

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.



Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

**The Enhancement of a Flight Simulator System with
Teaching and Research Applications**

Peter Lawn

A Thesis

in

The Department

of

Mechanical Engineering

**Presented in Partial Fulfilment of the Requirements
for the Degree of Master of Applied Science at
Concordia University
Montreal, Quebec, Canada**

April 1998

© Peter Lawn, 1998



National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-39982-6

Canada

ABSTRACT

The Enhancement of a Flight Simulator System with Teaching and Research Applications

Peter Lawn

Modern computers, offering high performance at a low cost have created the opportunity to upgrade the computer hardware of Concordia's flight simulator and to develop / implement a more sophisticated and versatile simulation software. The widely used 80X86 family of personal computers was chosen lending the new simulation software a high degree of portability, so that it can be used in the laboratory, in the class room or in the student's home. The simulation program was developed to be applicable as a research and teaching tool in the area of aircraft stability, control, systems, and human factors. This was achieved by designing a large flexibility in definition of the simulated aircraft's dimensions and coefficients, and by providing different options to monitor and store real time data. To offer insight into aircraft characteristics, a second program was developed to estimate the longitudinal stability of an aircraft using the USAF DATCOM method as a basis. This program demonstrates all the steps in the solution including the use of all necessary graphical data. Following an introduction outlining the history and objectives of this work, chapter two details the development of the dynamic flight model and the hybrid dynamic and kinematic ground model. Chapter three describes the implementation of the models into a working simulation program. Chapter four and five describe in detail the operation and simulation of the flight and navigational equipment typically found on a general aviation aircraft. The upgraded hardware is presented in chapter six, including custom built analog interface devices. Lastly chapter seven outlines the DATCOM method and its implementation into a computer program.

ACKNOWLEDGMENTS

The author wishes to express his gratitude to his supervisors Dr. J.V. Svoboda and Dr. K. Foster, for their guidance in completing this thesis.

The invaluable assistance of Mr. Gilles Huard in developing the custom electronic hardware is also greatly appreciated.

This thesis is dedicated to the memory of Dr. Jaan Saber.

TABLE OF CONTENTS

	Page
LIST OF FIGURES	xii
LIST OF TABLES	xvi
NOMENCLATURE	xvii
1 INTRODUCTION	1
1.1 Historical Background of Flight Simulation	1
1.1.1 Early Flight Simulation Systems	1
1.1.2 Introduction of the Analog Computer	3
1.1.3 Introduction of the Digital Computer	4
1.1.4 Full Flight Simulators and Flight Training Devices	4
1.1.5 Simulator Motion Systems	5
1.1.6 Simulator Visual Systems	6
1.2 Flight Simulators as Research and Design Tools	9
1.3 The Original Concordia Flight Simulator System	10
1.4 Thesis Objectives	11
1.4.1 Hardware Upgrade	12
1.4.2 Flight Simulator as a Teaching Aid	13
1.4.3 Flight Simulator as a Research Tool	14
1.4.4 Estimation of Stability and Control Coefficients	15
1.4.5 Summary of Objectives	15
2 AIRCRAFT DYNAMIC MODELLING	17
2.1 Coordinate Systems	17
2.1.1 The Body Axis System	17
2.1.2 The Stability Axis System	19
2.1.3 The Fixed Axis System	20
2.1.4 Euler Angles	21
2.2 Equations of Motion	22

2.2.1	Rotational Equations of Motion	22
2.2.2	Translational Equations of Motion	25
2.3	Aerodynamic Forces and Moments Modelling	26
2.4	Thrust Model	29
2.5	Gravity Model	30
2.6	Ground Model	31
2.6.1	Ground Kinematic Model	31
2.6.2	Ground Dynamic Model	33
2.7	Atmospheric Model	35
2.7.1	Standard Atmospheric Conditions Below 11000 Meters ...	36
2.7.2	Standard Atmospheric Conditions Above 11000 Meters ...	38
3	COMPUTER CODE IMPLEMENTATION OF FLIGHT MODEL	40
3.1	Implementation of Flight Model Simulation	40
3.1.1	Program Initial Condition	41
3.1.2	Simulation Loop	42
3.1.2.1	The Program Timer	42
3.1.2.2	Calculation of Atmospheric Conditions	43
3.1.2.3	Reading of Control Inputs	44
3.1.2.4	Setting of Control Surfaces	44
3.1.2.5	Calculate Aerodynamic Forces and Moments in Stability Axis	45
3.1.2.6	Transformation Aerodynamic Forces and Moments to Body Axis	45
3.1.2.7	Calculation of the Forces and Moments Due to Thrust	46
3.1.2.8	Summation of Forces and Moments in Body Axis ..	46
3.1.2.9	Calculation of Motion in Body Axis	47
3.1.2.10	Transformation of Motion to Fixed Axis System	47
3.2	Simulated Aircraft Configuration	48
3.2.1	Stability and Control Coefficients	48

