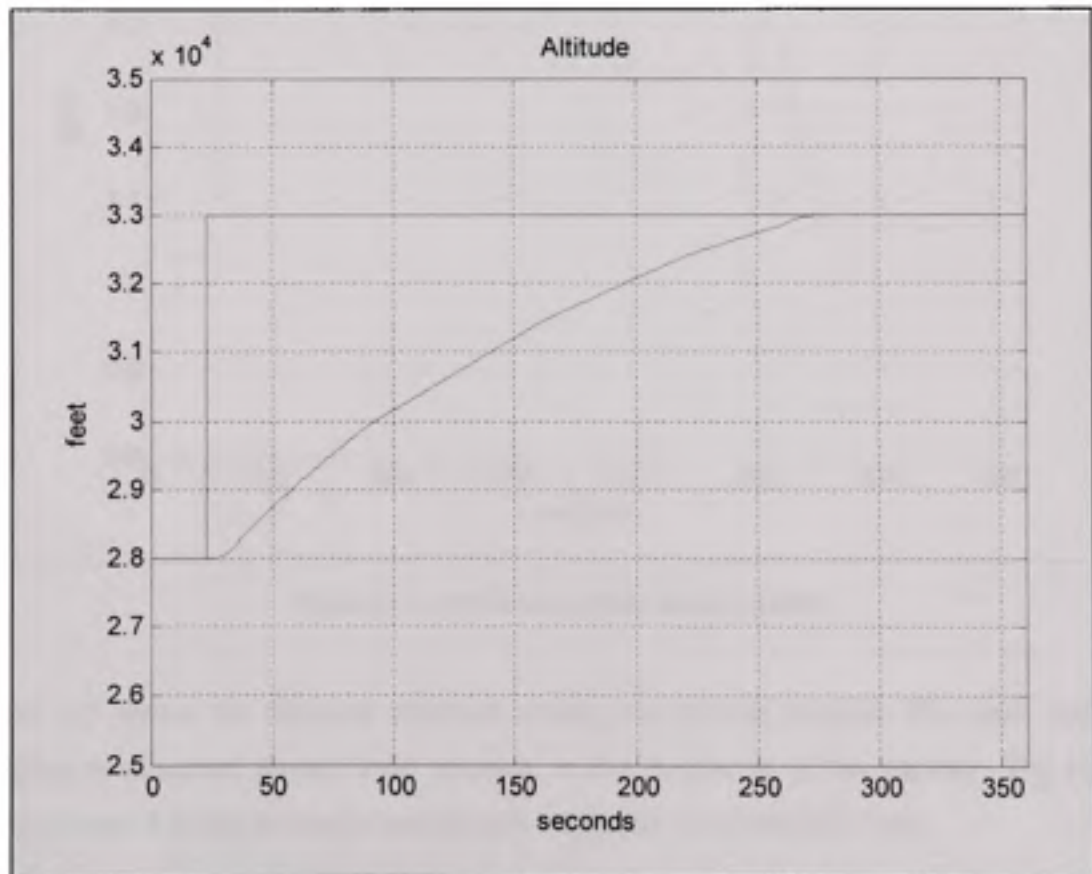


Figure 8.3 Altitude capture during climb.

The EPR rating used as reference for the Thrust controller is variable, and depends on the distance between the actual and requested altitude, as shown in Figure 8.4. The EPR reference variable values were chosen during tests to achieve the smoothest transition that didn't delay the capture time response.

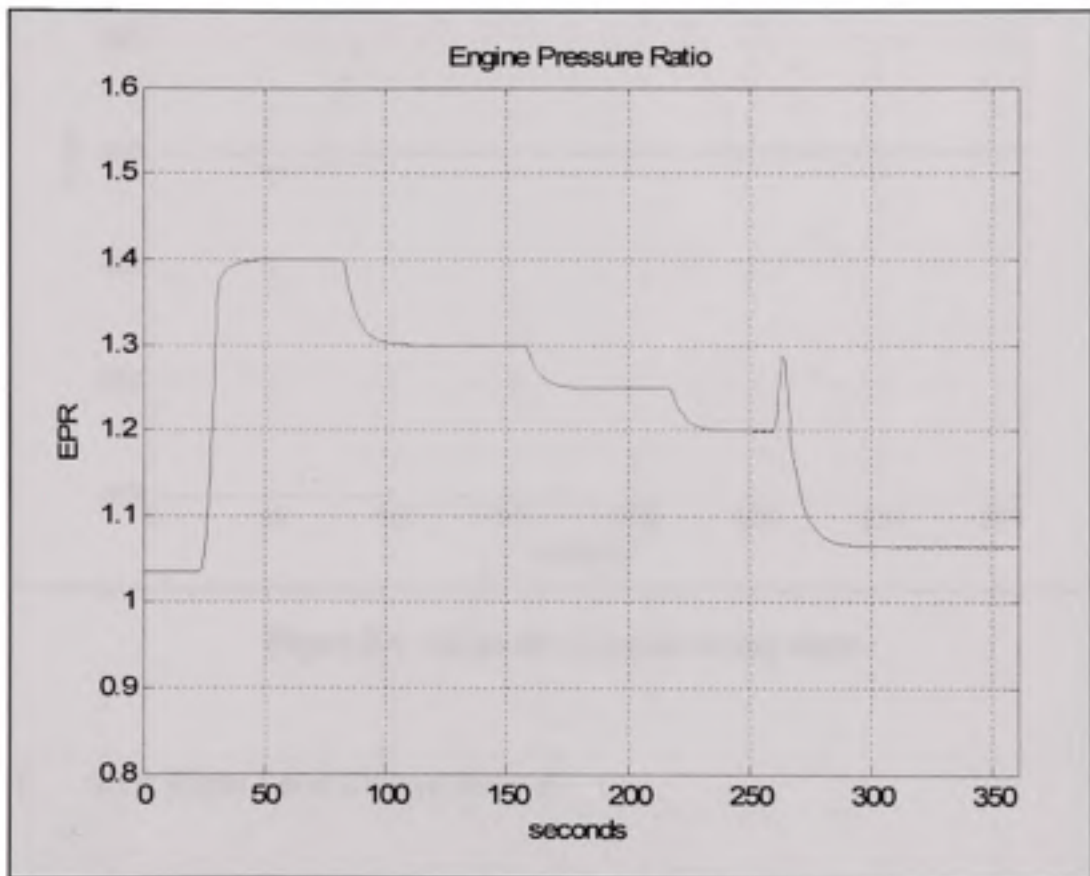


Figure 8.4 EPR variation during climb.

Figure 8.5 shows the airspeed variation during the altitude capture. The most important variation is observed around $t=50$ seconds, at the beginning of the capture. The airspeed varies around 1 m/sec at most which is well within the 1% allowable limit.

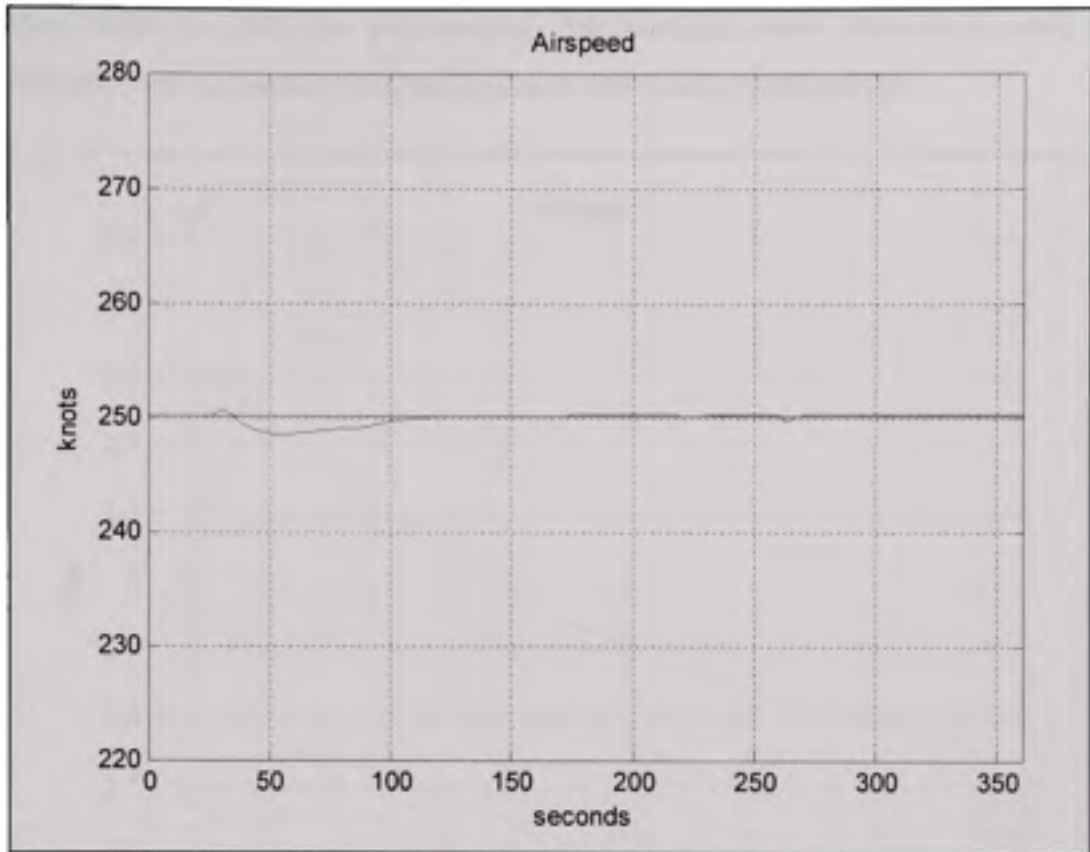


Figure 8.5 Airspeed variation during climb.

8.2.4 EPR Flight Level Change Descent

This test case will perform requirement check of the EPR Capture Mode's ability to decrease thrust until aircraft captures a desired predefined altitude. The decrease of thrust will cause descent and level change and will be done at constant speed, coupling the Autothrottle with the Autopilot IAS mode.

Figure 8.6 shows the altitude capture, with no downhoot. The requested altitude is achieved within 250 to 300 seconds. Initially positioned at 33000 feet (10058.4 meters), the requested altitude is 28000 feet (8534.4 meters). The controller achieves 5000 feet within 250 to 300 seconds, a vertical speed of around 20 feet per second (or as it is more often referred to in

aviation, 1000 to 1200 feet per minute). The vertical speed observed is well within expectations, and the capture time and response well within requirements.

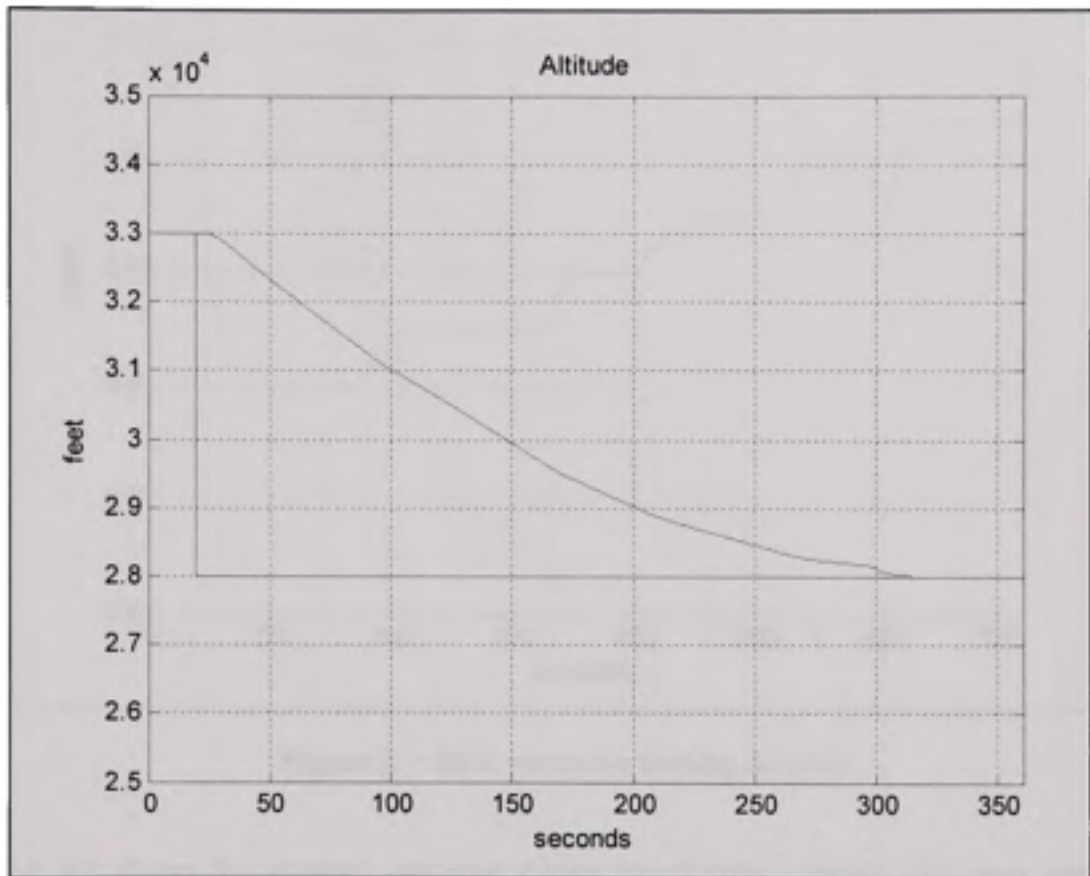


Figure 8.6 Altitude capture during descent.

The EPR rating used as reference for the Thrust controller is variable, and depends on the distance between the actual and requested altitude, as shown in Figure 8.7. The EPR reference variable values were chosen during tests to achieve the smoothest transition that didn't delay the capture time response.

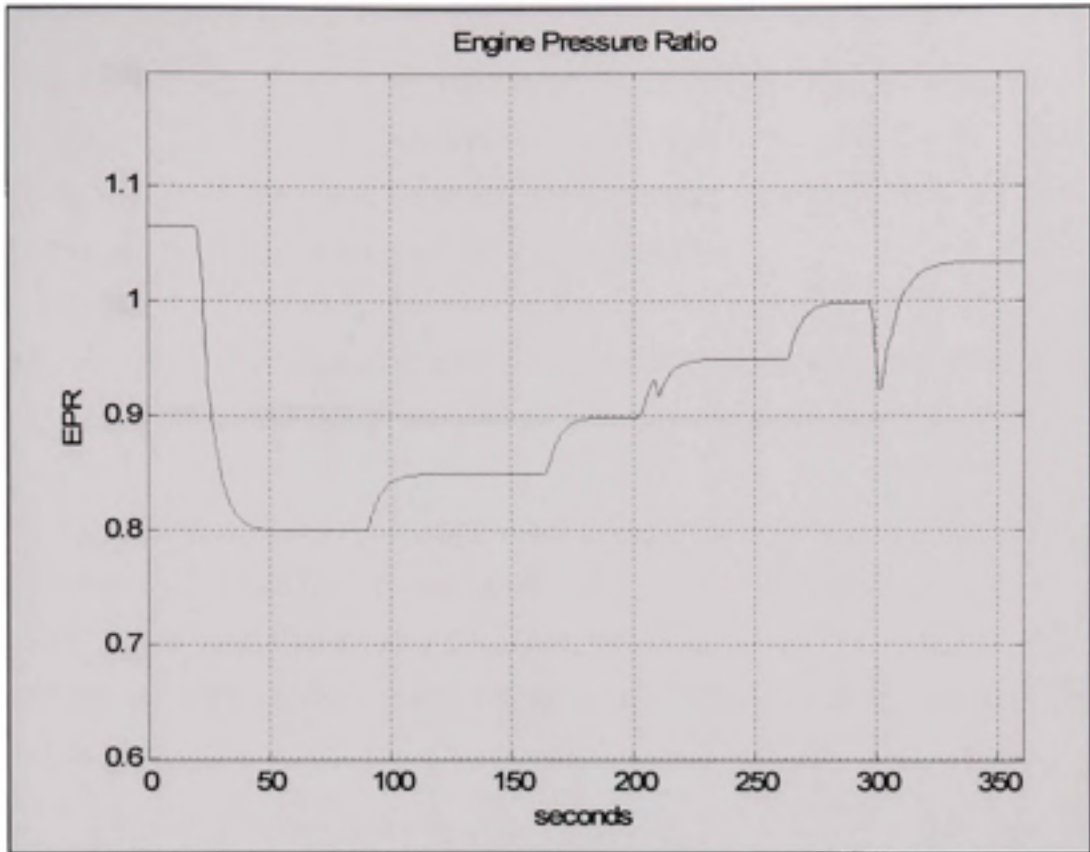


Figure 8.7 EPR variation during descent.

Figure 8.8 shows the airspeed variation during the altitude capture. The most important variation is observed around $t=50$ seconds, at the beginning of the capture. The airspeed varies around 1 m/sec at most which is well within the 1% allowable limit.

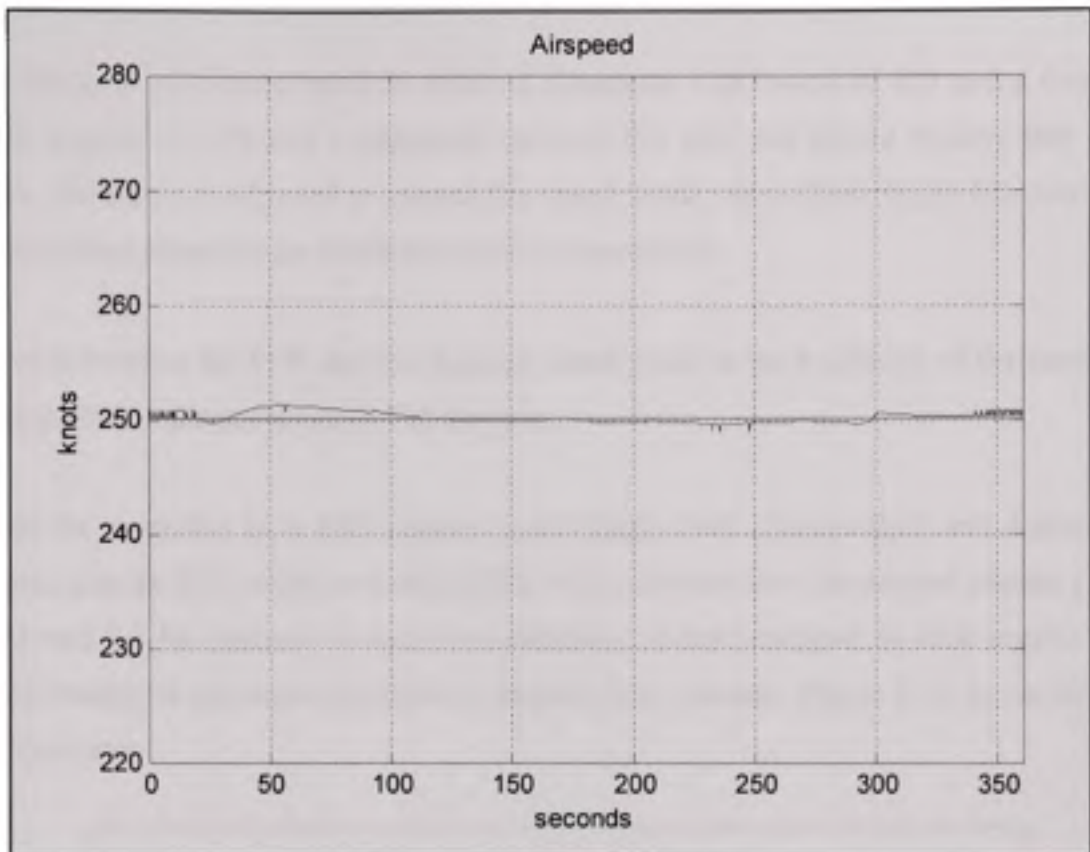


Figure 8.8 Airspeed variation during descent.