3.2.2	Control Surface Deflection	48
3.2.3	Engine Configuration	49
3.2.4	Landing Gear Configuration	49
3.2.5	Initial Position and Atmospheric Conditions	50
3.3	Simulation Control Configuration	51
3.4	Computer Display Configuration	53
3.4.1	Flight Instrument Display	53
3.4.2	Map Display	56
3.4.3	Strip Chart Display	57
3.5	Optional Visual System	58
3.6	Analysis Configuration	58
 4	 AIRCRAFT FLIGHT INSTRUMENTS, OPERATION AND SIMULATION	 59
4.1	The Magnetic Compass	60
4.1.1	Magnetic Compass Operation and Simulation	61
4.1.1.1	The Earth's Magnetic Field	61
4.1.1.2	Magnetic Compass Acceleration and Turning Errors	65
4.2	The Air Speed Indicator	67
4.2.1	ASI Operation	67
4.2.2	ASI Simulation	70
4.3	The Attitude Indicator	72
4.3.1	Attitude Indicator Operation	73
4.3.2	Attitude Indicator Simulation	73
4.4	The Altimeter	74
4.4.1	Altimeter Operation	75
4.4.2	Altimeter Simulation	76
4.5	The Turn Coordinator	78
4.5.1	Turn Coordinator Operation	79
4.5.2	Turn Coordinator Simulation	79

4.6	The Directional Gyrocompass	80
4.6.1	Gyroscopic Compass Operation	80
4.6.2	Gyroscopic Compass Simulation	81
4.7	The Vertical Speed Indicator	81
4.7.1	VSI Operation	82
4.7.2	VSI Simulation	82
5	RADIO NAVIGATION SYSTEMS, OPERATION AND SIMULATION .	84
5.1	The Automatic Direction Finder	84
5.1.1	ADF Theory of Operation	86
5.1.2	ADF Simulation	89
5.2	The Very High Frequency Omni-Range Receiver	91
5.2.1	VOR Theory of Operation	94
5.2.2	VOR Simulation	95
5.3	The Distance Measuring Equipment	98
5.3.1	DME Theory of Operation	98
5.3.2	DME Simulation	99
5.4	The Instrument Landing System	99
5.4.1	ILS Theory of Operation	101
5.4.2	ILS Simulation	103
6	HARDWARE CONFIGURATION	105
6.1	Full Simulated Cockpit Environment	105
6.1.1	The Original Simulator System	105
6.1.2	Objectives of Upgraded System	106
6.1.3	Communication Systems	107
6.1.4	Cockpit Signal Requirement	107
6.1.4.1	Analog Output Requirements	108
6.1.4.1.1	Brushless DC Motors	108
6.1.4.1.2	Galvometer Gauges	109

6.1.4.1.3	Stepper Motor Driven Attitude Indicator	110
6.1.5	Analog Output Card	111
6.1.6	Analog Input Requirements	114
6.1.7	Digital Outputs	116
6.1.8	Digital Inputs	117
6.1.9	Radio Stack Requirements	117
6.2	Desktop Configuration	118
6.2.1	Interfacing With the PC Game Port	118
6.2.2	Interfacing With the PC Mouse	119
7	STABILITY AND CONTROL COEFFICIENT CALCULATION	121
7.1	Aircraft Design Constraints	121
7.1.1	Fuselage Geometry	123
7.1.2	Wing Geometry	123
7.1.3	Horizontal Stabilizer Geometry	125
7.1.4	Vertical Stabilizer Geometry	126
7.2	Aerodynamics	126
7.2.1	Lift	126
7.2.1.1	Wing Lift Characteristics	128
7.2.1.2	Horizontal Stabilizer Lift Characteristics	128
7.2.1.3	Fuselage Lift Characteristics	129
7.2.2	Drag	129
7.2.2.1	Wing Profile Drag Characteristics	130
7.2.2.2	Wing Induced Drag Characteristics	133
7.2.2.3	Horizontal Stabilizer Drag Characteristics	135
7.2.2.4	Vertical Stabilizer Drag Characteristics	135
7.2.2.5	Fuselage Drag Characteristics	135
7.2.3	Pitch	136
7.3	Aircraft Aerodynamics	139
7.3.1	Aircraft Lift Coefficient	139

7.3.1.1	Aircraft Lift Coefficient at Zero Angle of Attack	140
7.3.1.2	Aircraft Lift Curve Slope	141
7.3.1.3	Aircraft Zero Lift Angle of Attack	142
7.3.1.4	Aircraft Lift Coefficient Due to Rate of Change of Angle of Attack	142
7.3.1.5	Aircraft Lift Coefficient Due to Pitch Rate	143
7.3.1.6	Aircraft Lift Coefficient Due to Elevator Deflection . .	144
7.3.2	Aircraft Drag	147
7.3.2.1	Aircraft Drag at Zero Angle of Attack	148
7.3.2.2	Aircraft Drag at Angle of Attack	148
7.3.3	Aircraft Pitch Moment Coefficient	148
7.3.3.1	Aircraft Pitch moment Coefficient at Zero Lift	148
7.3.3.2	Aircraft Pitch Moment Coefficient Due to Angle of Attack	150
7.3.3.3	Aircraft Pitch Moment Coefficient at Zero Angle of Attack	151
7.3.3.4	Aircraft Pitch Moment Coefficient Due to Rate of Change of Angle of Attack	152
7.3.3.5	Aircraft Pitch Moment Coefficient Due to Pitch Rate	152
7.3.3.6	Aircraft Pitch Moment Coefficient Due to Elevator Deflection	153
8	CONCLUSION	154
8.1	General Review	154
8.2	Overview and Discussion	154
8.2.1	The Dynamic Model	154
8.2.2	Computer Implementation of the Dynamic Model	156
8.2.3	Aircraft Instruments, Operation and Simulation	159
8.2.4	Simulation Hardware Selection	159
8.2.5	Aircraft Stability and Control Estimation	160
8.3	Future Work	161