8.2.5 EPR Capture Speed Limits Protection

This test case will perform requirement check of the EPR Capture Mode's ability to respect speed limits. Maximum speed will be chosen to correspond to a lower thrust level than the EPR climb reference, and minimum speed will be chosen to correspond to a higher thrust level than the EPR descent reference. The EPR capture loop should not be able to achieve the reference EPR, while we observe the speed limitation.

During the climb phase, the EPR reference request is 1.4, while during the descent phase EPR reference is 0.8. EPR protection limits are 1.8 and 0.6. Thrust should never be operated outside of this range.

We choose to restrict the speed to within a Maximum VMO value of 180 m/s, a maximum Mach number of 0.75 and a minimum value of 120 m/s. We should observe that during climb, the thrust is adjusted to respect the speed limits, since these limits fall outside the normal thrust operation for climb and descent manoeuvres.

Figure 8.9 shows the EPR capture, with the climb phase at the beginning of the simulation, and the descent phase starting at 150 seconds.

While the controller is in EPR capture mode (flight level change climb and descent), we observe that the EPR settles at a value (EPR=1.07) different then the original request (1.4 for climb and 0.8 for descent). In fact, the controller can not command an EPR outside of the speed limitations and adjust the thrust to respect those placards. Figure 8.10 shows the speed capture curve

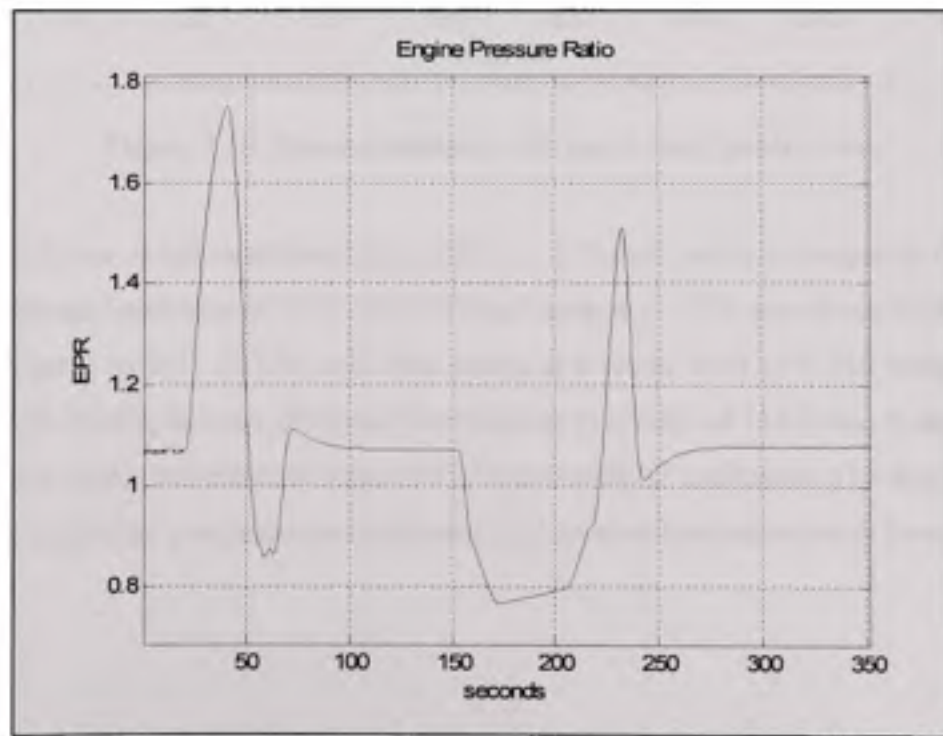


Figure 8.9 EPR variation during speed limit protection control.

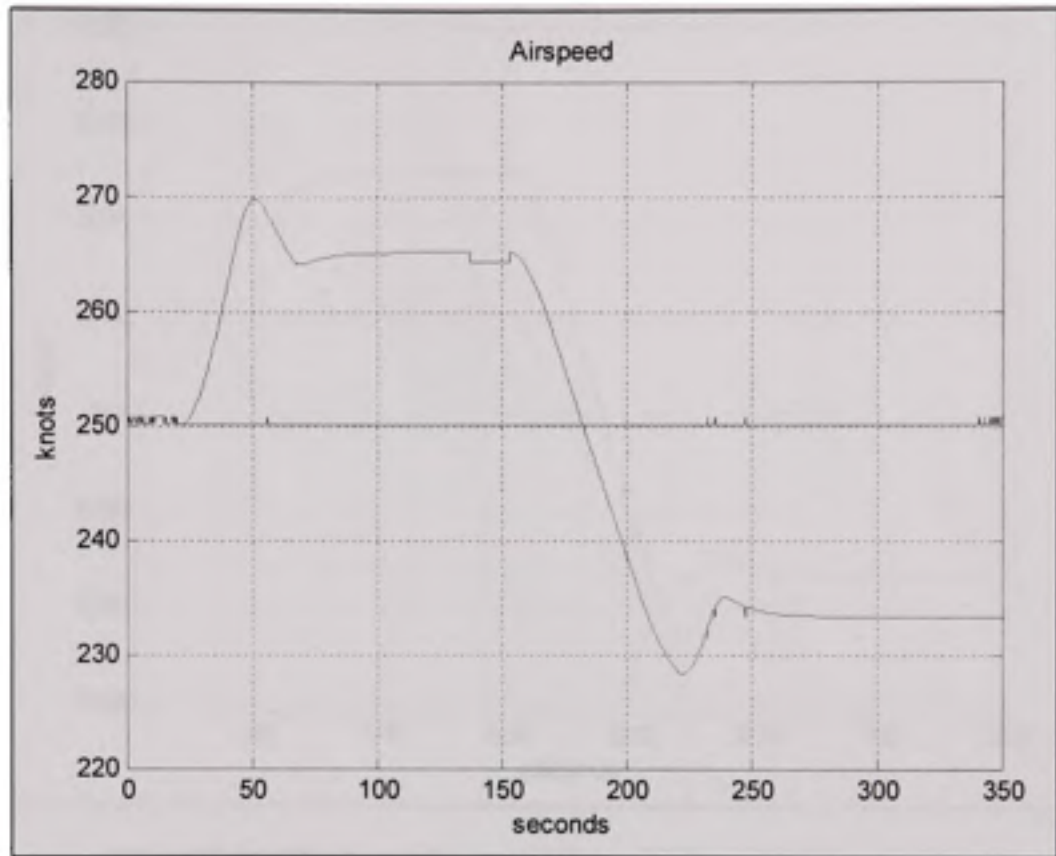


Figure 8.10 Speed variation with speed limit protection.

During climb, the speed overshoots then settles at 137 m/s, which corresponds to the Mach speed maximum limitation of 0.75 presented in Figure 8.11. The overshoot slightly reaches the MMO value of 0.75 (0.751) and then settles at a lower level of 0.745 which is within requirements. During descent, the speed downshoots to a value of 118 m/sec (a downshoot of 2 m/sec) then settles at minimum value of 120 m/sec with no oscillation. The downshoot falls outside the controller's requirements tolerance, and some enhancements may be requested.

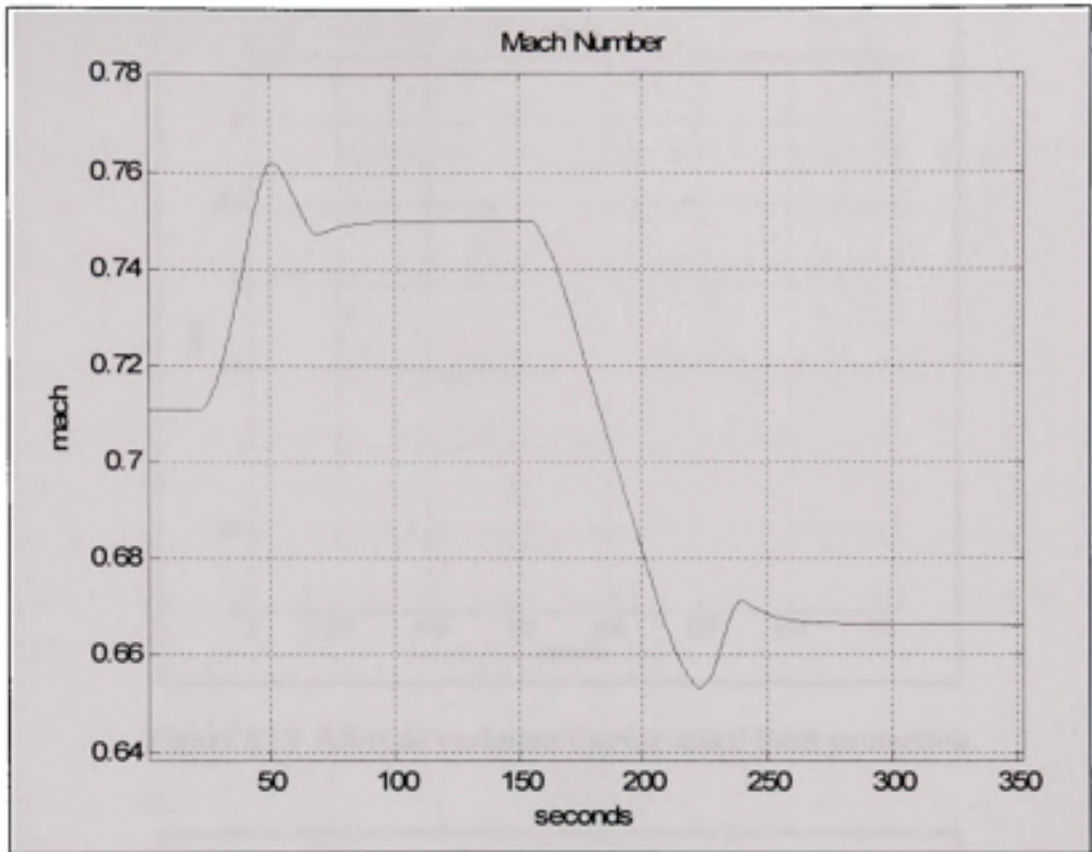


Figure 8.11 Mach number variation with speed limit protection.

During the speed and thrust variation, altitude is maintained at the initial level as shown in Figure 8.12. Figure 8.13 shows the acceleration variation which is well within tolerated G force values.

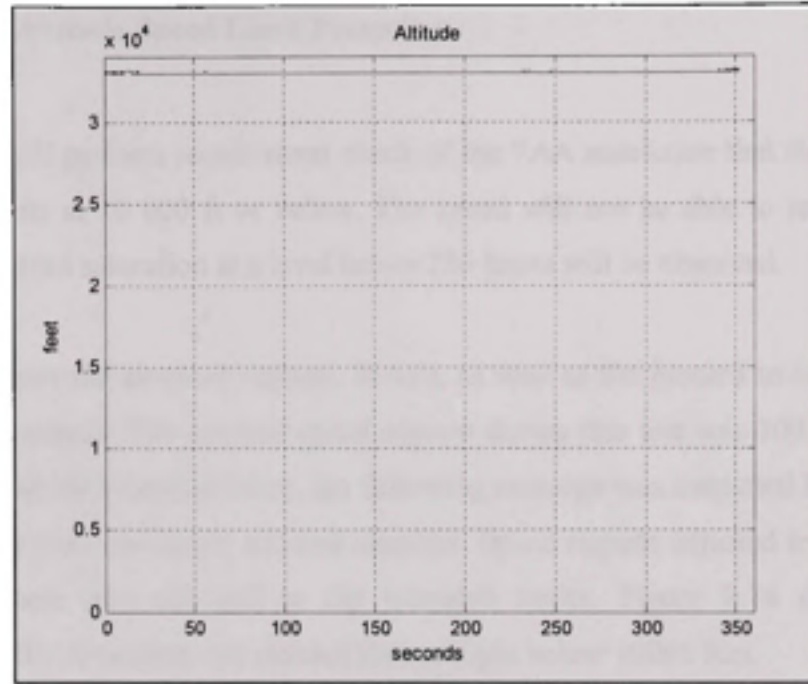


Figure 8.12 Altitude variation during speed limit protection.

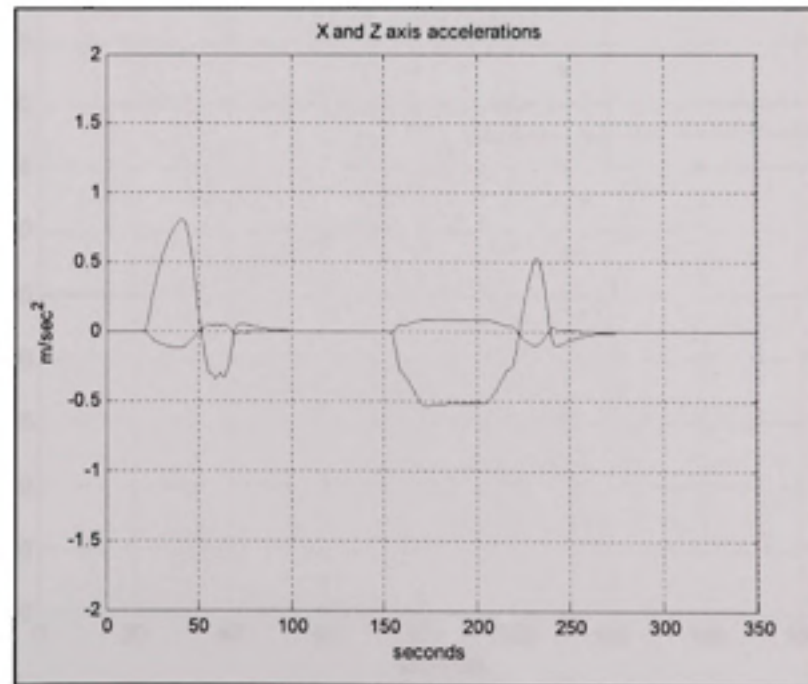


Figure 8.13 Accelerations variation during speed limit protection.

8.2.6 Low Altitude Speed Limit Protection

This test case will perform requirement check of the FAA restriction that the speed does not exceed 250 knots at 10 000 ft or below. The speed will not be able to reach a 300 knots reference and speed saturation at a level below 250 knots will be observed.

Figure 8.14 shows the airspeed capture, in m/s, as well as the placard to be respected (250 knots or 128.6 m/sec). The original speed request during this test was 300 knots. Since the request is outside the tolerated limits, the following message was outputted by the controller: "Speed Request over maximum allowed airspeed. Speed request adjusted to 250 knots", and the speed request was adjusted to the tolerated limits. Figure 8.14 demonstrates the controller's ability to respect that placard during flight below 10000 feet.

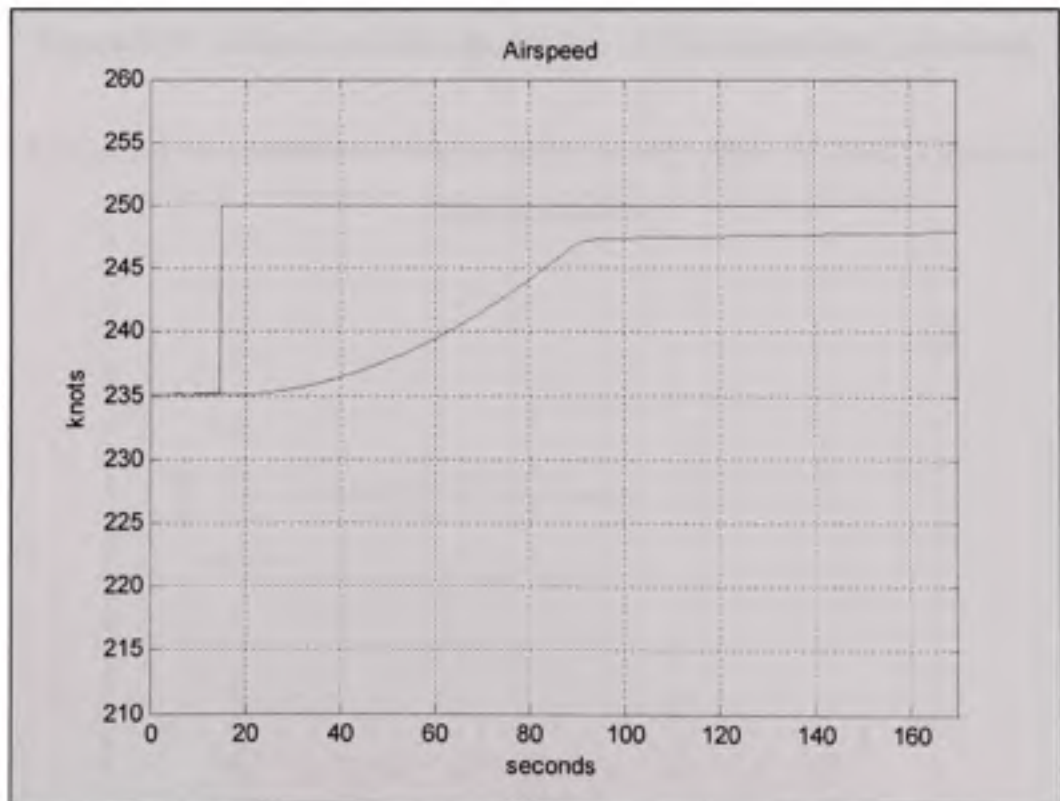


Figure 8.14 Speed variation during low altitude speed limit protection.

During that manoeuvre, altitude is maintained constant at 8000 feet as shown in Figure 8.15:

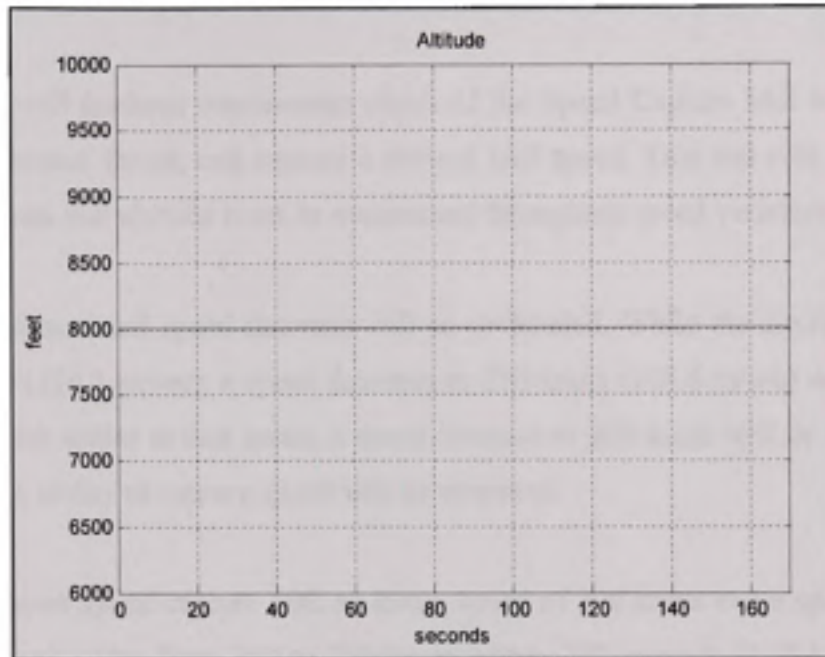


Figure 8.15 Altitude variation during low altitude speed limit protection.

Figure 8.16 shows the acceleration variation which is well within tolerated G force values.

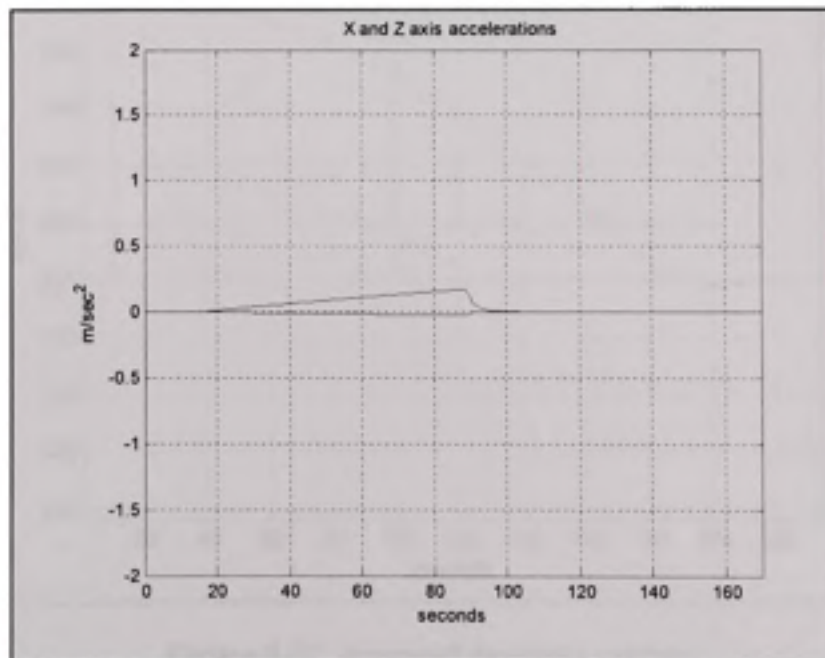


Figure 8.16 Acceleration variation during low altitude speed limit protection.

8.2.7 Speed Cruise IAS Capture

This test case will perform requirement check of the Speed Capture IAS Mode's ability to increase or decrease thrust, and capture a desired IAS speed. This test will be performed at Cruise conditions and altitude must be maintained throughout speed variation.

Both speed increase and speed decrease will be performed. While the aircraft at 33000 feet and 300 knots (154.3 m/sec), a speed decrease to 250 knots (128.6 m/sec) will be requested. After the aircraft settles at that speed, a speed increase to 300 knots will be commanded and the controller's ability to capture speed will be observed.

Figure 8.17 shows speed capture with an initial speed of 300 knots and a speed command of 250 knots. Speed varies from 300 to 250 knots within 200 seconds (0.25 knot per second), with no downshoot or oscillation.

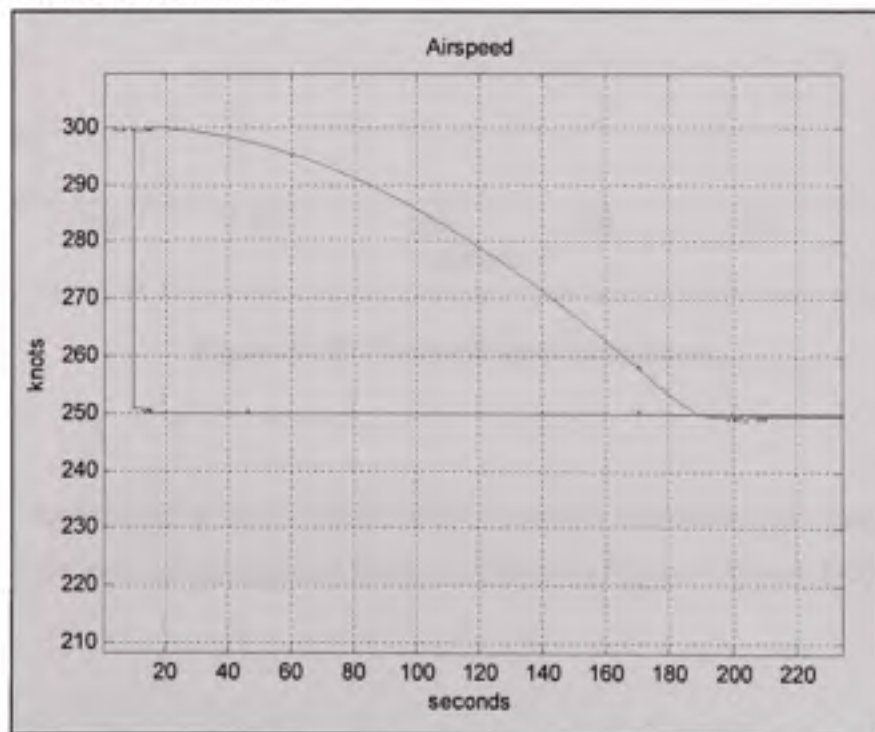


Figure 8.17 Airspeed decrease capture.

At 250 seconds, the new speed command is 300 knots (154.3 m/sec) and speed reaches the desired value within 200 seconds, also without any overshoot or oscillation, as shown in Figure 8.18.

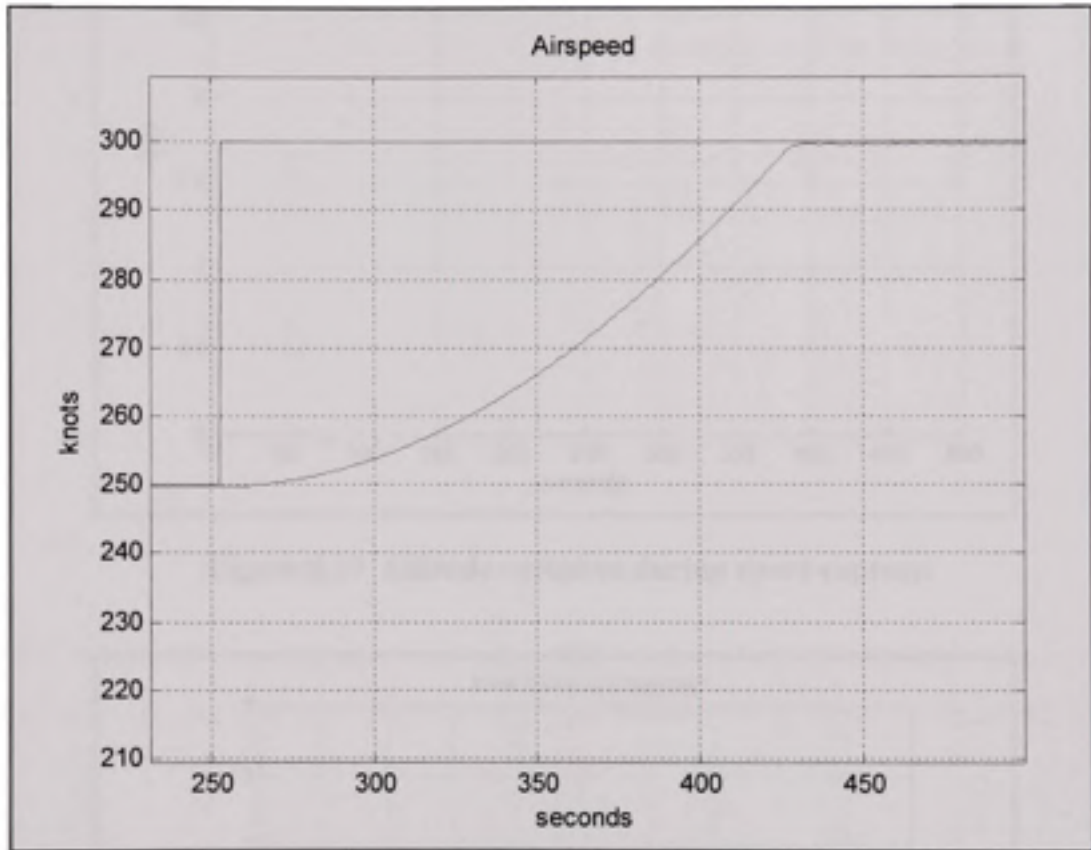


Figure 8.18 Airspeed increase capture.

Altitude was maintained at 33000 feet (10058.4 meters) throughout the speed capture, and accelerations are well within tolerated limits, as shown in Figure 8.19 and 8-20:

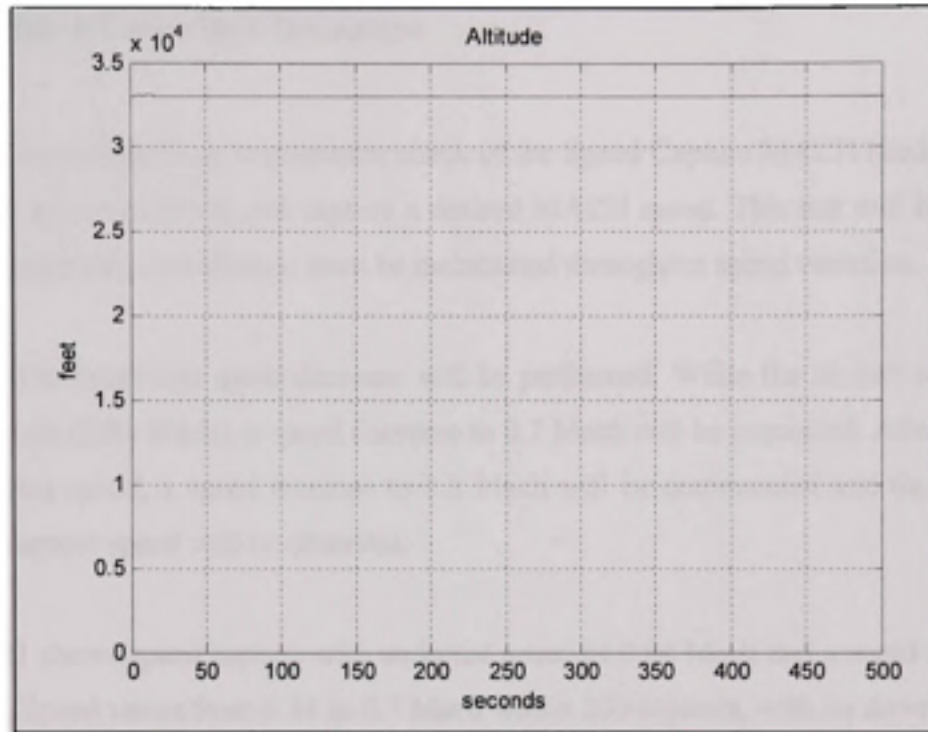


Figure 8.19 Altitude variation during speed capture.

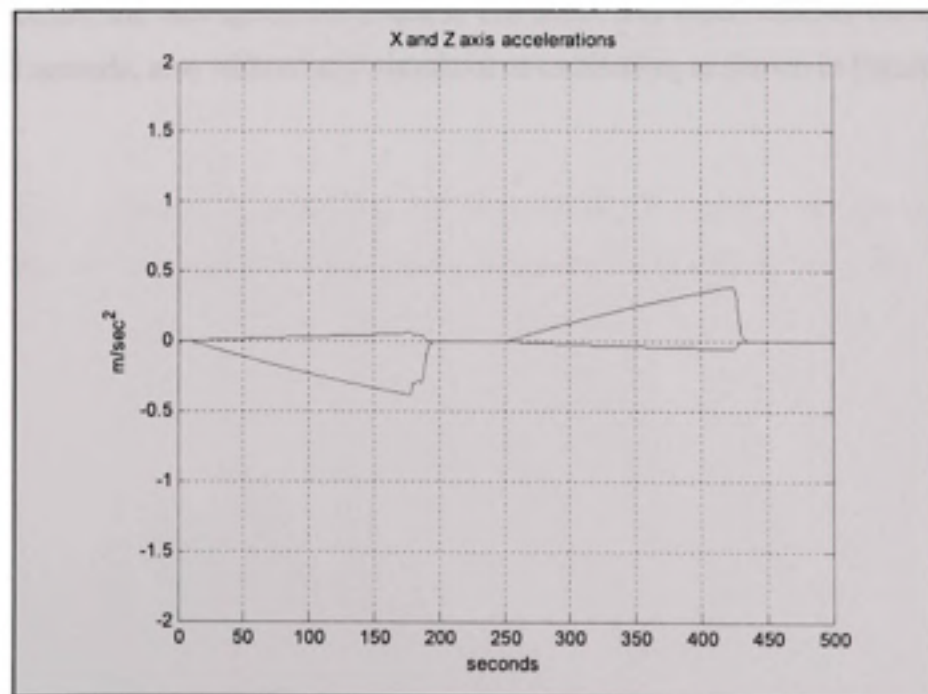


Figure 8.20 Acceleration variation during speed capture.

8.2.8 Speed Cruise MACH Capture

This test case will perform requirement check of the Speed Capture MACH Mode's ability to increase or decrease thrust, and capture a desired MACH speed. This test will be performed at Cruise conditions and altitude must be maintained throughout speed variation.

Both speed increase and speed decrease will be performed. While the aircraft at 33000 feet and 300 knots (0.84 Mach), a speed decrease to 0.7 Mach will be requested. After the aircraft settles at that speed, a speed increase to 0.8 Mach will be commanded and the controller's ability to capture speed will be observed.

Figure 8.21 shows speed capture with an initial speed of 0.84 Mach and a speed command of 0.7 Mach. Speed varies from 0.84 to 0.7 Mach within 200 seconds, with no downhoot, steady state error over 1% or any oscillation.

At 300 seconds, the new speed command is 0.8 Mach and speed reaches the desired value within 150 seconds, also without any overshoot or oscillation, as shown in Figure 8.21.

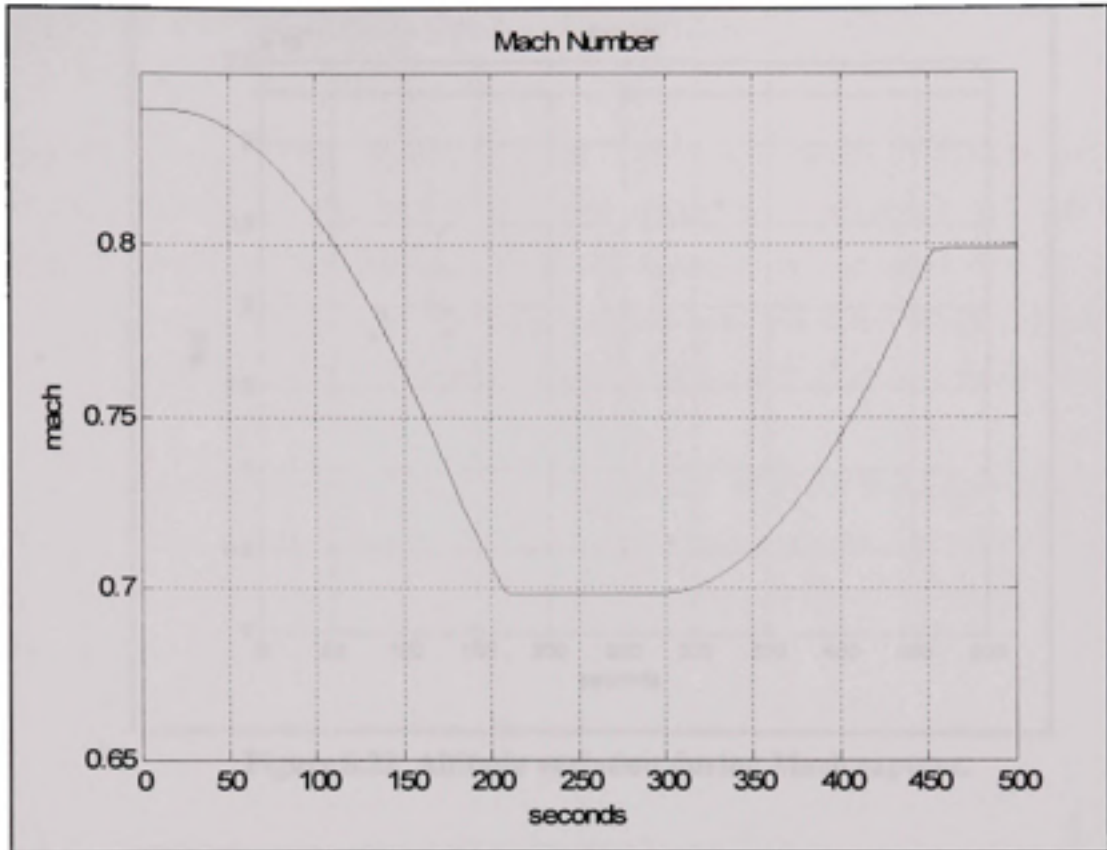


Figure 8.21 Mach speed capture.

Altitude was maintained at 33000 feet (10058.4 meters) throughout the speed capture, and accelerations are well within tolerated limits, as shown in Figure 8.22 and 8-23:

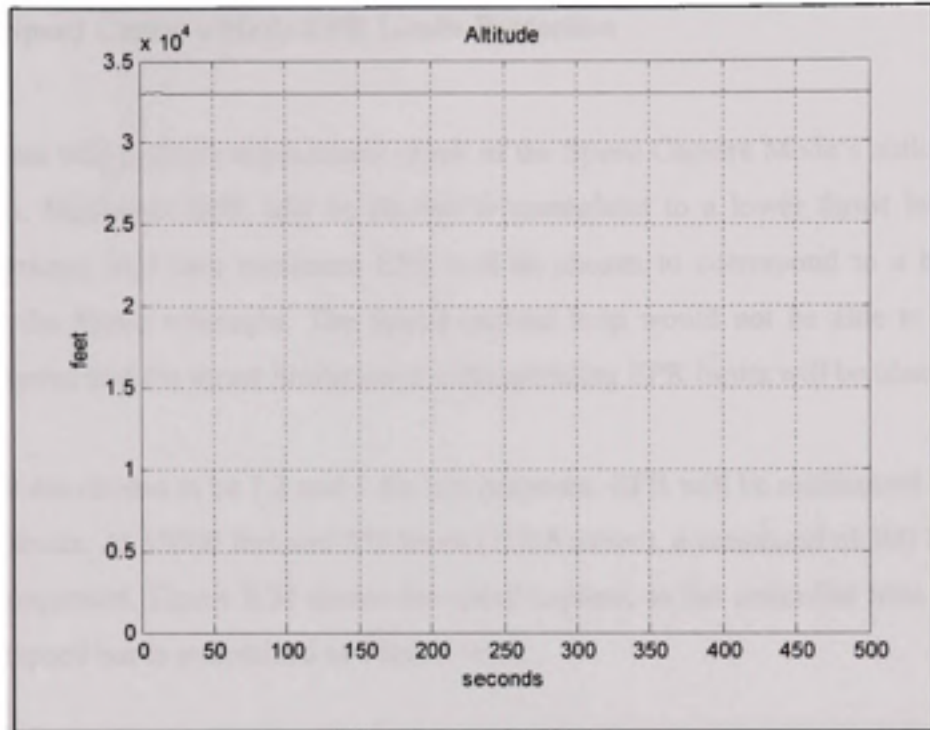


Figure 8.22 Altitude variation during Mach capture.