8.3.1	Sound Simulation	161
8.3.2	Navigational Instruments	161
8.3.3	Propulsion Modelling	162
8.3.4	Control Loading	162
8.4	Concluding Remarks	163
REFERENCES		164
APPENDIX 1	SIMULATION GLOBAL VARIABLES	170
APPENDIX 2	SIMULATION TIMING FUNCTION	182
APPENDIX 3	ICAO STANDARD ATMOSPHERE SIMULATION FUNCTION	185
APPENDIX 4	CALCULATION OF FORCES IN THE BODY AXIS	188
APPENDIX 5	CALCULATION OF MOMENTS IN THE BODY AXIS ..	192
APPENDIX 6	CALCULATION OF MOTION IN THE BODY AXIS	196
APPENDIX 7	CALCULATION OF MOTION IN THE FIXED AXIS	199
APPENDIX 8	CALCULATION OF POSITION IN THE FIXED AXIS ..	203
APPENDIX 9	MAGNETIC VARIATION AND DIP MODEL	206
APPENDIX 10	RANGE AND BEARING FUNCTION	213
APPENDIX 11	VOR, LOCALIZER AND GLIDESLOPE FREQUENCIES	217
APPENDIX 12	SELECTED STEPPER MOTOR SPECIFICATIONS ...	221
APPENDIX 13	SELECTED AT-MIO-64E-5 SPECIFICATIONS	224
APPENDIX 14	SELECTED PCDIO 216 SPECIFICATION	227
APPENDIX 15	STEP BY STEP SAMPLE COEFFICIENT SOLUTION	234
APPENDIX 16	NUMERICAL SOLUTION OF DATCOM GRAPHICAL DATA	237

LIST OF ILLUSTRATIONS

Fig	Description	Page
1.1	Antoinette's Apprenticeship Barrel	1
1.2	Link Trainer	2
1.3	Link C11 jet trainer	3
1.4	Comet IV FFS	5
1.5	Modern FFS	5
1.6	World War II bomber trainer	6
1.7	Early visual system airport model	7
1.8	Modern computer generated visual scene	8
1.9	Canadair Reconfigurable Engineering Flight Simulator (REFS) ..	9
2.1	The body axis system	18
2.2	Velocity in the body axis system	19
2.3	Stability axis system	20
2.4	Burt Rutan's asymmetrical Boomerang	24
2.5	Control surface deflection	27
2.6	Thrust vector point of application	29
2.7	Gravity in the body axis system	30
2.8	Ground kinematic model	32
2.9	Landing strut compression	34
2.10	Standard ICAO atmosphere	37
2.11	Simulated ICAO standard atmosphere	39
3.1	Flight model block diagram	41
3.2	Flight dynamic model implementation flow chart	42
3.3	Simplified EFIS display	55
3.4	Actual EFIS display	55
3.5	Map display	56
3.6	Strip chart recorder	57
4.1	Simulated cockpit section	59

4.2	Magnetic compass construction	60
4.3	The Earth's magnetic field	62
4.4	Schematic representation of a magnetic compass	66
4.5	Air speed indicator, cutaway view	68
4.6	Pitot tube and static port location	69
4.8	Beech Duchess indicated to calibrated airspeed conversion	71
4.9	Attitude indicator	72
4.10	Attitude indicator construction	73
4.11	Simulated attitude indicator construction	74
4.12	Exploded view of a typical altimeter	75
4.13	Simulated altimeter construction	77
4.14	Turn coordinator	78
4.15	Gyroscopic compass	81
4.16	Vertical speed indicator	81
4.17	Vertical speed indicator operation	82
5.1	Aircraft bearing to a Non Directional Beacon (NDB)	85
5.2	Aircraft tracking an NDB without compensating for cross wind	85
5.3	NDB station and ADF receiver	86
5.4	ADF operating principle	87
5.5	Typical ADF control panel	88
5.6	Relative bearing from aircraft to NDB beacon	89
5.7	ADF simulation flowchart	91
5.8	Course deviation indicator	91
5.9	Aircraft course deviation, approaching a VOR	92
5.10	VOR indication	93
5.11	VOR receiver block diagram	95
5.12	Bearing from VOR to aircraft	96
5.13	VOR simulation flowchart	97
5.14	CDI with glide slope indication	100
5.15	Localizer deviation indication	100

5.16	Glide slope deviation indication	101
5.17	Localizer signal pattern	102
5.18	Glide slope signal pattern	103
5.19	Typical ILS glide slope or localizer receiver	103
6.1	Brushless DC motor operation	108
6.2	Stepper motor micro stepping	111
6.3	Analog output board schematic	112
6.4	Brushless DC motor step size with 8 bit DAC resolution	112
6.5	Brushless DC motor optimization	113
6.6	Optimized brushless DC motor step size with 8 bit DAC resolution	114
6.7	PC game port pinout	118
7.1	Typical aircraft under consideration	122
7.2	Wing geometry	123
7.3	Vertical stabilizer geometry	126
7.4	Typical airfoil lift curve	127
7.5	Boeing 747-400 with winglets	130
7.6	Wing fuselage interference factor	131
7.7	Lift correction for a swept wing	131
7.8	Skin friction coefficient	132
7.9	Airfoil thickness location parameter	133
7.10	Leading edge suction parameter	143
7.11	Fuselage dimensions	136
7.12	Lift and pitching coefficient about the aerodynamic center	138
7.13	NACA 2412 Lift and pitching moment coefficients	138
7.14	Horizontal stabilizer downwash angle	140
7.15	Horizontal stabilizer and center of gravity location	143
7.16	Effect of taper ratio and flap span	145
7.17	Plain flap lift effectiveness	146
7.18	Plain flap lift correction factor	146
7.19	Effect of aspect ratio and flap chord ratio on three	

	dimensional flap effectiveness	147
7.20	Fuselage segmentation	149
7.21	Effect of fuselage slenderness ratio	150
7.22	Fuselage pitch due to angle of attack factor	151
7.23	Wing correction factor for pitch damping	153

LIST OF TABLES

Tab	Description	Page
2.1	Velocity, force, and moment notation in the body axis	17
2.2	Position and attitude in the fixed axis system	21
2.3	Standard atmosphere at sea level	36
2.4	Standard atmosphere at 11 000 meters	38
3.1	Control identification	52
3.2	Control changes	52
4.1	Airspeed indicator indicated airspeed vs angular position	71
4.2	Simple altimeter error limits	75
4.3	Vertical speed Indicator angular displacement	83
6.1	Instruments driven by brushless dc motors	109
6.2	Galvometer instruments	110
6.3	Potentiometer analog outputs	115
6.4	Digital Indicators	116
6.5	On / off switches	117
7.1	Stability and control coefficients	139

NOMENCLATURE

A	Aspect ratio
a_0	Speed of sound in standard conditions (340.3 m/s)
a	Standard atmosphere temperature lapse rate (-6.5 °C/1000m)
a_N	Aircraft northerly acceleration
a_E	Aircraft easterly acceleration
a_V	Aircraft vertical acceleration
B_h	Earth's tangential magnetic field potential
B_r	Earth's radial magnetic field potential
b	Wing span
C_D	Overall drag coefficient
$C_{D_{L_h}}$	Vertical stabilizer induced drag due to lift
$C_{D_{L_w}}$	Wing induced drag due to lift
C_{D0}	Aircraft drag coefficient at zero angle of attack
C_{D0_f}	Fuselage drag coefficient at zero angle of attack

C_{D0_h} Horizontal stabilizer drag coefficient at zero angle of attack

C_{D0_v} Vertical stabilizer drag coefficient at zero angle of attack

C_{D0_w} Wing drag coefficient at zero angle of attack

$C_{D\alpha}$ Aircraft drag coefficient due to angle of attack

$C_{D\alpha_h}$ Horizontal stabilizer drag coefficient due to angle of attack

$C_{D\alpha_w}$ Wing drag coefficient due to angle of attack

$C_{D\delta_e}$ Drag coefficient due to elevator deflection

C_{ff} Turbulent flat plate friction coefficient of the fuselage

C_{fw} Turbulent flat plate friction coefficient of the wing

C_g Landing gear strut compression

\dot{C}_g Rate of landing gear strut compression

C_L Aircraft lift coefficient

C_{Lq} Aircraft lift coefficient due to aircraft pitch rate

C_{Lq_h} Horizontal stabilizer lift coefficient due to aircraft pitch rate

C_{Lq_w} Wing lift coefficient due to aircraft pitch rate

C_{L_w}	Wing lift coefficient
$C_{L_{wf}}$	Wing and fuselage lift coefficient
$C_{L_{\alpha_{wf}}}$	Wing and fuselage lift curve slope
C_{L_0}	Aircraft lift coefficient at zero angle of attack
C_{L_α}	Aircraft lift coefficient due to aircraft angle of attack
$C_{L_{\alpha_h}}$	Horizontal stabilizer lift coefficient due to angle of attack
$C_{L_{\alpha_v}}$	Vertical stabilizer lift coefficient due to angle of attack
$C_{L_{\alpha_w}}$	Wing lift coefficient due to angle of attack
$C_{L_{\dot{\alpha}}}$	Aircraft lift coefficient to rate of change of aircraft angle of attack
$C_{L_{\delta_e}}$	Horizontal stabilizer lift coefficient due to elevator deflection
$C_{L_0_w}$	Wing zero angle of attack lift coefficient
$C_{L_{\alpha_h}}$	Horizontal stabilizer lift curve slope
$C_{L_{\alpha_v}}$	Vertical stabilizer lift curve slope
$C_{L_{\alpha_w}}$	Wing lift curve slope
$C_{L_{\delta_e}}$	Lift coefficient due to elevator deflection