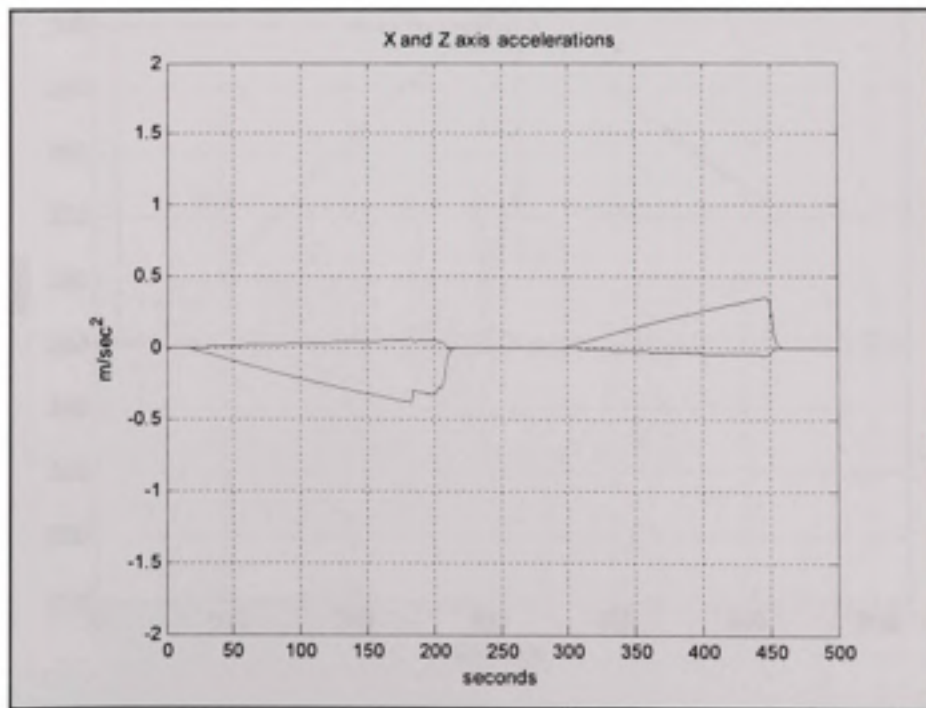


Figure 8.23 Acceleration variation during Mach capture.

8.2.9 Speed Capture Mode EPR Limits Protection

This test case will perform requirement check of the Speed Capture Mode's ability to respect EPR limits. Maximum EPR will be chosen to correspond to a lower thrust level than the Speed reference, and then minimum EPR will be chosen to correspond to a higher thrust level than the Speed reference. The Speed capture loop would not be able to achieve the reference speed and the thrust limitation at corresponding EPR limits will be observed.

EPR limits are chosen to be 1.2 and 1 for test purposes. EPR will be maintained within these operating limits. At 33000 feet and 250 knots (128.6 m/sec), a command of 300 knots (154.3 m/sec) is requested. Figure 8.24 shows the speed capture, as the controller tries to reach the desired airspeed but is maintained at a lower level:

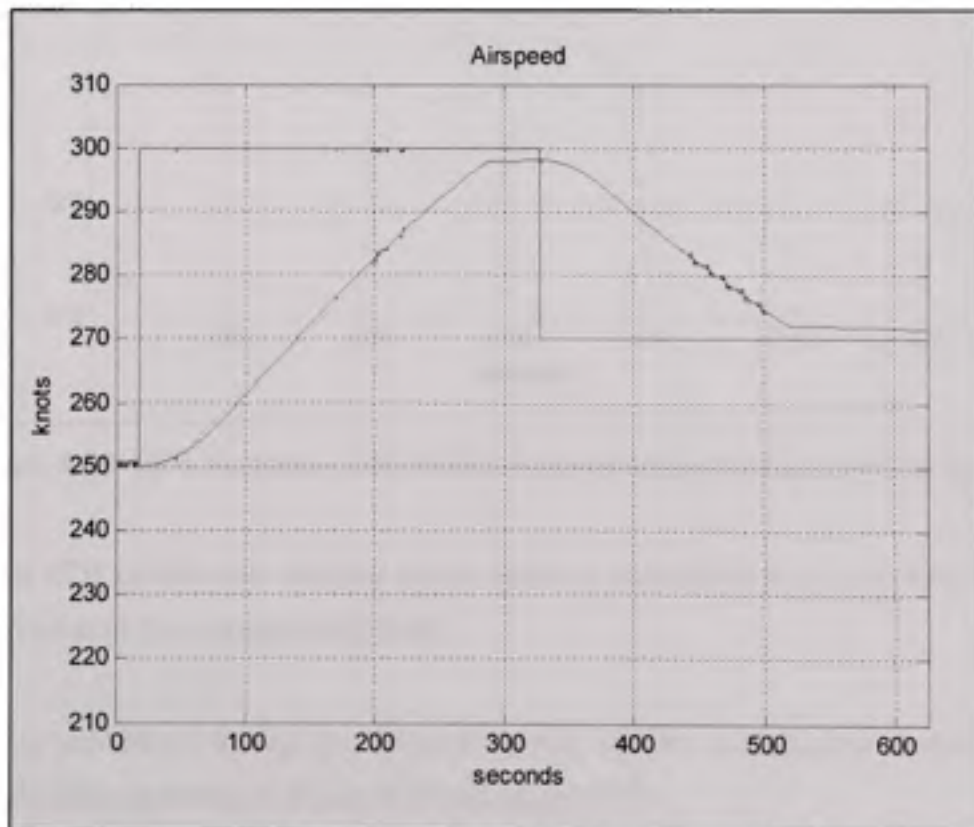


Figure 8.24 Airspeed capture.

At 320 seconds, a speed command of 180 knots (92.6 m/sec) is requested. Again, Figure 8.24 shows that the airspeed settles at a higher level.

The limits at which the airspeed settles correspond to the EPR limits of 1.2 and 1 as shown in Figure 8.25.

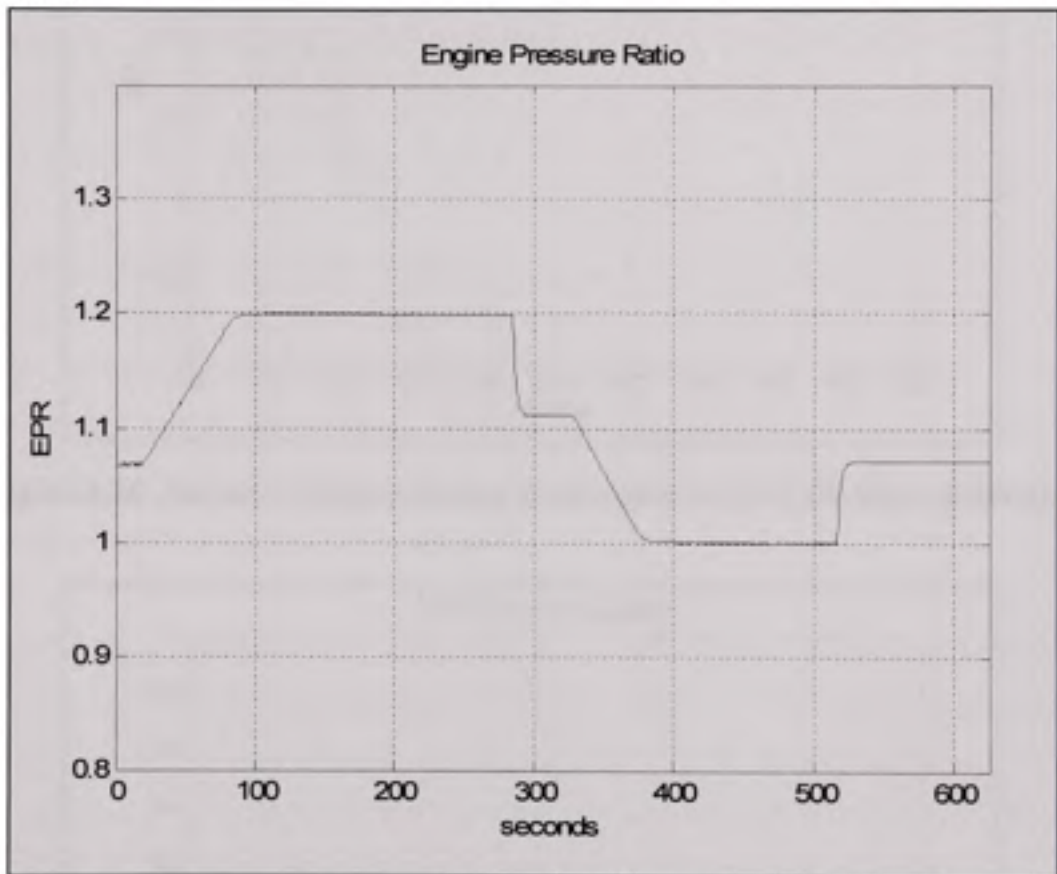


Figure 8.25 EPR variation with EPR limits protection during airspeed capture.

In fact, as EPR reaches the tolerated limits, thrust is maintained to respect these limits and airspeed settles at the corresponding level.

Altitude is maintained during the airspeed capture and the accelerations variation is well within tolerance, as shown in Figure 8.26 and Figure 8.27:

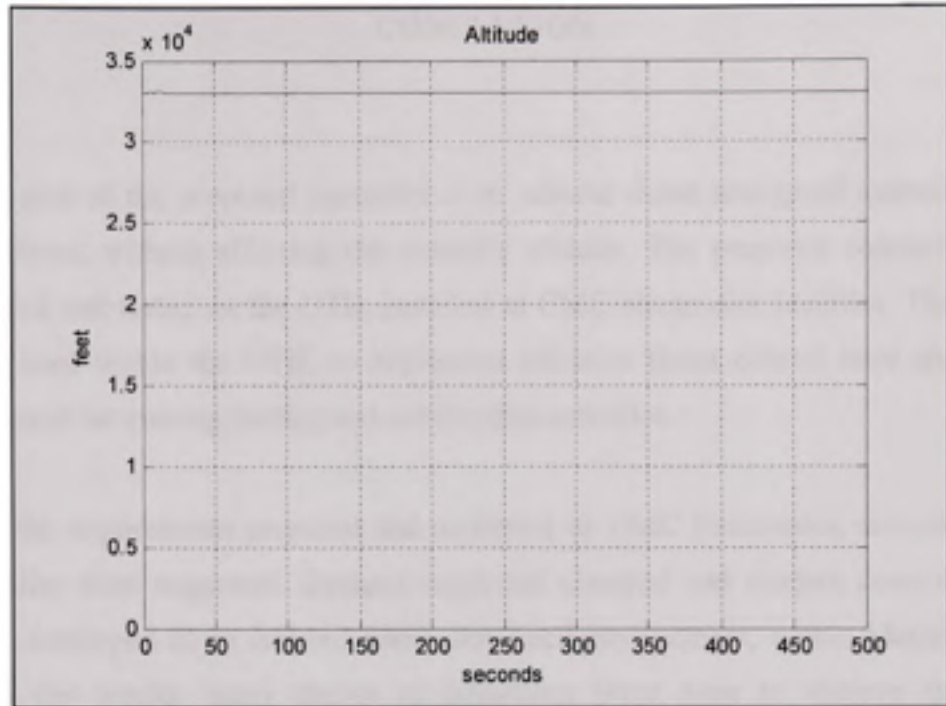


Figure 8.26 Altitude variation during speed capture with EPR limits protection.

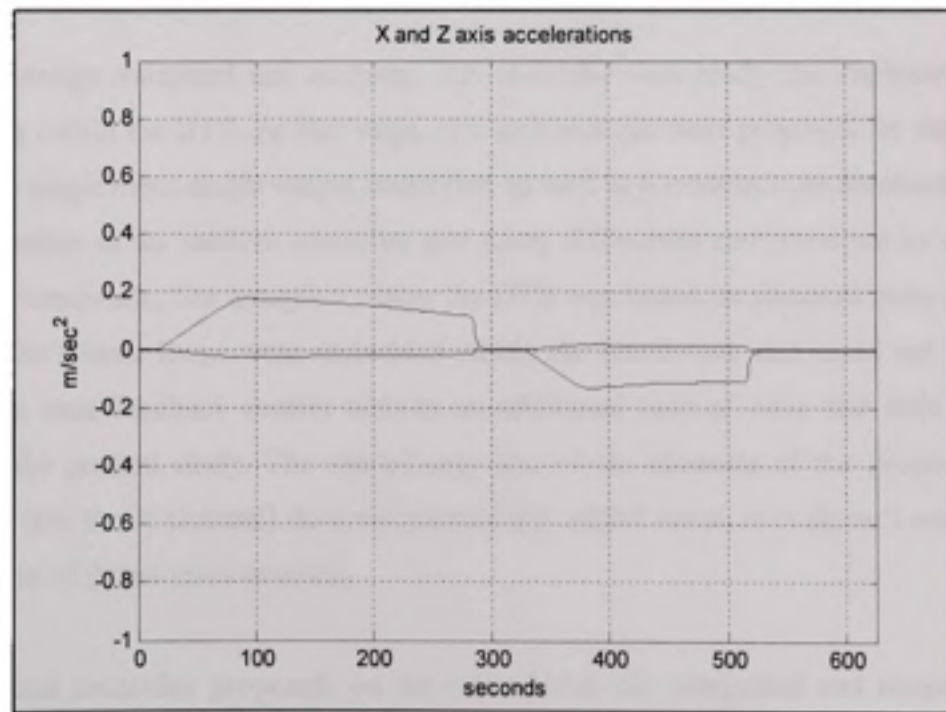


Figure 8.27 Acceleration variation during speed capture with EPR limits protection.

CONCLUSION

The main goal of the proposed controller is to achieve thrust and speed control, using the aircraft's thrust, without affecting the aircraft's altitude. The proposed controller is to be implemented and tested on the DTB, installed at CMC electronics facilities. The controller would be used within the DTB, to implement effective thrust control laws and the DTB would be used for training, testing and certification activities.

Based on the requirements proposed and requested by CMC Electronics, several designs to the controller were suggested. Designs suggested classical and modern control laws, and were first developed in an isolated linear simulated environment, within Matlab/Simulink. Based on the results, many design modifications were done to achieve the different requirements, and were validated and reviewed by CMC, as the customer. The whole design process followed rigorous quality control.

Once the design validated and accepted, the controller was ready for implementation and integration within the DTB. At that stage, two architectures were proposed for the controller: a classical single input-single output controller, as well as a modern state feedback controller. The integration of the modern controller met many difficulties and could not be done within the DTB framework. The autopilot within the DTB was based on classical control laws, and the autopilot's inner loops were embedded within the framework and could not be replaced by modern state feedback control without an additional load of work that falls outside the scope of the present study. The use of only one of the channels of the proposed modern controller (the thrust channel) does not present any added value, as it doesn't meet the main requirement of thrust-pitch coupling.

The classical controller proposed, on the other hand, was integrated and tested within the DTB. Even though the controller is based on classical control laws, the proposed design added many security features, limit protections and enhancements to the classical

Autothrottle. It also added coupling with the pitch control law, to guarantee no altitude variation during thrust variation. All these features, enhancements and protections were integrated to the DTB, tested and validated. Test procedures were written within the scope of this study to demonstrate the controller's ability to meet all the system requirements. The tests were run and validated at CMC, at the end of which the controller was delivered. All requirements were met, and the controller's performance was judged satisfactory.

The thrust control system proposed has proved to be able to capture thrust commands and speed commands, at different points of the envelope, with no overshoot and no steady state error of over 1%. Furthermore, the controller does not cause any altitude variation during thrust variations. It has also shown to respect thrust limits, not overshooting or downshooting beyond these limits at any time. Speed limits are also respected, however we have seen in the results presented in chapter 8 that the controller overshoots and downshoots slightly while respecting these limits during a thrust capture. Improvements at that point can be made and recommendations are presented in the next section. However the results were judged satisfactory, as the aircraft settles within the limitations. The proposed controller also respects the G limits requested in the requirements section.

The proposed Thrust control system meets all defined requirements. Improvements can still be made to minimize energy consumption, integrate a fully coupled pitch-thrust modern controller and improve speed limit protection. These improvements are the subject of the next section.

RECOMMENDATIONS

Even though a modern control concept has been studied for the Thrust controller, a fully integrated modern Thrust controller couldn't be achieved. Integration of the proposed controller within the DTB was problematic, and requests further study and development. The autopilot has been developed within the DTB using classical laws, and any fully modern controller would request modifications to the DTB. An independent autothrottle modern controller without bringing any modification to the Autopilot control laws, especially the inner loops, would not bring any additional benefit, and the value of the added modern controller would be limited. In order to achieve the desired and essential improvements by introducing modern control laws, a fully integrated controller, that would replace both the Autopilot and Autothrottle, needs to be developed. That development would integrate full pitch-thrust coupling, thus minimizing energy consumption during climb and descent, and would eliminate any unwanted pitch or thrust variation due to single channel de-coupled control.

It is important to note that even though it was out of the scope of the present study, an attempt was made to replace one of the Autopilot functionalities by a modern control law, the pitch and altitude hold control laws, to be used with the modern Autothrottle. However, when trying to integrate the modern elevator state feedback control law, coupled with the modern state feedback thrust control law, an additional limitation was apparent. The framework in which the DTB is developed (Flightsim) integrates classical inner loops into the framework, which could not be overridden. Development is done around these Autopilot inner loops, on the outer loops. A modern state feedback controller would need to overpass and replace these inner loops. An attempt was made to try and modify the framework's architecture, but in vain. In order to integrate a modern state feedback controller that would replace both the Autopilot and Autothrottle modifications to the framework need to be done in order to support such integration. That kind of modification falls outside the scope of the present study, even though a modest attempt was made to implement a similar modification. The

modification introduced major instability when switching from classical inner loops to the modern control law. The integration of a modern state feedback loop while keeping the classical controller active was a painful experience, and therefore any integration would request full Autopilot/Autothrottle state feedback control, or a framework which supports multi controller environment.

Despite the fact we couldn't achieve any integrated results for the modern controller, we are confident that the proposed controller would achieve the desired behaviour and requirements within a proper environment, coupled with a modern Autopilot. Any future developments would need to start by the development of a modern state feedback Autopilot, capable of achieving pitch hold, altitude capture and altitude hold. Then, coupling the Autopilot with the Autothrottle state feedback, speed capture, thrust capture and flight level change can be achieved with the coupling between the modern controller's different inputs and channels.

On another note, within the classical controller architecture, several improvements can also be done to achieve better results and to meet the desired requirements. One of the points observed during the results presentation is the controller's overshoot and downshoot when trying to respect a speed limitation during thrust control. In fact, at all times, the controller's output is compared to a parallel controller's output. The first controller tries to achieve the requested thrust, while the second tries to achieve the maximum speed limit. Whenever the command of the first controller is above the command outputted by the second controller, the second output is used as the thrust lever angle command. The opposite happens when protecting the minimum speed limit. This logic achieves the speed protection. However we have seen that the second controller presents overshoot and downshoot at speed limit. When trying to modify the controller's gains to improve the control behaviour, instability was presented when switching from one controller to another. The reason of the instability was caused by the high slope of the response of the second controller that would result in a command above the commanded thrust, thus initiating the speed protection, even tho no speed protection was needed. The time response of the speed protection controller needs to

be slower than the thrust controller, all while achieving acceptable control behaviour. The results we got with the present controller were satisfying to the client; however improvement can be made on the speed protection controller to eliminate any overshoot without affecting the time response too much. Different controller architectures can be studied and developed, and different gains should be tested.