C_l Overall rolling moment coefficient

C_{l0} Basic rolling moment coefficient

$C_{l\beta}$ Aircraft rolling moment coefficient due to side slip angle

C_{lp} Aircraft rolling moment coefficient due to roll rate

C_{lr} Aircraft rolling moment coefficient due to yaw rate

$C_{l\delta_a}$ Aircraft rolling moment coefficient due to aileron deflection

$C_{l\delta_r}$ Aircraft rolling moment coefficient due to rudder deflection

$(C_{l\delta})_{theory}$ Theoretical lift coefficient due to flap deflection

$$\frac{C_{l\delta}}{(C_{l\delta})_{theory}}$$
 Plain flap lift correction factor

C_m Aircraft pitching moment coefficient

C_{m_L} Aircraft pitching moment coefficient due to lift

$C_{m_{0L}}$ Aircraft pitching moment coefficient at zero lift

C_{m0} Aircraft pitching moment coefficient at zero angle of attack

C_{m0L} Aircraft pitching moment coefficient at zero lift

C_{m0L_f} Fuselage pitching moment coefficient at zero lift

C_{m0L_h} Horizontal stabilizer pitching moment coefficient at zero lift

C_{m0L_w} Wing pitching moment coefficient at zero lift

$C_{m\alpha}$ Aircraft pitching moment coefficient due to aircraft angle of attack

$C_{m\dot{\alpha}}$ Aircraft pitching moment coefficient due to rate of change of angle of attack

C_{mq_h} Horizontal stabilizer pitching moment coefficient due to pitch rate

C_{mq_w} Wing pitching moment coefficient due to pitch rate

C_{mq} Aircraft pitching moment coefficient due to pitch rate

$C_{m\delta_e}$ Aircraft pitching moment coefficient due to elevator deflection

C_n Overall yawing moment coefficient

C_{n0} Aircraft basic yawing moment coefficient

C_{np} Aircraft yawing moment coefficient due to roll rate

C_{nr} Aircraft yawing moment coefficient due to yaw rate

$C_{n\beta}$ Aircraft Yawing moment coefficient due to side slip angle

$C_{n\delta_a}$ Yawing moment coefficient due to aileron deflection

$C_{n\delta_r}$ Aircraft yawing moment coefficient due to rudder deflection

- C_Y Aircraft overall side force coefficient
- C_{Y_p} Aircraft side force coefficient due to roll rate
- C_{Y_r} Aircraft side force coefficient due to yaw rate
- C_{Y0} Aircraft basic side force coefficient
- $C_{Y\beta}$ Aircraft side force coefficient due to side slip angle
- C_{Yp} Aircraft side force coefficient due to roll rate
- C_{Yr} Side force coefficient due to yaw rate
- $C_{Y\delta_r}$ Side force coefficient due to rudder deflection
- c Airfoil aerodynamic chord
- \bar{c} Mean aerodynamic chord
- \bar{c}_{exp} Mean aerodynamic chord of the exposed wing
- c_f Chord at the fuselage
- c_f/c Flap chord ratio
- c_{l_α} Airfoil lift curve slope
- $c_{l_{\alpha_h}}$ Horizontal stabilizer airfoil lift curve slope

$c_{l_{\alpha_w}}$	Wing airfoil lift curve slope
c_m	Airfoil pitching moment coefficient
c_{m_0}	Airfoil pitching moment coefficient at zero angle of attack
c_r	Root chord
c_t	Tip chord
d_f	Fuselage diameter
e	Span efficiency factor
\bar{F}_b	Net force acting on the aircraft in the body axis system
g	Gravitational acceleration
h	Maximum fuselage diameter
h_h	Horizontal stabilizer height above the wing
I_x	Aircraft moment of inertia about the X body axis
I_{xy}	Aircraft product of inertia about the Z body axis
I_{xz}	Aircraft product of inertia about the Y body axis
I_y	Aircraft moment of inertia about the Y body axis

I_{yz}	Aircraft product of inertia about the X body axis
I_z	Aircraft moment of inertia about the Z body axis
i_h	Horizontal stabilizer angle of incidence relative to the fuselage
i_w	Wing angle of incidence relative to the fuselage
\hat{i}	Unit vector along X body axis
\hat{j}	Unit vector along Y body axis
K_A	Factor for calculating horizontal stabilizer downwash based on the wing aspect ratio
K_b	Flap span factor
$K_{b_{main}}$	Main gear damping coefficient
$K_{b_{nose}}$	Nose gear damping coefficient
K_h	Factor for calculating horizontal stabilizer downwash based on its height
$K_{s_{main}}$	Main gear spring coefficient
$K_{s_{nose}}$	Nose gear spring coefficient
K_{TC}	Turn coordinator spring constant
K_w	Wing pitch damping correction factor

- K_{wf} Wing - fuselage lift interference factor
- K_λ Factor for calculating horizontal stabilizer downwash based on the wing taper ratio
- k Actual to ideal airfoil lift curve slope ratio
- k' Correction factor for flap nonlinearities at large deflections
- \hat{k} Unit vector along Z body axis
- $k_2 - k_1$ Effect of fuselage slenderness ratio on the fuselage pitching coefficient
- L Net rolling moment acting on the aircraft about the X body axis
- L' Airfoil thickness location parameter
- L_1 Magnetic compass moment arm for moments due to magnetic forces
- L_2 Distance from magnetic compass cg to its pivot
- L_A Aerodynamic rolling moment acting on the aircraft about the X body axis
- L_g Length of an extended landing gear strut
- l_{LER} Leading edge radius
- l_{cyl} Length of fuselage cylindrical portion
- l_f Overall fuselage length

- l_h Horizontal distance from the wing's 1/4 MAC position to the horizontal stabilizer's 1/4 MAC position.
- l_{nose} Length of fuselage nose portion
- l_{tail} Length of fuselage tail portion
- l_v Distance from the center of gravity to the vertical stabilizer's 1/4 MAC
- M Net pitching moment acting on the aircraft about the Y body axis
- M_A Aerodynamic pitching moment acting on the aircraft about the Y body axis
- M_{brk} Net braking pitch moment about the Y body axis
- M_T Pitching moment body axis due to both engines' thrust
- m Aircraft mass
- m_i Aircraft mass increment at position \bar{x}_i in the body axis system
- N Net yawing moment acting on the aircraft about the Z body axis
- N_A Aerodynamic yawing moment in the body axis.
- N_T Net yawing moment in the body axis due to both engines' thrust
- P Pressure
- P_{sl} Sea level pressure

P_{sta} Static pressure

P_{tot} Total pressure

\bar{P} Aircraft's angular momentum about its center of mass

P_0 Standard pressure (101.3 kPa)

p Aircraft roll rate about the X body axis

\dot{p} Aircraft roll acceleration about the X body axis

Q Dynamic pressure

q Aircraft pitch rate about the Y body axis

\bar{q} Free stream dynamic pressure

\bar{q}_h Dynamic pressure acting on the horizontal stabilizer

\dot{q} Aircraft pitch acceleration about the Y body axis

R Leading edge suction parameter

R Gas constant (29.26 m³/K)

R_{cg} Ground model turning radius at the center of gravity

R_{1_f} Reynolds number based on the fuselage length

$R_{l_{LER}}$ Reynolds number based on leading edge radius

R_{main} Ground model turning radius at main landing gear

R_{Nfus} Fuselage Reynolds number

R_{Nw} Wing Reynolds number

R_{wff} Wing - fuselage drag interference factor

R_{LS} Lifting surface correction factor

r Aircraft yaw rate about the Z body axis

\dot{r} Aircraft yaw acceleration about the Z body axis

\bar{r}_i Position of mass increment m_i in the body axis system

r_1 Fuselage radius at the 1/4 chord point of the vertical stabilizer

S Projected wing area

S_{cg} Center of gravity speed for ground model calculation

S_h Projected horizontal stabilizer area

S_v Projected vertical stabilizer area

S_{wet} Total exposed wing area above and below

S_{wet_f} Total exposed fuselage area

T Temperature

\bar{T} Torque acting of the aircraft about its center of mass

T_0 Standard temperature (15.16 °C or 288.16 °K)

T_L Left engine thrust

T_R Right engine thrust

T_{sl} Sea level temperature

T_T Total engine thrust

t Time

t/c Thickness to chord ratio

U_0 Reference flight speed

u Aircraft velocity along the X body axis

\dot{u} Aircraft acceleration along the X body axis

V_{cal} Calibrated air speed

\bar{V}_b Aircraft velocity in body axis system

\bar{V}_h	Horizontal tail volume coefficient
V_{true}	True air speed
V_{XF}	Aircraft velocity along X fixed axis
V_{YF}	Aircraft velocity along Y fixed axis
v	Aircraft velocity along the Y body axis
\dot{v}	Aircraft acceleration along the Y body axis
w	Aircraft velocity along the Z body axis
\dot{w}	Aircraft acceleration along the Z body axis
w_f	Average diameter of a fuselage segment
$w_{f_{max}}$	Maximum fuselage width
X	Net force acting on the aircraft along the X body axis
X_{brk}	Net braking force
X_{cp}	Location of the airfoil's center of pressure
X_{ac}	Location of the airfoil's aerodynamic center
X_{ac_h}	Location of the horizontal stabilizer's aerodynamic center