Another aspect of the presented controller that can be improved is the speed capture time. In fact we observed a speed capture time of 0.3 knots per second. No request has been done concerning the speed capture time, and therefore the controller was well within requirements. However it is desirable to achieve a speed capture time of 1 knot per second, as mentioned by the client during the controller's evaluation and test session (CMC electronics). That speed capture time is a reasonable time to achieve speed capture without causing any important acceleration variation while guaranteeing an acceptable capture time for manoeuvres.

Many studies have been done on different controller concepts, architectures and designs. A controller based on Total energy concept can be an interesting alternative. It is however important to study beforehand what would be the added value of any other architecture studied or proposed.

ANNEX I

LINEARISATION RESULTS

```
% ***** FLSIM Linear Model *****  
%  
% Aircraft      : cmc_747_100  
% Initial Conditions: Default  
%  
% ----- Trim Conditions -----  
% Altitude      = 12000 meters  
% Speed (TAS)   = 415 knots  
% Speed (IAS)   = 220 knots  
% Speed (Mach)  = 0.72  
% AoA           = 1.38 deg  
% Throttle      = 73.71 percent  
  
A=[0.262179 3.35594e-006 -0.00307785 -0.00055771 ;...  
5.19622 -0.00495275 0.0285191 -9.83355 ;...  
211.266 -0.106788 -0.569298 -0.224814 ;...  
1 6.57296e-030 6.57296e-030 3.76603e-027 ];  
  
B=[0.0919853 0.0125859 ;...  
0.0345005 2.00458 ;...  
0.847234 -0.087522 ;...  
6.57296e-028 6.57296e-028 ];
```

```

% ***** FLSIM Linear Model *****
%
% Aircraft      : cmc_747_100
% Initial Conditions: Default
%
% ----- Trim Conditions -----
% Altitude      = 12000 meters
% Speed (TAS)   = 449 knots
% Speed (IAS)   = 240 knots
% Speed (Mach)  = 0.78
% AoA           = 0.48 deg
% Throttle      = 74.55 percent

A=[0.29159 -0.000178953 -0.00306647 -0.000189243 ;...
2.0575 -0.00534325 0.0188184 -9.76962 ;...
228.628 -0.104994 -0.627252 -0.083077 ;...
1 3.99533e-031 3.99533e-031 2.28915e-028 ];

B=[0.11051 0.0131734 ;...
0.0116156 2.09815 ;...
1.01825 -0.091607 ;...
3.99533e-029 3.99533e-029 ];

```

```

% ***** FLSIM Linear Model *****
%
% Aircraft      : cmc_747_100
% Initial Conditions: Default
%
% ----- Trim Conditions -----
% Altitude      = 12000 meters
% Speed (TAS)   = 483 knots
% Speed (IAS)   = 260 knots
% Speed (Mach)  = 0.84
% AoA           = -0.25 deg
% Throttle      = 76.34 percent

A=[0.321544 -0.00147816 -0.00391591 0.000355335 ;...
0.985106 -0.0132263 0.0144143 -9.67448 ;...
245.471 -0.163503 -0.715137 0.0289464 ;...
1 0 0 0 ];

B=[0.131327 0.0129354 ;...
0.017664 2.06024 ;...
1.21567 -0.0899521 ;...
0 0 ];

```

```
% ***** FLSIM Linear Model *****  
%  
% Aircraft      : cmc_747_100  
% Initial Conditions: Default  
%  
% ----- Trim Conditions -----  
% Altitude      = 10000 meters  
% Speed (TAS)   = 335 knots  
% Speed (IAS)   = 200 knots  
% Speed (Mach)  = 0.57  
% AoA          = 2.85 deg  
% Throttle     = 67.39 percent  
  
A=[0.260535 7.91873e-005 -0.00328878 -0.000847043 ;...  
8.65105 -0.00487916 0.0441127 -9.88554 ;...  
169.596 -0.11548 -0.579838 -0.464887 ;...  
1 1.05979e-029 1.05979e-029 6.07216e-027 ];  
  
B=[0.0752986 0.0144808 ;...  
0.0549333 2.30639 ;...  
0.69678 -0.100699 ;...  
1.05979e-027 1.05979e-027 ];
```

```

% ***** FLSIM Linear Model *****
%
% Aircraft      : cmc_747_100
% Initial Conditions: Default
%
% ----- Trim Conditions -----
% Altitude      = 10000 meters
% Speed (TAS)   = 382 knots
% Speed (IAS)   = 230 knots
% Speed (Mach)  = 0.66
% AoA           = 1.12 deg
% Throttle      = 67.93 percent

A=[0.297903 -5.93701e-005 -0.00371194 -0.000780559 ;...
3.93143 -0.00527373 0.0366611 -9.87453 ;...
193.611 -0.118292 -0.685267 -0.182349 ;...
1 0 0 0];

B=[0.0986665 0.0154103 ;...
0.0345045 2.45442 ;...
0.914192 -0.107162 ;...
0 0];

```



```

% ***** FLSIM Linear Model *****
%
% Aircraft      : cmc_747_100
% Initial Conditions: Default
%
% ----- Trim Conditions -----
% Altitude      = 10000 meters
% Speed (TAS)   = 412 knots
% Speed (IAS)   = 250 knots
% Speed (Mach)  = 0.71
% AoA           = 0.31 deg
% Throttle      = 68.77 percent

A=[0.328055 -0.000115229 -0.00381864 -9.12847e-005 ;...
1.24698 -0.00689147 0.021157 -9.75864 ;...
209.224 -0.118324 -0.756374 -0.0558982 ;...
1 -3.15544e-030 -3.15544e-030 -1.80794e-027 ];

B=[0.116826 0.0163995 ;...
0.00296505 2.61198;...
1.07373 -0.114041 ;...
3.15544e-028 -3.15544e-028 ];

```

```

% ***** FLSIM Linear Model *****
%
% Aircraft      : cmc_747_100
% Initial Conditions: Default
%
% ----- Trim Conditions -----
% Altitude      = 8000 meters
% Speed (TAS)   = 271 knots
% Speed (IAS)   = 180 knots
% Speed (Mach)  = 0.45
% AoA           = 4.64 deg
% Throttle      = 62.06 percent

A=[0.264046 0.000227709 -0.00345747 -0.000378859 ;...
11.333 -0.0047308 0.0610227 -9.80448 ;...
136.382 -0.112382 -0.579413 -0.76474 ;...
1 0 0 0];

B=[0.245974 0.0158361 ;...
0.197129 2.52225 ;...
2.27093 -0.110124 ;...
0 0];

```

```

% ***** FLSIM Linear Model *****
%
% Aircraft      : cmc_747_100
% Initial Conditions: Default
%
% ----- Trim Conditions -----
% Altitude      = 8000 meters
% Speed (TAS)   = 314 knots
% Speed (IAS)   = 210 knots
% Speed (Mach)  = 0.52
% AoA           = 2.43 deg
% Throttle      = 62.09 percent

A=[0.302691 7.73412e-005 -0.00377749 -0.000479986 ;...
6.87429 -0.00603746 0.0614738 -9.83554 ;...
158.685 -0.124237 -0.684872 -0.397339 ;...
1 0 0 0];

B=[0.0823484 0.0159314 ;...
0.0461256 2.53743 ;...
0.761076 -0.110787 ;...
0 0];

```

```
% ***** FLSIM Linear Model *****  
%  
% Aircraft      : cmc_747_100  
% Initial Conditions: Default  
%  
% ----- Trim Conditions -----  
% Altitude      = 8000 meters  
% Speed (TAS)   = 371 knots  
% Speed (IAS)   = 250 knots  
% Speed (Mach)  = 0.62  
% AoA           = 0.54 deg  
% Throttle      = 64.88 percent  
  
A=[0.354166 -7.86528e-005 -0.00439394 -0.000290875 ;...  
1.92233 -0.00722518 0.0270799 -9.79903 ;...  
187.591 -0.125582 -0.841302 -0.0918029 ;...  
1 6.31089e-030 6.31089e-030 3.61587e-027 ];  
  
B=[0.11481 0.0176367 ;...  
0.0164363 2.80904 ;...  
1.06225 -0.122645 ;...  
6.31089e-028 6.31089e-028 ];
```

```

% ***** FLSIM Linear Model *****
%
% Aircraft      : cmc_747_100
% Initial Conditions: Default
%
% ----- Trim Conditions -----
% Altitude      = 6000 meters
% Speed (TAS)   = 243 knots
% Speed (IAS)   = 180 knots
% Speed (Mach)  = 0.39
% AoA           = 4.72 deg
% Throttle      = 55.32 percent

A=[0.294493 0.000273994 -0.00395644 0.000185063 ;...
10.3285 -0.00588535 0.0722661 -9.72033 ;...
121.921 -0.123078 -0.652621 -0.781592 ;...
1 2.77094e-030 2.77094e-030 1.58763e-027 ];

B=[0.246452 0.0165399 ;...
0.193099 2.63435 ;...
2.28811 -0.115018 ;...
2.77094e-028 2.77094e-028 ];

```

```

% ***** FLSIM Linear Model *****
%
% Aircraft      : cmc_747_100
% Initial Conditions: Default
%
% ----- Trim Conditions -----
% Altitude      = 6000 meters
% Speed (TAS)   = 308 knots
% Speed (IAS)   = 230 knots
% Speed (Mach)  = 0.50
% AoA           = 1.56 deg
% Throttle      = 58.43 percent

A=[0.364192 4.97073e-005 -0.00480233 2.62753e-005 ;...
4.38375 -0.00828477 0.0536837 -9.76102 ;...
155.411 -0.133566 -0.847254 -0.258172 ;...
1 0 0 0];

B=[0.0981075 0.0179827 ;...
0.0308598 2.86414 ;...
0.908744 -0.125051 ;...
0 0];

```

```

% ***** FLSIM Linear Model *****
%
% Aircraft      : cmc_747_100
% Initial Conditions: Default
%
% ----- Trim Conditions -----
% Altitude      = 6000 meters
% Speed (TAS)   = 360 knots
% Speed (IAS)   = 270 knots
% Speed (Mach)  = 0.59
% AoA           = 0.10 deg
% Throttle      = 62.87 percent

A=[0.417206 -0.000155131 -0.00509621 0.000161376 ;...
0.43087 -0.00948142 0.0302727 -9.72856 ;...
181.549 -0.133068 -1.02552 -0.0244124 ;...
1 0 0 0];

B=[0.132409 0.0204092 ;...
0.00111692 3.25061 ;...
1.22238 -0.141925 ;...
0 0];

```

```

% ***** FLSIM Linear Model *****
%
% Aircraft      : cmc_747_100
% Initial Conditions: Default
%
% ----- Trim Conditions -----
% Altitude      = 4000 meters
% Speed (TAS)   = 219 knots
% Speed (IAS)   = 180 knots
% Speed (Mach)  = 0.35
% AoA           = 4.80 deg
% Throttle      = 44.82 percent

A=[0.32522 0.000349073 -0.0044988 -0.00158794 ;...
9.50814 -0.00481491 0.0838567 -10.0083 ;...
109.348 -0.133333 -0.72979 -0.782902 ;...
1 1.24989e-030 1.24989e-030 7.16133e-028 ];

B=[0.245822 0.00798459 ;...
0.244217 1.27172 ;...
2.28492 -0.0555245 ;...
1.24989e-028 1.24989e-028 ];

```



```

% ***** FLSIM Linear Model *****
%
% Aircraft      : cmc_747_100
% Initial Conditions: Default
%
% ----- Trim Conditions -----
% Altitude      = 4000 meters
% Speed (TAS)   = 279 knots
% Speed (IAS)   = 230 knots
% Speed (Mach)  = 0.44
% AoA           = 1.66 deg
% Throttle      = 53.16 percent

A=[0.401353 1.82585e-005 -0.00536301 0.000222057 ;...
4.22357 -0.0101635 0.0586916 -9.73642 ;...
139.873 -0.133178 -0.951125 -0.275521 ;...
1 1.26218e-029 1.26218e-029 7.23174e-027 ];

B=[0.0979098 0.0195231 ;...
0.028323 3.10948 ;...
0.907948 -0.135763 ;...
1.26218e-027 1.26218e-027 ];

```

```

% ***** FLSIM Linear Model *****
%
% Aircraft      : cmc_747_100
% Initial Conditions: Default
%
% ----- Trim Conditions -----
% Altitude      = 4000 meters
% Speed (TAS)   = 326 knots
% Speed (IAS)   = 270 knots
% Speed (Mach)  = 0.52
% AoA           = 0.31 deg
% Throttle      = 58.29 percent

A=[0.458882 -0.000102798 -0.00575003 0.000263279 ;...
1.0222 -0.0113674 0.0352458 -9.7212 ;...
163.85 -0.141274 -1.12487 -0.0580706 ;...
1 -1.57772e-030 -1.57772e-030 -9.03968e-028 ];

B=[0.132073 0.0211612 ;...
0.00559691 3.37038 ;...
1.22144 -0.147154 ;...
1.57772e-028 -1.57772e-028 ];

```

```

% ***** FLSIM Linear Model *****
%
% Aircraft      : cmc_747_100
% Initial Conditions: Default
%
% ----- Trim Conditions -----
% Altitude      = 2000 meters
% Speed (TAS)   = 198 knots
% Speed (IAS)   = 180 knots
% Speed (Mach)  = 0.31
% AoA           = 4.87 deg
% Throttle      = 33.51 percent

A=[0.358272 0.000410093 -0.00504353 -0.00144686 ;...
8.72355 -0.00566165 0.0968154 -9.99162 ;...
98.3566 -0.144644 -0.811139 -0.795994 ;...
1 1.2549e-029 1.2549e-029 7.19003e-027 ];

B=[0.247818 0.0103835 ;...
0.243758 1.6538 ;...
2.29459 -0.0722065 ;...
1.2549e-027 1.2549e-027 ];

```

```

% ***** FLSIM Linear Model *****
%
% Aircraft      : cmc_747_100
% Initial Conditions: Default
%
% ----- Trim Conditions -----
% Altitude      = 2000 meters
% Speed (TAS)   = 253 knots
% Speed (IAS)   = 230 knots
% Speed (Mach)  = 0.39
% AoA           = 1.76 deg
% Throttle      = 34.42 percent

A=[0.440865 7.92893e-005 -0.00592773 -0.00159275 ;...
4.07399 -0.00829118 0.0643293 -10.0323 ;...
126.146 -0.138447 -1.05764 -0.279605 ;...
1 -6.31089e-030 -6.31089e-030 -3.61587e-027 ];

B=[0.0983806 0.0101515 ;...
0.0780419 1.61685 ;...
0.917444 -0.070593 ;...
6.31089e-028 -6.31089e-028 ];

```

```

% ***** FLSIM Linear Model *****
%
% Aircraft      : cmc_747_100
% Initial Conditions: Default
%
% ----- Trim Conditions -----
% Altitude      = 2000 meters
% Speed (TAS)   = 296 knots
% Speed (IAS)   = 270 knots
% Speed (Mach)  = 0.46
% AoA           = 0.43 deg
% Throttle      = 53.48 percent

A=[-0.503764 -0.00013255 -0.00647753 0.000244543 ;...
1.2615 -0.0131092 0.0417751 -9.73185 ;...
148.12 -0.135112 -1.25392 -0.0768179 ;...
1 -1.57772e-030 -1.57772e-030 -9.03968e-028 ];

B=[0.133259 0.023797 ;...
0.00948723 3.7902 ;...
1.2351 -0.165483 ;...
1.57772e-028 -1.57772e-028 ];

```

ANNEX II

SOFTWARE/SYSTEM VALIDATION & TEST PROCEDURES

2.1 Introduction

This document will define the software and system validation procedure for the thrust control system, which is part of the Dynamic Test Bed (DTB) for the Flight Management System (FMS).

2.2 Scope

The validation of the TC model shall be performed using the procedures outlined in this document.

2.3 Reference Documents

The following is the list of reference documents.

[D1] Dynamic Test Bed Training Project Course Notes

[D2] FLSIM version 9.1 Reference Guide

[D3] Autothrottle Literature Review

[D4] FLSIM version 9.1

[D5] CMA-900 FMS Part Task Trainer v. Engineering, Part Number: 169-614378-014

[D6.1] CMC GUI (AFCS_CCG and COCKPIT_CCG), P/N: 169-615002-001

[D6.2] ETS TC GUI.

[D7] Software Operational Requirements Document for the Thrust Control System Revision 1

[D8] Software Design Document for the Thrust Control System Revision 3

2.4 Test Preparation

2.4.1 Equipment List

[TE1] 1 PC with 2 Monitors named DTB1

[TE2] Diskette – Microsoft Windows XP Service Pack 2

[TE3] Diskette – FLSIM 9.1 software

[TE4] Diskette – CMA -900 FMS Part Task Trainer Modifications v. Engineering, Part Number: 169-614378-014

[TE5] Diskette – DTB simulation extension dll software

[TE6] Diskette – ets_um_tc.dll, dtb_global.h and tc_user_interface.exe

[TE7] Diskette – Snapshot/Restore Files

2.4.2 System Setup

Software Installation and Configuration

Install Item 2 to Item 7 from the equipment list into Item 1, and set-up all required system parameters to be able to run the FLSIM [D4] and the PTT [D5] properly. The system configurations are listed in Table 2.1.

Table 2.1
System Variables

System Variables	Value
FLSIM_BASE_FEATURE	200
FLSIM_DIR	C:\eti\flsim91
LM_LICENSE_FILE	@diamonds.ele.etsmtl.ca
PTT_DIR	C:\ptt
DTB_DIR	C:\DTB

2.4.3 User Model Installation

Install the user model to be tested. Copy the files ets_um_tc.dll [TE6] into folder: C:\DTB\working_simulation_extension\CMC_flsim9_dll.

2.4.4 DTB Global Header File Installation

Install the header file used to specify all message handling and data structure used for the ETS TC GUI [D6.2] to communicate with the FLSIM [D4] environment. Copy the file dtb_global.h [TE6] into the folder:

C:\DTB\working_simulation_extension\Common.

2.4.5 ETS Temporary GUI Installation

Install the temporary GUI [D6.2] to be used to aid in testing of the Thrust Controller model. Copy the file tc_user_interface.exe [TE6] into the folder: C:\eti\flsim91\bin.

2.4.6 Snapshot / Restore File Installation

Install the files required to recapture the testing conditions being preformed. Copy the contents of [TE7] into the folder: C:\eti\flsim91\data\rap.

2.5 Software Initialisation

2.5.1 Initialization of PTT

This section will initialize and validate the PTT [D5] software. This will help choose a flight plan and engage the autopilot vertical navigation for use with FMS Guidance mode tests.