X_{cg}	Location of the aircraft's center of pressure
X_g	Force due to gravity along the X body axis
X_{main}	Location of main gear along X body axis
X_{nose}	Location of nose gear along X body axis
X_{ref}	Reference position
X_T	Net force in body axis due to both engines' thrust along the X body axis
X_w	Horizontal distance from the wing's 1/4 MAC position to the center of gravity
x	Aircraft X coordinate in the fixed axis system
\dot{x}	Aircraft X velocity in the fixed axis system
\bar{x}_{ac}	Reference position nondimensionalized to mean aerodynamic chord
x_m	Horizontal distance from the nose to the center of gravity
\bar{x}_{ref}	Reference position nondimensionalized to mean aerodynamic chord
Y	Net force acting on the aircraft along the Y body axis
Y_A	Aerodynamic side force along the Y body axis
Y_E	Location of engine thrust vectors on either side in body axis system

Y_g Side force due to gravity along the Y body axis

Y_{main} Location of main gear along Y body axis

Y Aircraft Y coordinate in the fixed axis system

Y_{mac} Location of mean aerodynamic chord

\dot{Y} Aircraft Y velocity in the fixed axis system

Z Net force acting on the aircraft along the Z body axis

Z_E Location of engine thrust vectors in body axis system

Z_g Force due to gravity along the Z body axis

Z_{gear} Net landing gear vertical force in body axis system\

Z_{g_l} Left main gear vertical force in body axis system

Z_{g_n} Nose gear vertical force in body axis system

Z_{g_r} Right main gear vertical force in body axis system

$Z_{g_{top}}$ Z coordinate of the top of a landing gear strut in the fixed axis system

$Z_{g_{ext}}$ Z coordinate of the bottom of an extended landing gear strut in the fixed axis system

Z_g Force due to gravity along the Z body axis

- z Aircraft Z coordinate in the fixed axis system
- \dot{z} Aircraft Z velocity in the fixed axis system
- z_v Vertical location of the vertical stabilizer mean aerodynamic chord
- α Aircraft angle of attack
- $\frac{(\alpha_\delta)_{c_L}}{(\alpha_\delta)_{c_L}}$ Effect of aspect ratio and flap chord ratio on flap effectiveness
- α_h Horizontal stabilizer angle of attack
- $\alpha_{h_{\delta e}}$ Horizontal stabilizer angle of attack due to elevator deflection
- α_w Wing angle of attack
- α_{0L} Aircraft zero lift angle of attack
- α_{0L_h} Horizontal stabilizer zero lift angle of attack
- α_{0L_w} Wing zero lift angle of attack
- α_{0l} Airfoil zero lift angle of attack
- β Aircraft sideslip angle
- Δt Time increment
- Δx Length of a fuselage segment

δ_a	Aileron deflection angle
δ_e	Elevator deflection angle
δ_r	Rudder deflection angle
ε_h	Horizontal stabilizer downwash angle
ε_{0h}	Horizontal stabilizer downwash angle at zero aircraft angle of attack
ϵ	Nose gear steering angle
η_h	horizontal stabilizer to wing aerodynamic pressure ratio
η_v	vertical stabilizer to wing aerodynamic pressure ratio
λ	Taper ratio
λ_{exp}	Taper ratio of the exposed wing
Λ_{le}	Leading edge sweep angle
Λ_{mid}	Mid-chord sweep angle
$\Lambda_{1/4}$	Quarter chord sweep angle
μ_0	Standard absolute viscosity (1.789×10^{-3} Ns/m ²)
ν_0	Standard kinematic viscosity (1.460×10^{-3} m ² /s)

ρ	Density
ρ_0	Standard density (1.225 kg/m ³)
ϕ	Aircraft roll angle in the fixed axis system
ϕ_{TC_S}	Turn coordinator slip indicator deflection
ϕ_{TC_T}	Turn coordinator turn indicator deflection
$\dot{\phi}$	Aircraft roll rate in the fixed axis system
θ	Aircraft pitch angle in the fixed axis system
θ_{ADF}	Relative bearing from aircraft to NDB station
$\dot{\theta}$	Aircraft pitch rate in the fixed axis system
T	Angle between main gear and cg turning radius vector
ψ	Aircraft yaw angle in the fixed axis system
$\dot{\psi}$	Aircraft yaw rate in the fixed axis system
$\bar{\omega}_b$	Aircraft angular velocity in the body axis system
ω_{TC_N}	Turn coordinator gyro frame angular velocity

1.0 INTRODUCTION

1.1 Historical Background of Flight Simulation.

Before the first flight of the Wright Brothers in 1908, it was generally believed that a powered aircraft would be a stable and simple device to control, requiring few piloting skills to be developed. [1.1] This was soon found to be an incorrect assumption as early aircraft often had unique and fatal characteristics. The need to effectively prepare pilots for the task of flying lead to the idea that building devices which could simulate the flight characteristics of an aircraft would be invaluable as a flight training tool. This was the conception of flight simulation.