Setup:

Edit the C:\ptt\code\ptt\lib\setup_flsim9.bat file as following:

```
@echo off
SET FLSIM_MMI_RECV_PORT=8000
SET FLSIM_SIM_RECV_PORT=8001
SET FLSIM_SCENE_RECV_PORT=8002
SET FLSIM_RE_RECV_PORT=8003
SET FLSIM_EXPORT_KEY=0x3010
SET FLSIM_SIM_MACHINE_ADDRESS=
SET CMC_FLSIM_SIM_ADDRESS=127.0.0.1
SET CMC_PTT_ADDRESS=127.0.0.1
fmsptt.exe pttconfig-DTB_soft.xml
```

Configuration:

Configure the PTT as presented in table 2.2:

Table 2.2
PTT configuration steps

Step #	Actions	Expected Result(s)	Check Box
1	Execute c:\ptt\code\ptt\lib\setup_fsim9.bat.	PTT program will start, message box "Library Load" will be displayed.	
2	On the MCDU MENU 1/2 page, select FMS1 <REQUESTING>	Enter the FMS1 1/1 IDENT page.	
3	On the FMS1 IDENT 1/1 page, clear messages if any present in message area by selecting CLR from the keypad	Messages in yellow cleared	
4	On the FMS1 IDENT 1/1 page, select from keypad INIT/REF.	Enter the FMS1 INIT/REF INDEX 1/2 page.	
5	On the FMS1 INIT/REF INDEX 1/2 page, select POS INIT.	Enter the FMS1 POS INIT 1/2 page.	
6	On the POS INIT 1/2 page, type "CYUL" in the message area using the Keypad, then select REF AIRPORT.	The REF AIRPORT is now CYUL.	
7	On the POS INIT 1/2 page, select ROUTE	Enter the FMS1 ROUTE 1/1 page.	
8	On the ROUTE 1/1 page, type "CYQB" in the message area using the Keypad, then select DEST.	The DEST airport is now CYQB while the ORIGIN airport is CYUL.	
9	On the RTE 1/1 page, select EXEC from the Keypad.	Enter the FMS1 ROUTE 1/1 page.	

Table 2.3
PTT configuration steps (continued)

Step #	Actions	Expected Result(s)	Check Box
10	Select from keypad DEP/ARR	Enter the FMS1 DEP/ARR INDEX 1/1 page.	
11	On the CYQB ARRIVALS page, select APPROACHES ILS06	The term <SEL> is mentioned next to APPROACHES ILS06	
12	On the CYQB ARRIVALS page, select EXEC from the Keypad.	The term <ACT> is mentioned next to APPROACHES ILS06	
13	On the CYQB ARRIVALS page, select TRANS YQB	The term <SEL> is mentioned next to TRANS YQB	
14	On the CYQB ARRIVALS page, select EXEC from the Keypad	The term <ACT> is mentioned next to TRANS YQB	
15	Select from keypad DEP/ARR	Enter the FMS1 DEP/ARR INDEX 1/1 page.	
16	On the FMS1 DEP/ARR INDEX 1/1 page, select DEP.	Enter the CYUL DEPARTURES page.	
17	On the CYUL DEPARTURES page, select RUNWAY 10	The term <SEL> is mentioned next to RUNWAY 10	
18	On the CYUL DEPARTURES page, select EXEC from the Keypad.	The term <ACT> is mentioned next to RUNWAY 10	
19	Select PROG from the Keypad to monitor progress	Enter the FMS1 ACT PROGRESS 1/4 page.	
20	On the PTT left side, select "HDG SEL" button	the green "HDG SEL" button turns into a yellow "AUTO ON" button to indicate LNAV is ON	
21	On the PTT left side, select "ALT SEL" button	the green "ALT SEL" button turns into a yellow "VS SEL" button to indicate VNAV is ON	

2.5.2 Initialisation of FLSIM

The FLSIM [D4] is the main software program used in this test. The user models that were created will be added to the FLSIM [D4] as an extension. The functional requirements of the TC model are to be tested under the FLSIM [D4] platform. This section will initialize and validate the FLSIM [D4] software. It will also initialise the plots to be used during tests procedures. NOTE: The following set-up is that used at ETS.

Setup:

Edit the C:\eti\flsim9\setup_flsim9.bat file as following:

```
@echo off
SET FLSIM_DIR=C:\eti\Flsim91
SET XENVIRONMENT=C:\eti\flsim91\X_Flsim_Resources
SET FLSIM_MMI_RECV_PORT=8000
SET FLSIM_SIM_RECV_PORT=8001
SET FLSIM_SCENE_RECV_PORT=8002
SET FLSIM_RE_RECV_PORT=8003
SET FLSIM_EXPORT_KEY=0x3010
SET FLSIM_ATLAS_COLOR_FILE=C:\eti\flsim91\data\Atlas_Colors_NEW
SET FLSIM_SIM_MACHINE_ADDRESS=
SET FLSIM_SHARED_LIB_DIR=c:\dtb\working_simulation_extension\CMC_flsim9_dll
SET FLSIM_BASE_FEATURE=200
SET CMC_PTT_ADDRESS=127.0.0.1
SET CMC_FLSIM_SIM_ADDRESS=127.0.0.1
cd bin
start flsimde.exe -g CMC_747_100 -s FlsimSceneUser.exe
start cmd.exe /K flsimsim.exe
pause
```

```

start AFCS_CCG.exe -geometry 512x300+0+0 -norefresh -WinGdi
pause
start cockpit_ccg.exe -geometry 512x300+513+0 -norefresh -WinGdi
pause
start tc_user_interface.exe

```

Configuration:

Configure FLSIM as presented in Table 2.3:

Table 2.3
FLSIM configuration steps

Step#	Actions	Expected Result(s)	Check Box
1	Execute c:\eti\flsim91\setup_flsim9.bat	The FLSIM [D4] program will launch including FlsimDE, flsimsim, the 2 VAPS user control panels [D6.1] and the TC User Interface [D6.2]	
2	Select the window labeled setup_flsim9.bat, press the space bar three times.	The three user panels will now be activated: On the left, afcs_ccg panel [D6.1], On the right, cockpit_ccg panel [D6.1], In the middle, the TC User Interface panel [6.2].	
3	On the FlsimDE menu, Aircraft -- - Identification, replace the default Autothrottle model with the user model: ETS_TC	The user model will be loaded into the FLSIM [D4].	
	On the FlsimDE menu, Aircraft -- - Identification, select SAVE and save Aircraft Profile as ETS_747_100	The Aircraft profile loaded into the FLSIM is now ETS_747_100	
4	On the Identification menu, select Close.	Identification menu window will be closed.	

Table 2.3
FLSIM configuration steps (continued)

Step#	Actions	Expected Result(s)	Check Box
5	On the FlsimDe menu, select Utilities --- Plots.	Plots menu window will be displayed.	
6	On the Plots menu, select Enable Plots.	Plots are now activated in the FLSIM [D4].	
7	On the Plots Menu - Tab 0, Select Variables.	The Parameter Selection menu will be displayed.	
8	On the Parameter Selection menu, choose Tab Eng 1, select variable "EPR_1" (Engine Pressure Ratio).	"EPR" will be displayed in the Select Variable window.	
9	On the Plots Menu - Tab 0, enter Axis Name as "EPR".	"EPR" will be displayed in the Axis Name window.	
10	On the Plots Menu - Tab 0, enter Maximum as "3".	"3" will be displayed in the Maximum window.	
11	On the Plots menu - Tab 0, select Show Plot.	The "EPR" plot will be displayed.	
12	On the Plots Menu, select Tab 1.	The Tab 1 window will be displayed.	
13	On the Plots Menu - Tab 1, Select Variables.	The Parameter Selection menu will be displayed.	
14	On the Parameter Selection menu, choose Tab std, select variable "Vi" (Indicated Airspeed).	"IAS" will be displayed in the Select Variable window.	
15	On the Plots menu - Tab 1, select Unit as "Speed (Knots)".	"Speed" will be displayed in the Units window.	
16	On the Plots Menu - Tab 1, enter Axis Name as "IAS".	"IAS" will be displayed in the Axis Name window.	
17	On the Plots Menu - Tab 1, enter Maximum as "600".	"600" will be displayed in the Maximum window.	
18	On the Plots menu - Tab 1, select Show Plot.	The "IAS" plot will be displayed.	

Table 2.3
FLSIM configuration steps (continued)

Step#	Actions	Expected Result(s)	Check Box
19	On the Plots Menu, select Tab 2.	The Tab 2 window will be displayed.	
20	On the Plots Menu - Tab 2, Select Variables.	The Parameter Selection menu will be displayed.	
21	On the Parameter Selection menu, choose Tab std, select variable "Mn"	"Mn" will be displayed in the Select Variable window.	
22	On the Plots Menu - Tab 2, enter Axis Name as "Mach".	"Mach" will be displayed in the Axis Name window.	
23	On the Plots Menu - Tab 2, enter Maximum as "1.2".	"1.2" will be displayed in the Maximum window.	
24	On the Plots menu - Tab 2, select Show Plot.	The "Mach" plot will be displayed.	
25	On the Plots Menu, select Tab 3.	The Tab 3 window will be displayed.	
26	On the Plots Menu - Tab 3, Select Variables.	The Parameter Selection menu will be displayed.	
27	On the Parameter Selection menu, choose Tab std, select variable "h rad" (Radar Altitude).	"h rad" will be displayed in the Select Variable window.	
28	On the Plots menu - Tab 3, select Unit as "Height (m)".	"Height" will be displayed in the Units window.	
29	On the Plots Menu - Tab 3, enter Axis Name as "Radar ALT".	"Radar ALT" will be displayed in the Axis Name window.	
30	On the Plots Menu - Tab 3, enter Maximum as "12000".	"12000" will be displayed in the Maximum window.	
31	On the Plots menu - Tab 3, select Show Plot.	The "Radar ALT" plot will be displayed.	
32	On the Plots Menu, select Tab 4.	The Tab 4 window will be displayed.	

Table 2.3
FLSIM configuration steps (continued)

Step#	Actions	Expected Result(s)	Check Box
33	On the Plots Menu - Tab 4, Select Variables.	The Parameter Selection menu will be displayed.	
34	On the Parameter Selection menu, choose Tab std, select variable "V_dot_x".	"V_dot_x" will be displayed in the Select Variable window.	
35	On the Plots menu – Tab4, select Unit as "Acceleration (m/s2)".	"Acceleration" will be displayed in the Units window.	
36	On the Plots Menu - Tab 4, enter Axis Name as "Forward acceleration".	"Forward acceleration" will be displayed in the Axis Name window.	
37	On the Plots Menu - Tab 4, enter Maximum as "10".	"10" will be displayed in the Maximum window.	
38	On the Plots menu - Tab 4, select Show Plot.	The "Forward acceleration" plot will be displayed.	
39	On the Plots Menu, select Tab 5.	The Tab 5 window will be displayed.	
40	On the Plots Menu - Tab 5, Select Variables.	The Parameter Selection menu will be displayed.	
41	On the Parameter Selection menu, choose Tab std, select variable "V_dot_z".	"V_dot_z" will be displayed in the Select Variable window.	
42	On the Plots menu – Tab5, select Unit as "Acceleration (m/s2)".	"Acceleration" will be displayed in the Units window.	
43	On the Plots Menu - Tab 5, enter Axis Name as "Climb rate".	"Climb Rate" will be displayed in the Axis Name window.	
44	On the Plots Menu - Tab 5, enter Maximum as "10".	"10" will be displayed in the Maximum window.	
45	On the Plots menu - Tab 5, select Show Plot.	The "Climb rate" plot will be displayed.	
46	On the Plots Menu, select Tab 6.	The Tab 6 window will be displayed.	

Table 2.3
FLSIM configuration steps (continued)

Step#	Actions	Expected Result(s)	Check Box
47	On the Plots Menu - Tab 6, Select Variables.	The Parameter Selection menu will be displayed.	
48	On the Parameter Selection menu, choose Tab std, select variable "pil_eng_thr0".	"pil_eng_thr0" will be displayed in the Select Variable window.	
49	On the Plots menu – Tab6, select Unit as None.	None will be displayed in the Units window.	
50	On the Plots Menu - Tab 6, enter Axis Name as "Throttle".	"Throttle" will be displayed in the Axis Name window.	
51	On the Plots menu - Tab 6, select Show Plot.	The "Throttle" plot will be displayed.	
52	On the Plots menu, select SAVE and save plot profile under the file name ETS_plots_profile	The plots profile loaded into the FLSIM is now ETS_plots_profile	
53	On the FlsimDE menu, Aircraft -- Identification, select SAVE and save new Aircraft Profile including plots as ETS_747_100_plots	The Aircraft profile loaded into the FLSIM is now ETS_747_100_plots	

2.6 TC model validation procedures

TC Model Functionality Test Procedures

This section will verify that all the functionalities of the TC model have been met according to the Software Requirements Document for the TC Model [D7].

The autothrottle has requirements that were set forth in the Software Requirements Document for the TC Model [D7]. These requirements include EPR capture and EPR limits protection, speed capture with speed and acceleration limits protection, as well as pitch coupling.

In order to assure that the testing is performed properly and all requirements are met, different test sections will be set up.

2.6.1 Thrust Idle

This test case will perform requirement check of the Thrust idle Mode's ability to decrease thrust until idle position is attained.

Configure FLSIM as presented in Table 2.4:

Table 2.4
Thrust Idle test simulation startup steps

Step #	Actions	Expected Results	Check Box
1	On the FlsimDe menu, Control Execution, click RAP tab.	Record and Playback menu will be displayed.	
2	On RAP tab, select Playback, click on Filename, choose file 'Thrust_idle', click on Snapshot/Restore, click on Start Time and enter 00:09:30.	The simulation is set to resume flight at altitude of 30000 feet and speed of 400 Knots	
3	On the FlsimDe menu, Control Execution, click Plot Duration.	The Plot Duration window will be displayed.	
4	On the Plot Duration window, enter 300.	Plotting duration is set to 300 seconds.	
5	On the FlsimDe menu, Control Execution, click Run.	The simulation starts and A/C is flying at altitude of 30000 feet and speed of 400 Knots	
6	Press escape.	Exit simulation to FlsimDe.	
7	On the FlsimDe menu, Control Execution, click Resume no RAP.	The simulation starts and A/C is now held at altitude of 30000 feet and speed of 400 Knots	

Run test procedures presented in Table 2.5:

Table 2.5
Thrust Idle test procedures

Step #	Actions	Expected Result(s)	Check Box
1	In the Maximum allowed EPR box of the ETS TC Temporary GUI [D6.2], enter an EPR of 1.8. In the Minimum allowed EPR box, enter a value of 1.1. Click on Activate.	On the ETS Temporary GUI [D6.2] message section, the following message is announced: "New EPR limits set"	
2	On the ETS TC Temporary GUI [D6.2], check Thrust idle. Click on Activate. Select and activate ATS mode on CMC Temporary GUI [D6.1].	On the ETS Temporary GUI [D6.2] message section, the following message is announced: "Thrust idle Mode Armed".	
3	Observe A/C thrust and altitude variation on OTW (Out-The-Window) scene and display.	Throttles slowly decrease till they reach required idle position. Observe A/C descent and altitude variation on the OTW display.	
4	Select Plot 6, "Throttle".	The Capture Trajectory is displayed. A/C throttles position will eventually reach and maintain the reference idle value of 5% (pilot selected idle of 0)	
5	Select Clear on ETS TC temporary GUI [D6.2].	All messages will be cleared for ETS TC temporary GUI [D6.2].	
6	Press the escape key on the keyboard.	Simulation will be terminated.	
7	On the Plots menu, select Clear All.	All plots will be cleared.	

2.6.2 EPR Arm and Reference Limit Protection

This test case will perform requirement check of the EPR Arm Mode's ability to respect the EPR limits for EPR reference inputs and the functioning of EPR Arm mode logic.

Configure FLSIM as presented in Table 2.6:

Table 2.6
EPR Arm and Reference Limit test simulation startup steps

Step #	Actions	Expected Results	Check Box
1	On the FlsimDe menu, Control Execution, click RAP tab.	Record and Playback menu will be displayed.	
2	On RAP tab, select Playback, click on Filename, choose file 'epr_arm', click on Snapshot/Restore, click on Start Time and enter 00:09:30.	The simulation is set to resume flight at altitude of 30000 feet and speed of 400 knots.	
3	On the FlsimDe menu, Control Execution, click Plot Duration.	The Plot Duration window will be displayed.	
4	On the Plot Duration window, enter 300.	Plotting duration is set to 300 seconds.	
5	On the FlsimDe menu, Control Execution, click Run.	The simulation starts and A/C is flying at altitude of 30000 feet and speed of 400 knots.	
6	Press escape.	Exit simulation to FlsimDe.	
7	On the FlsimDe menu, Control Execution, click Resume no RAP.	The simulation starts and A/C is now held at altitude of 30000 feet and speed of 400 knots.	

Run test procedures presented in Table 2.7:

Table 2.7
EPR Arm and Reference Limit test procedures

Step #	Actions	Expected Result(s)	Check Box
1	In the Maximum allowed EPR box of the ETS TC Temporary GUI [D6.2], enter an EPR of 1.8. In the Minimum allowed EPR box, enter a value of 1.1.	On the ETS Temporary GUI [D6.2] message section, the following message is announced: "New EPR limits set"	
2	On the ETS TC Temporary GUI [D6.2], check EPR control and enter an EPR Reference value of 3. Click on Activate.	On the ETS Temporary GUI [D6.2] message section, the following message is announced: "Selected EPR not within acceptable limits, EPR Reference value adjusted to maximum EPR".	
3	On the ETS TC Temporary GUI [D6.2], keep EPR control checked and enter an EPR Reference value of 0.5. Click on Activate.	On the ETS Temporary GUI [D6.2] message section, the following message is announced: "Selected EPR not within acceptable limits, EPR Reference value adjusted to minimum EPR".	
4	On the ETS TC Temporary GUI [D6.2], check EPR control and enter an EPR Reference value of 1.6. Click on Activate.	On the ETS Temporary GUI [D6.2] message section, the following message is announced: "EPR Capture Mode Armed".	
	Select and activate ATS mode on CMC Temporary GUI [D6.1].	ATS button light turns on.	
5	Select Clear on ETS TC temporary GUI [D6.2].	All messages will be cleared for ETS TC temporary GUI [D6.2].	
6	Press the escape key on the keyboard.	Simulation will be terminated.	
7	On the Plots menu, select Clear All.	All plots will be cleared.	

2.6.3 EPR Capture Flight Level Change Climb

This test case will perform requirement check of the EPR Capture Mode's ability to increase thrust and capture a desired predefined climb EPR. The increase of thrust will cause climb and level change and will be done at constant speed using IAS pitch control to maintain speed at desired level.

Configure FLSIM as presented in Table 2.8:

Table 2.8
EPR Climb capture test simulation startup steps

Step #	Actions	Expected Results	Check Box
1	On the FlsimDe menu, Control Execution, click RAP tab.	Record and Playback menu will be displayed.	
2	On RAP tab, select Playback, click on Filename, choose file 'epr_climb', click on Snapshot/Restore, click on Start Time and enter 00:05:30.	The simulation is set to resume flight at altitude of 20000 feet and speed of 300 knots.	
3	On the FlsimDe menu, Control Execution, click Plot Duration.	The Plot Duration window will be displayed.	
4	On the Plot Duration window, enter 300.	Plotting duration is set to 300 seconds.	
5	On the FlsimDe menu, Control Execution, click Run.	The simulation starts and A/C is flying at altitude of 20000 feet and speed of 300 knots.	
6	Press escape.	Exit simulation to FlsimDe.	
7	On the FlsimDe menu, Control Execution, click Resume no RAP.	The simulation starts and A/C is now held at altitude of 20000 feet and speed of 300 knots.	

Run test procedures presented in Table 2.9:

Table 2.9
EPR Climb capture test procedures

Step #	Actions	Expected Result(s)	Check Box
1	In the Maximum allowed EPR box of the ETS TC Temporary GUI [D6.2], enter an EPR of 1.8. In the Minimum allowed EPR box, enter a value of 1. Click on Activate.	On the ETS Temporary GUI [D6.2] message section, the following message is announced: "New EPR limits accepted".	
2	Select an Altitude Reference of 25000 feet in the input box of ALT on afcs_ccg panel of CMC Temporary GUI [D6.1] which corresponds to 180 m/s	On afcs_ccg panel of CMC Temporary GUI [D6.1], "25000" is displayed in ALT window.	
3	Select a Speed Reference of 350 knots in the input box of Speed on afcs_ccg panel of CMC Temporary GUI [D6.1] which corresponds to 180 m/s	On afcs_ccg panel of CMC Temporary GUI [D6.1], "350" is displayed in Speed window.	
4	Activate IAS MODE on afcs_ccg panel of CMC Temporary GUI [D6.1] by clicking on the IAS button.	IAS button light turns on.	
5	Select plot 1, "IAS" Plot.	A/C IAS Speed reaches and maintains reference value of 180 m/s	
6	Select and activate ATS mode on CMC Temporary GUI [D6.1].	ATS button light turns on.	
7	Observe A/C climb and altitude variation on the OTW (out the window) scene and display.	Throttles slowly increase. Observe A/C climb and altitude variation. Speed does not vary.	

Table 2.9
EPR Climb capture test procedures (continued)

8	Select Plot 0, "EPR" Plot.	The Capture Trajectory is displayed. A/C EPR will reach and maintain the reference preset Climb EPR value of 1.7 without ever overshooting 0.1.	
9	Select Clear on ETS TC temporary GUI [D6.2].	All messages will be cleared for ETS TC temporary GUI [D6.2].	
10	Press the escape key on the keyboard.	Simulation will be terminated.	
11	On the Plots menu, select Clear All.	All plots will be cleared.	

2.6.4 EPR Capture Flight Level Change Descent

This test case will perform requirement check of the EPR Capture Mode's ability to decrease thrust and capture a desired predefined descent EPR. The decrease of thrust will cause climb and level change and will be done at constant speed.

Configure FLSIM as presented in Table 2.10:

Table 2.10
EPR Descent capture test simulation startup steps

Step #	Actions	Expected Results	Check Box
1	On the FlsimDe menu, Control Execution, click RAP tab.	Record and Playback menu will be displayed.	
2	On RAP tab, select Playback, click on Filename, choose file 'epr_descent', click on Snapshot/Restore, click on Start Time and enter 00:07:30.	The simulation is set to resume flight at altitude of 30000 feet and speed of 350 knots.	
3	On the FlsimDe menu, Control Execution, click Plot Duration.	The Plot Duration window will be displayed.	
4	On the Plot Duration window, enter 300.	Plotting duration is set to 300 seconds.	
5	On the FlsimDe menu, Control Execution, click Run.	The simulation starts and A/C is flying at altitude of 30000 feet and speed of 350 knots.	
6	Press escape.	Exit simulation to FlsimDe.	
7	On the FlsimDe menu, Control Execution, click Resume no RAP.	The simulation starts and A/C is now held at altitude of 30000 feet and speed of 350 knots.	

Run test procedures presented in Table 2.11:

Table 2.11
EPR Descent capture test procedures

Step #	Actions	Expected Result(s)	Check Box
1	In the Maximum allowed EPR box of the ETS TC Temporary GUI [D6.2], enter an EPR of 1.8. In the Minimum allowed EPR box, enter a value of 1. Click on Activate.	On the ETS Temporary GUI [D6.2] message section, the following message is announced: "New EPR limits set"	
2	Select an Altitude Reference of 25000 feet in the input box of ALT on afcs_ccg panel of CMC Temporary GUI [D6.1] which corresponds to 180 m/s	On afcs_ccg panel of CMC Temporary GUI [D6.1], "25000" is displayed in ALT window.	
3	Select a Speed Reference of 350 knots in the input box of Speed on afcs_ccg panel of CMC Temporary GUI [D6.1] which corresponds to 180 m/s	On afcs_ccg panel of CMC Temporary GUI [D6.1], "350" is displayed in Speed window.	
4	Activate IAS MODE on afcs_ccg panel of CMC Temporary GUI [D6.1] by clicking on the IAS button.	IAS button light turns on.	
5	Select plot 1, "IAS" Plot.	A/C IAS Speed reaches and maintains reference value of 180 m/s	
6	Select and activate ATS mode on CMC Temporary GUI [D6.1].	ATS button light turns on.	
7	Observe A/C descent and altitude variation on the OTW (out the window) scene and display.	Throttles slowly decrease. Observe A/C descent and altitude variation. Speed does not vary.	

Table 2.11
EPR Descent capture test procedures (continued)

8	Select plot 0, "EPR" Plot.	The Capture Trajectory is displayed. A/C EPR will eventually reach and maintain the reference preset EPR descent value of 1.1 without ever overshooting 0.1.	
9	Select Clear on ETS TC temporary GUI [D6.2].	All messages will be cleared for ETS TC temporary GUI [D6.2].	
10	Press the escape key on the keyboard.	Simulation will be terminated.	
11	On the Plots menu, select Clear All.	All plots will be cleared.	

2.6.5 EPR Capture Speed Limits Protection

This test case will perform requirement check of the EPR Capture Mode's ability to respect speed limits. Maximum speed will be chosen to correspond to a lower thrust level than the EPR climb reference, and minimum speed will be chosen to correspond to a higher thrust level than the EPR descent reference. The EPR capture loop would not be able to achieve the reference EPR and the thrust limitation at corresponding speed limits will be observed.

Configure FLSIM as presented in Table 2.12:

Table 2.12
EPR Capture Speed Limits test simulation startup steps

Step #	Actions	Expected Results	Check Box
1	On the FlsimDe menu, Control Execution, click RAP tab.	Record and Playback menu will be displayed.	
2	On RAP tab, select Playback, click on Filename, choose file 'epr_speed_limits', click on Snapshot/Restore, click on Start Time and enter 00:05:30.	The simulation is set to resume flight at altitude of 20000 feet and speed of 300 knots.	
3	On the FlsimDe menu, Control Execution, click Plot Duration.	The Plot Duration window will be displayed.	
4	On the Plot Duration window, enter 300.	Plotting duration is set to 300 seconds.	
5	On the FlsimDe menu, Control Execution, click Run.	The simulation starts and A/C is flying at altitude of 20000 feet and speed of 300 knots.	
6	Press escape.	Exit simulation to FlsimDe.	
7	On the FlsimDe menu, Control Execution, click Resume no RAP.	The simulation starts and A/C is now held at altitude of 20000 feet and speed of 300 knots.	

Run test procedures presented in Table 2.13:

Table 2.13
EPR Capture Speed Limits test procedures

Step #	Actions	Expected Result(s)	Check Box
1	In the Maximum allowed EPR box of the ETS TC Temporary GUI [D6.2], enter an EPR of 2. In the Minimum allowed EPR box, enter a value of 1.1. Click on Activate.	On the ETS Temporary GUI [D6.2] message section, the following message is announced: "New EPR limits set"	
2	In the Maximum allowed IAS box of the ETS TC Temporary GUI [D6.2], enter an IAS of 200 m/s. In the Minimum allowed IAS box, enter a value of 160m/s. Click on Activate.	On the ETS Temporary GUI [D6.2] message section, the following message is announced: "VMO limit set – Minimum operating speed set"	
3	In the Maximum allowed Mach box of the ETS TC Temporary GUI [D6.2], enter a Mach number of 0.8. Click on Activate.	On the ETS Temporary GUI [D6.2] message section, the following message is announced: "MMO limit set"	
4	On the ETS TC Temporary GUI [D6.2], check EPR control and enter an EPR Reference value of 1.8. Click on Activate.	On the ETS Temporary GUI [D6.2] message section, the following message is announced: "EPR Capture Mode Armed"	
5	Select and activate ATS mode on CMC Temporary GUI [D6.1].	ATS button light turns on.	
6	Observe A/C altitude variation on the OTW (out the window) scene and display.	Throttles slowly increase. Observe A/C climb and altitude variation.	
7	Select plot 0, "EPR" Plot.	The Capture Trajectory is displayed. A/C EPR tries to reach reference value of 1.8 but eventually saturates at lower level, corresponding to speed limit as seen on speed plot.	

Table 2.13
EPR Capture Speed Limits test procedures (continued)

8	Select "IAS" Plot.	The Capture Trajectory is displayed. A/C IAS will eventually reach and maintain the maximum value of 200m/s.	
9	On the ETS TC Temporary GUI [D6.2], check EPR control and now enter an EPR Reference value of 1.1. Click on Activate.	On the ETS Temporary GUI [D6.2] message section, the following message is announced: "EPR Capture Mode Armed"	
10	Select and activate ATS mode on CMC Temporary GUI [D6.1].	ATS button light turns on.	
11	Observe A/C altitude variation on the OTW (out the window) scene and display.	Throttles slowly decrease. Observe A/C descent and altitude variation.	
12	Select plot 0, "EPR" Plot.	The Capture Trajectory is displayed. A/C EPR tries to reach reference value of 1.1 but eventually saturates at lower level, corresponding to speed limit.	
13	Select "IAS" Plot.	The Capture Trajectory is displayed. A/C IAS will eventually reach and maintain the minimum value of 150 m/s.	
14	Select Clear on ETS TC temporary GUI [D6.2].	All messages will be cleared for ETS TC temporary GUI [D6.2].	
15	Press the escape key on the keyboard.	Simulation will be terminated.	
16	On the Plots menu, select Clear All.	All plots will be cleared.	

2.6.6 Low Altitude Speed Limit Protection

This test case will perform requirement check of the FAA restriction that the speed does not exceed 250 knots at 10 000 ft or below. The speed will not be able to reach a 300 knots reference and speed saturation at 250 knots will be observed.

Configure FLSIM as presented in Table 2.14:

Table 2.14
Low Altitude Speed Limit test simulation startup steps

Step #	Actions	Expected Results	Check Box
1	On the FlsimDe menu, Control Execution, click RAP tab.	Record and Playback menu will be displayed.	
2	On RAP tab, select Playback, click on Filename, choose file 'epr_speed_limits', click on Snapshot/Restore, click on Start Time and enter 00:02:30.	The simulation is set to resume flight at altitude of 8000 feet and speed of 200 knots.	
3	On the FlsimDe menu, Control Execution, click Plot Duration.	The Plot Duration window will be displayed.	
4	On the Plot Duration window, enter 300.	Plotting duration is set to 300 seconds.	
5	On the FlsimDe menu, Control Execution, click Run.	The simulation starts and A/C is flying at altitude of 8000 feet and speed of 200 knots.	
6	Press escape.	Exit simulation to FlsimDe.	
7	On the FlsimDe menu, Control Execution, click Resume no RAP.	The simulation starts and A/C is now held at altitude of 8000 feet and speed of 200 knots.	

Run test procedures presented in Table 2.15:

Table 2.15
Low Altitude Speed Limit test procedures

Step #	Actions	Expected Result(s)	Check Box
1	In the Maximum allowed EPR box of the ETS TC Temporary GUI [D6.2], enter an EPR of 2. In the Minimum allowed EPR box, enter a value of 1.1. Click on Activate.	On the ETS Temporary GUI [D6.2] message section, the following message is announced: "New EPR limits set"	
2	In the Maximum allowed IAS box of the ETS TC Temporary GUI [D6.2], enter an IAS of 200 m/s. In the Minimum allowed IAS box, enter a value of 100m/s. Click on Activate.	On the ETS Temporary GUI [D6.2] message section, the following message is announced: "VMO limit set – Minimum operating speed set"	
3	In the Maximum allowed Mach box of the ETS TC Temporary GUI [D6.2], enter a Mach number of 0.8. Click on Activate.	On the ETS Temporary GUI [D6.2] message section, the following message is announced: "MMO limit set"	
4	Select an IAS of 300 knots, corresponding to a speed of ~150 m/s, in the input box of IAS on afcs_ccg panel of CMC Temporary GUI [D6.1]. Activate ATS MODE on afcs_ccg panel of CMC Temporary GUI [D6.1] by clicking on the ATS button on the mode selection panel.	On the ETS Temporary GUI [D6.2] message section, the following message is announced: "Speed Capture mode armed" On afcs_ccg panel of CMC Temporary GUI [D6.1], 300 is displayed in IAS window. The ATS mode button's light is on.	
5	Observe A/C speed increasing on the OTW (out the window) scene and display	A/C increases speed.	

Table 2.15
Low Altitude Speed Limit test procedures (continued)

6	Select "IAS" Plot.	The Capture Trajectory is displayed. A/C IAS increases and tries to reach the reference value of 150 m/sec without success, saturating at maximum value of 129 m/sec (250 knots). On the ETS Temporary GUI message section the following message is announced: "Speed tracking error greater then 10 knots"	
7	Select Clear on ETS TC temporary GUI [D6.2].	All messages will be cleared for ETS TC temporary GUI [D6.2].	
8	Press the escape key on the keyboard.	Simulation will be terminated.	
9	On the Plots menu, select Clear All.	All plots will be cleared.	

2.6.7 Speed Arm and Reference Limit Protection

This test case will perform requirement check of the Speed Arm Mode's ability to respect the Speed limits for Speed reference inputs (Mach and IAS) and the functioning of Speed Arm mode logic.

Configure FLSIM as presented in Table 2.16:

Table 2.16
Speed Arm and Reference Limit test simulation startup steps

Step #	Actions	Expected Results	Check Box
1	On the FlsimDe menu, Control Execution, click RAP tab.	Record and Playback menu will be displayed.	
2	On RAP tab, select Playback, click on Filename, choose file 'speed_arm', click on Snapshot/Restore, click on Start Time and enter 00:07:30.	The simulation is set to resume flight at altitude of 30000 feet and speed of 350 knots.	
3	On the FlsimDe menu, Control Execution, click Plot Duration.	The Plot Duration window will be displayed.	
4	On the Plot Duration window, enter 300.	Plotting duration is set to 300 seconds.	
5	On the FlsimDe menu, Control Execution, click Run.	The simulation starts and A/C is flying at altitude of 30000 feet and speed of 350 knots.	
6	Press escape.	Exit simulation to FlsimDe.	
7	On the FlsimDe menu, Control Execution, click Resume no RAP.	The simulation starts and A/C is now held at altitude of 30000 feet and speed of 350 knots.	

Run test procedures presented in Table 2.17:

Table 2.17
Speed Arm and Reference Limit test procedures

Step #	Actions	Expected Result(s)	Check Box
1	In the Maximum allowed IAS box of the ETS TC Temporary GUI [D6.2], enter an IAS of 250 m/s. In the Minimum allowed IAS box, enter a value of 120m/s. Click on Activate.	On the ETS Temporary GUI [D6.2] message section, the following message is announced: "VMO limit set – Minimum operating speed set"	
2	In the Maximum allowed Mach box of the ETS TC Temporary GUI [D6.2] enter a Mach number of 0.9. Click on Activate.	On the ETS Temporary GUI [D6.2] message section, the following message is announced: "MMO limit set"	
3	Select an IAS of 585 knots in the input box of IAS on afcs_ccg panel of CMC Temporary GUI [D6.1] which corresponds to 300 m/s. Activate ATS MODE on afcs_ccg panel of CMC Temporary GUI [D6.1] by clicking on the ATS button on the mode selection panel.	On the ETS Temporary GUI [D6.2] message section, the following message is announced: "Selected Speed not within acceptable limits, speed Reference value adjusted to maximum IAS". The reference IAS is now 250 m/sec (486 knots) The ATS mode button's light is on.	
4	Select an IAS of 195 knots in the input box of IAS on afcs_ccg panel of CMC Temporary GUI [D6.1]. Activate ATS MODE on afcs_ccg panel of CMC Temporary GUI [D6.1] by clicking on the ATS button on the mode selection panel.	On the ETS Temporary GUI [D6.2] message section, the following message is announced: "Selected Speed not within acceptable limits, speed Reference value adjusted to minimum IAS" The reference IAS is now 120 m/sec (233 knots) The ATS mode button's light is on	

Table 2.17
Speed Arm and Reference Limit test procedures (continued)

5	<p>Select a Mach number of 3 in the input box of Mach Number on afcs_ccg panel of CMC Temporary GUI [D6.1]. Activate ATS MODE on afcs_ccg panel of CMC Temporary GUI [D6.1] by clicking on the ATS button on the mode selection panel.</p>	<p>On the ETS Temporary GUI [D6.2] message section, the following message is announced: "Selected Speed not within acceptable limits, speed Reference value adjusted to maximum Mach number" The reference Mach is now 0.9 The ATS mode button's light is on</p>	
6	<p>Select a Mach number of 0 in the input box of Mach Number on afcs_ccg panel of CMC Temporary GUI [D6.1]. Activate ATS MODE on afcs_ccg panel of CMC Temporary GUI [D6.1] by clicking on the ATS button on the mode selection panel.</p>	<p>On the ETS Temporary GUI [D6.2] message section, the following message is announced: "Selected Speed not within acceptable limits, speed Reference value adjusted to minimum Mach number" The reference Mach is now 0.2 The ATS mode button's light is on</p>	
7	<p>Select Clear on ETS TC temporary GUI [D6.2].</p>	<p>All messages will be cleared for ETS TC temporary GUI [D6.2].</p>	
8	<p>Press the escape key on the keyboard.</p>	<p>Simulation will be terminated.</p>	
9	<p>On the Plots menu, select Clear All.</p>	<p>All plots will be cleared.</p>	

2.6.8 Speed Cruise IAS Capture

This test case will perform requirement check of the Speed Capture IAS Mode's ability to increase or decrease thrust, and capture a desired IAS speed. This test will be performed at Cruise conditions and altitude must be maintained throughout speed variation.

Configure FLSIM as presented in Table 2.18:

Table 2.18
Airspeed Cruise Capture test simulation startup steps

Step #	Actions	Expected Results	Check Box
1	On the FlsimDe menu, Control Execution, click RAP tab.	Record and Playback menu will be displayed.	
2	On RAP tab, select Playback, click on Filename, choose file 'spd_cruise_ias', click on Snapshot/Restore, click on Start Time and enter 00:09:30.	The simulation is set to resume flight at altitude of 30000 feet and speed of 400 knots.	
3	On the FlsimDe menu, Control Execution, click Plot Duration.	The Plot Duration window will be displayed.	
4	On the Plot Duration window, enter 300.	Plotting duration is set to 300 seconds.	
5	On the FlsimDe menu, Control Execution, click Run.	The simulation starts and A/C is flying at altitude of 30000 feet and speed of 400 knots.	
6	Press escape.	Exit simulation to FlsimDe.	
7	On the FlsimDe menu, Control Execution, click Resume no RAP.	The simulation starts and A/C is now held at altitude of 30000 feet and speed of 400 knots.	

Run test procedures presented in Table 2.19:

Table 2.19
Airspeed Cruise Capture test procedures

Step #	Actions	Expected Result(s)	Check Box
1	In the Maximum allowed IAS box of the ETS TC Temporary GUI [D6.2], enter an IAS of 400 m/s. In the Minimum allowed IAS box, enter a value of 100m/s. Click on Activate.	On the ETS Temporary GUI [D6.2] message section, the following message is announced: "VMO limit set – Minimum operating speed set"	
2	In the Maximum allowed Mach box of the ETS TC Temporary GUI [D6.2] enter a Mach number of 1. Click on Activate.	On the ETS Temporary GUI [D6.2] message section, the following message is announced: "MMO limit set"	
3	Activate ALT HOLD MODE on afcs_ccg panel of CMC Temporary GUI [D6.1] by clicking on the ALT HOLD button. The reference altitude is the current altitude of 30000 feet.	ALT HOLD button light turns on.	
4	Select an IAS of 486 knots in the input box of IAS on afcs_ccg panel of CMC Temporary GUI [D6.1]. (corresponding to 250m/sec) Activate IAS MODE on afcs_ccg panel of CMC Temporary GUI [D6.1] by clicking on the IAS button. Activate ATS MODE on afcs_ccg panel of CMC Temporary GUI [D6.1] by clicking on the ATS button on the mode selection panel.	On the ETS Temporary GUI [D6.2] message section, the following message is announced: "Speed Capture mode armed" On afcs_ccg panel of CMC Temporary GUI [D6.1], 486 is displayed in IAS window. The ATS mode button's light is on.	
5	Observe A/C speed increasing on the OTW (out the window) scene and display	A/C increases speed.	

Table 2.19
Airspeed Cruise Capture test procedures (continued)

6	Select "IAS" Plot.	The Capture Trajectory is displayed. A/C IAS will eventually reach and maintain the reference value of 250 m/sec without ever overshooting 10 m/sec and without ever violating speed limits of 400 m/sec maximum value and 100 m/sec minimum value,	
7	Select "Height" Plot.	The Altitude Trajectory is displayed. A/C Altitude will be maintained at hold value and never vary more then 200 ft from reference value of 30 000 feet.	
8	Select "Forward Acceleration" Plot.	The forward acceleration is displayed. A/C forward acceleration never exceeds 0.1g.	
9	Select "Climb rate" Plot.	Climb rate is displayed. A/C climb rate never exceeds 0.1g.	
10	Wait a while after system stabilizes at reference input of 250m/sec IAS.		
11	Select a reference IAS of 389 knots, corresponding to 200 m/s, in the input box of IAS on afcs_ccg panel of CMC Temporary GUI [D6.1].	On the ETS Temporary GUI [D6.2] message section, the following message is announced: "Speed Capture mode armed" On afcs_ccg panel of CMC Temporary GUI [D6.1], 389 is displayed in IAS window.	

Table 2.19
Airspeed Cruise Capture test procedures (continued)

12	Observe A/C speed decreasing on the OTW (out the window) scene and display	A/C decreases speed.	
13	Select "IAS" Plot.	The Capture Trajectory is displayed. A/C IAS decreases and will eventually reach and maintain the reference value of 200 m/sec without ever overshooting 10 m/sec and without ever violating speed limits. of 400 m/sec maximum value and 100 m/sec minimum value,	
14	Select "Height" Plot.	The Altitude Trajectory is displayed. A/C Altitude will be maintained at hold value and never vary more then 200 ft from reference value. of 30 000 feet.	
15	Select "Forward Acceleration" Plot.	The forward acceleration is displayed. A/C forward acceleration never exceeds 0.1g.	
16	Select "Climb rate" Plot.	Climb rate is displayed. A/C climb rate never exceeds 0.1g.	
17	Select Clear on ETS TC temporary GUI [D6.2].	All messages will be cleared for ETS TC temporary GUI [D6.2].	
18	Press the escape key on the keyboard.	Simulation will be terminated.	
19	On the Plots menu, select Clear All.	All plots will be cleared.	

2.6.9 Speed Cruise MACH Capture

This test case will perform requirement check of the Speed Capture MACH Mode's ability to increase or decrease thrust, and capture a desired MACH speed. This test will be performed at Cruise conditions and altitude must be maintained throughout speed variation.

Configure FLSIM as presented in Table 2.20:

Table 2.20
Mach Cruise Capture test simulation startup steps

Step #	Actions	Expected Results	Check Box
1	On the FlsimDe menu, Control Execution, click RAP tab.	Record and Playback menu will be displayed.	
2	On RAP tab, select Playback, click on Filename, choose file 'spd_cruise_mn', click on Snapshot/Restore, click on Start Time and enter 00:09:30.	The simulation is set to resume flight at altitude of 30000 feet and speed of 400 knots.	
3	On the FlsimDe menu, Control Execution, click Plot Duration.	The Plot Duration window will be displayed.	
4	On the Plot Duration window, enter 300.	Plotting duration is set to 300 seconds.	
5	On the FlsimDe menu, Control Execution, click Run.	The simulation starts and A/C is flying at altitude of 30000 feet and speed of 400 knots.	
6	Press escape.	Exit simulation to FlsimDe.	
7	On the FlsimDe menu, Control Execution, click Resume no RAP.	The simulation starts and A/C is now held at altitude of 30000 feet and speed of 400 knots.	

Run test procedures presented in Table 2.21:

Table 2.21
Mach Cruise Capture test procedures

Step #	Actions	Expected Result(s)	Check Box
1	In the Maximum allowed IAS box of the ETS TC Temporary GUI [D6.2], enter an IAS of 400 m/s. In the Minimum allowed IAS box, enter a value of 100m/s. Click on Activate.	On the ETS Temporary GUI [D6.2] message section, the following message is announced: "VMO limit set – Minimum operating speed set"	
2	In the Maximum allowed Mach box of the ETS TC Temporary GUI [D6.2] enter a Mach number of 1. Click on Activate.	On the ETS Temporary GUI [D6.2] message section, the following message is announced: "MMO limit set"	
3	Activate ALT HOLD MODE on afcs_ccg panel of CMC Temporary GUI [D6.1] by clicking on the ALT HOLD button. The reference altitude is the current altitude of 30000 feet.	ALT HOLD button light turns on.	
4	Select a Mach number of 0.85 in the input box of Mach numbers on afcs_ccg panel of CMC Temporary GUI [D6.1]. Activate ATS MODE on afcs_ccg panel of CMC Temporary GUI [D6.1] by clicking on the ATS button on the mode selection panel. Activate MACH MODE on afcs_ccg panel of CMC Temporary GUI [D6.1] by clicking on the MACH button.	On the ETS Temporary GUI [D6.2] message section, the following message is announced: "Speed Capture mode armed" On afcs_ccg panel of CMC Temporary GUI [D6.1], "0.85" is displayed in Mach number window. The ATS mode button's light is on.	
5	Observe A/C speed increasing on the OTW (out the window) scene and display	A/C increases speed.	

Table 2.21
Mach Cruise Capture test procedures (continued)

6	Select "Mach number" Plot.	The Capture Trajectory is displayed. A/C Mach number will eventually reach and maintain the reference value of 0.85 without ever overshooting 0.02 and without ever reaching maximum Mach value of 1.	
7	Select "Height" Plot.	The Altitude Trajectory is displayed. A/C Altitude will be maintained at hold value and never vary more than 200 ft from reference value of 30000 feet.	
8	Select "Forward Acceleration" Plot.	The forward acceleration is displayed. A/C forward acceleration never exceeds 0.1g.	
9	Select "Climb rate" Plot.	Climb rate is displayed. A/C climb rate never exceeds 0.1g.	
10	Wait till system stabilizes at reference input of 0.85 Mach.		
11	Select a reference Mach number of 0.6 in the input box of Mach number on afcs_ccg panel of CMC Temporary GUI [D6.1].	On the ETS Temporary GUI [D6.2] message section, the following message is announced: "Speed Capture mode armed" On afcs_ccg panel of CMC Temporary GUI [D6.1], "0.6" is displayed in Mach Number window.	

Table 2.21
Mach Cruise Capture test procedures (continued)

12	Observe A/C speed decreasing on the OTW (out the window) scene and display	A/C decreases speed.	
13	Select "Mach Number" Plot.	The Capture Trajectory is displayed. A/C Mach number decreases and will eventually reach and maintain the reference value of 0.6 without ever overshooting 0.02	
14	Select "Height" Plot.	The Altitude Trajectory is displayed. A/C Altitude will be maintained at hold value and never vary more than 200 ft from reference value of 30 000 feet.	
15	Select "Forward Acceleration" Plot.	The forward acceleration is displayed. A/C forward acceleration never exceeds 0.1g.	
16	Select "Climb rate" Plot.	Climb rate is displayed. A/C climb rate never exceeds 0.1g.	
17	Select Clear on ETS TC temporary GUI [D6.2].	All messages will be cleared for ETS TC temporary GUI [D6.2].	
18	Press the escape key on the keyboard.	Simulation will be terminated.	
19	On the Plots menu, select Clear All.	All plots will be cleared.	

2.6.10 Speed Capture Mode EPR Limits Protection

This test case will perform requirement check of the Speed Capture Mode's ability to respect EPR limits. Maximum EPR will be chosen to correspond to a lower thrust level than the Speed reference, and then minimum EPR will be chosen to correspond to a higher thrust level than the Speed reference. The Speed capture loop would not be able to achieve the reference speed and the thrust limitation at corresponding EPR limits will be observed.

Table 2.22

Speed Capture EPR Limits Protection test simulation startup steps

Step #	Actions	Expected Results	Check Box
1	On the FlsimDe menu, Control Execution, click RAP tab.	Record and Playback menu will be displayed.	
2	On RAP tab, select Playback, click on Filename, choose file 'spd_epr_limits', click on Snapshot/Restore, click on Start Time and enter 00:09:30.	The simulation is set to resume flight at altitude of 30000 feet and speed of 400 knots.	
3	On the FlsimDe menu, Control Execution, click Plot Duration.	The Plot Duration window will be displayed.	
4	On the Plot Duration window, enter 300.	Plotting duration is set to 300 seconds.	
5	On the FlsimDe menu, Control Execution, click Run.	The simulation starts and A/C is flying at altitude of 30000 feet and speed of 400 knots.	
6	Press escape.	Exit simulation to FlsimDe.	
7	On the FlsimDe menu, Control Execution, click Resume no RAP.	The simulation starts and A/C is now held at altitude of 30000 feet and speed of 400 knots.	

Configure FLSIM as presented in Table 2.22 and Run test procedures presented in Table 2.23:

Table 2.23
Speed Capture EPR Limits Protection test procedures

Step #	Actions	Expected Result(s)	Check Box
1	In the Maximum allowed EPR box of the ETS TC Temporary GUI [D6.2], enter an EPR of 1.5. In the Minimum allowed EPR box, enter a value of 1.3. Click on Activate.	On the ETS Temporary GUI [D6.2] message section, the following message is announced: "New EPR limits set"	
2	In the Maximum allowed IAS box of the ETS TC Temporary GUI [D6.2], enter an IAS of 300 m/s. In the Minimum allowed IAS box, enter a value of 120m/s. Click on Activate.	On the ETS Temporary GUI message section the following message is announced: "VMO limit set – Minimum operating speed set"	
3	In the Maximum allowed Mach box of the ETS TC Temporary GUI [D6.2], enter a Mach number of 0.9. Click on Activate.	On the ETS Temporary GUI [D6.2] message section, the following message is announced: "MMO limit set"	
4	Select an IAS of 545 knots, corresponding to a speed of ~280 m/s, in the input box of IAS on afcs_ccg panel of CMC Temporary GUI [D6.1]. Activate ATS MODE on afcs_ccg panel of CMC Temporary GUI [D6.1] by clicking on the ATS button on the mode selection panel.	On the ETS Temporary GUI [D6.2] message section, the following message is announced: "Speed Capture mode armed" On afcs_ccg panel of CMC Temporary GUI [D6.1], 545 is displayed in IAS window. The ATS mode button's light is on.	
5	Observe A/C speed increasing on the OTW (out the window) scene and display	A/C increases speed.	

Table 2.23
Speed Capture EPR Limits Protection test procedures (continued)

6	Select "IAS" Plot.	<p>The Capture Trajectory is displayed. A/C IAS increases and tries to reach the reference value of 280 m/sec without success, saturating at a point corresponding to the engine EPR limit.</p> <p>On the ETS Temporary GUI message section the following message is announced: "Speed tracking error greater then 10 knots"</p>	
7	Select "EPR" Plot.	<p>The Capture Trajectory is displayed. A/C EPR will eventually reach and maintain the maximum value of 1.5.</p>	
8	<p>Select a reference IAS of 253 knots, corresponding to 130 m/s, in the input box of IAS on afcs_ccg panel of CMC Temporary GUI [D6.1].</p>	<p>On the ETS Temporary GUI [D6.2] message section, the following message is announced: "Speed Capture mode armed"</p> <p>On afcs_ccg panel of CMC Temporary GUI [D6.1], 253 is displayed in IAS window.</p>	
9	<p>Observe A/C speed decreasing on the OTW (out the window) scene and display</p>	<p>A/C decreases speed.</p>	

Table 2.23
Speed Capture EPR Limits Protection test procedures (continued)

10	Select "IAS" Plot.	<p>The Capture Trajectory is displayed. A/C IAS decreases and tries to reach the reference value of 130 m/sec without success saturating at a point corresponding to the engine EPR limit.</p> <p>On the ETS Temporary GUI message section the following message is announced: "Speed tracking error greater then 10 knots"</p>	
11	Select "EPR" Plot.	<p>The Capture Trajectory is displayed. A/C EPR will eventually reach and maintain the minimum value of 1.3.</p>	
12	Select Clear on ETS TC temporary GUI [D6.2].	<p>All messages will be cleared for ETS TC temporary GUI [D6.2].</p>	
13	Press the escape key on the keyboard.	<p>Simulation will be terminated.</p>	
14	On the Plots menu, select Clear All.	<p>All plots will be cleared.</p>	

REFERENCES

- [1] Andry A.N., Shapiro E.Y. & Chung J.C., (1983) Eigenstructure assignement for linear systems, *IEEE Trans. On Aerospace and Electronic Systems*, pp.711-728.
- [2] Bader D.L. (1990). Automatic Throttle Control for Business and Commuter Aircraft, *Digital Avionics Systems Conference*, 1990 Proceedings, IEEE/AIAA/NASA 9th, p.205-211.
- [3] Blakelock J.H. (1991) *Automatic control of aircraft and missiles*, John Wiley & Sons, Inc.
- [4] Bruce K.R. (1988) *Integrated Autopilot/Autothrottle based on total energy concept*. NASA Scientifical and technical division, prepared for Langley Research center.
- [5] Collinson R.P.G. (1996) *Introduction to Avionics*. Microwave Technologies series 11, Chapman & Hall, London, UK.
- [6] Derbois A. (2002) *Mise en œuvre du modèle et synthèse d'une loi de commande par commutation de compensateurs pour un avion commercial*. Mémoire, École Polytechnique de Montréal.
- [7] Dynamic Test Bed Documentation (2007), *DTB Architecture*, CMC Electronics Internal documentation.
- [8] Flightsim Reference Guide 9.1 (2005), *Flig htsim Reference Guide*, Engenuity Technologies.

- [9] Hanke C.R., & Nordwall D.R. (1970) *The Simulation of a Jumbo Jet Transport Aircraft*. II, The Boeing Company, Wichita, Kansas.

- [10] Jeppesen Sanderson Training Products (1974) *Avionics Fundamentals*, United Airlines.

- [11] Kautsky J., Nichols N.K., & Dooren P.V., (1985) Robust pole assignment in linear state feedback, *Int. J. Control*, 41(5), pp. 1129-1155.

- [12] Lambregts A. A. (1983) *Vertical Flight Path and Speed Control Autopilot Design Using Total Energy Principles*. AIAA Paper 83-2239CP, August 1983.

- [13] Lambregts A. A. (2000) *Total energy based flight control system*. United States Patent, Patent number 6 062 513.

- [14] Larramendy P. (1991) Examination of a Longitudinal Law using the Eigenvectors method allowing Thrust/Pitch Decoupling. *Automatic Control World Congress 1990. 'In the Service of Mankind'*. Proceedings of the 11th Triennial World Congress of the International Federation of Automatic Control. p 61-8 (vol.3)

- [15] Larson W.R., & Stephan L.P. (1980) *Mach Hold control circuit for autothrottle system*. United States Patent, Patent number 4 205 814.

- [16] Leith D. J., & Leithead W. E. (2000) Survey of gain-scheduling analysis and design. *International Journal of Control*, Volume 73, Number 11, pp. 1001-1025 (25).

- [17] Liptak, Bela (1995). *Instrument Engineers' Handbook: Process Control*. Radnor, Pennsylvania: Chilton Book Company, 20-29

- [18] Liu, H.H.T. (2001) Design combination in integrated flight control, *American Control Conference*. Proceedings of the 2001, Volume 1, 25-27 June 2001, p494 - 499 vol.1.
- [19] Mahoney M. J. (last consulted in 2009), *Pointing an Instrument on an Airborne Platform*,
<http://mtp.jpl.nasa.gov/notes/pointing/pointing.html>
- [20] Mclean D., (1990) *Automatic Flight Control Systems*, Prentice Hall International.
- [21] Moir, I., & Seabridge, A. (2003) *Civil Avionics Systems*. AIAA Inc., Reston and Professional Publishing, UK.
- [22] Nelson R.C. (1998). *Flight Stability and automatic control*. McGraw-Hill, New York, NY.
- [23] Oshe'a, J. (2000) Dynamique du vol et autopilotage. *Note de cours*, École Polytechnique de Montréal.
- [24] Piou, J.E., & Sobel, K.M. (1996) Application of gain scheduled eigenstructure assignment to flight control design. *Proceedings of the 1996 IEEE International Conference on Control Applications*, pp. 101-106
- [25] Ramga W. & Ewertz W. (last consulted in 2009), *Turbofan*, Modern Aircraft Turbine Engines of the 21st Century
<http://modernaircraftengines.com/turbofan.htm>
- [26] Robbins R.E., & Simpson R.D. (1991) *Method and apparatus for aircraft pitch and thrust axes control*. United States Patent, Patent number 5 031 102.

- [27] RTCA/DO-178B, *Software Considerations in Airborne Systems and Equipment Certification*. Notice N8110.49: Software Approval Guidelines.
- [28] Scott J. (last consulted in 2009), *Parts of an Aircraft*, Aerospace Web
<http://www.aerospaceweb.org/question/design/q0101.shtml>
- [29] Spitzer C.R. (2001) *The Avionics Handbook*, CRC Press LLC.
- [30] US Centennial of Flight Commission (last consulted in 2009), *Four Forces of Flight*,
http://www.centennialofflight.gov/essay/Dictionary/four_forces/DI24.htm
- [31] User Guide (2009), *Aerospace Toolbox 2*, Matlab, Mathworks.
<http://www.mathworks.com/access/helpdesk/help/toolbox/aerotbx/helptoc.html>
- [32] Vakil S. S., & Hansman R. J. (1998) Applying a common conceptual model approach to vertical navigation automation. HESSD.
- [33] Van, Doren, Vance J. (July 1, 2003). Loop Tuning Fundamentals. *Control Engineering*. Red Business Information.
- [34] Wikipedia (last consulted in 2007), *PID controller*,
http://en.wikipedia.org/wiki/PID_controller