1.1.1 Early Flight Simulation Systems

The emphasis in early development remained the control and motion of the simulator in response to disturbances provided by the instructor. Before World War I, the Antoinette "Apprenticeship Barrel" shown in Figure 1.1 had the student sitting in a half barrel, the attitude of which he attempted to maintain through the 'flight controls' as two instructors created disturbances by moving the platform on which it balanced. The rough dynamic models and systems available at the time made the effectiveness of this and similar trainers, debatable. [1.2]



Figure 1.1 Antoinette's Apprenticeship Barrel. [1.2]

The Link trainer was developed from 1927 to 1929 and achieved the most reasonable motion simulation for its time. Link, though his experience in the piano and organ business, built the trainer in which the motion was created by pneumatic bellows similar to those in organs. Driven by a pump, the bellows were activated by valves linked to the stick and rudder pedals in the cockpit. Through extensive trial and error testing, Link manually tuned the response until it had the right feel. Light turbulence was achieved with a second pump which fed a repeated sequence of disturbances to the bellows. By the early 1930s the Link trainer had evolved to the point that it was fitted with instruments and was being used to train pilots in the US Army, shown in Figure 1.2. [1.2] Development of the Link trainer continued and it was still being used to train pilots in World War II. [1.3]

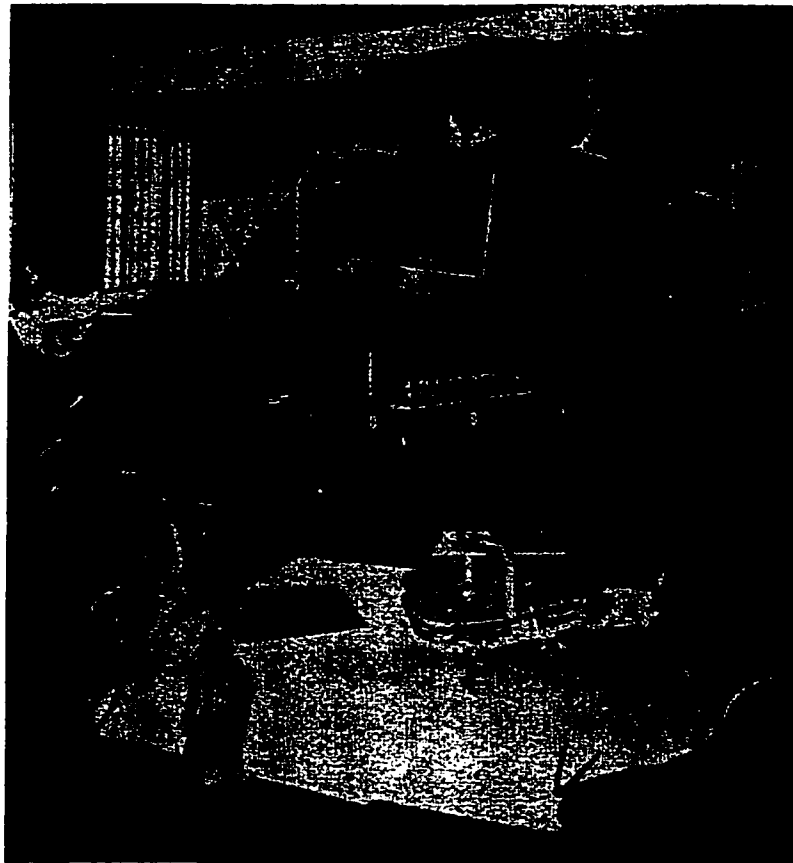


Figure 1.2 Link Trainer [1.2]

By the early 1920s, a reasonable mathematical model of flight dynamics was available, but the pace at which flight simulators could develop was dictated by the technology available to implement the mathematical models.

1.1.2 Introduction of the Analog Computer

Analog computers brought flight simulation into the electronic era. High gain amplifiers with resistive feedback provided acceptable values of coefficients, and amplifiers with capacitive feedback provided an integrating action. By that means, a circuit could be configured to model the differential equations of

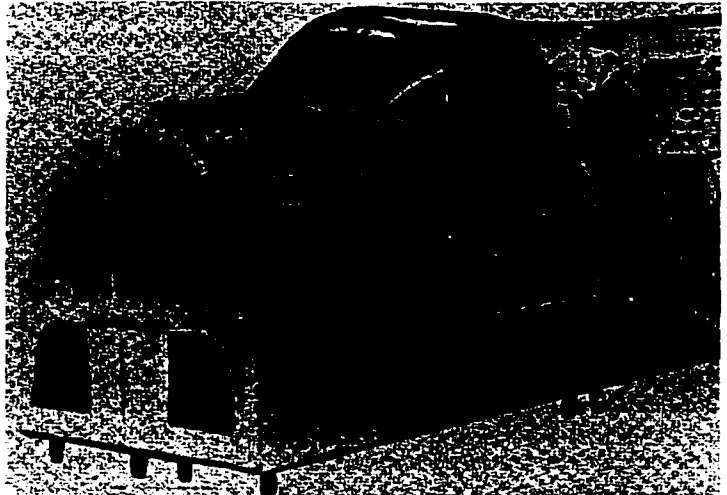


Figure 1.3 Link C11 jet trainer [1.2]
an aircraft's dynamic model. The introduction of the analog computer marked the point at which electronic systems would begin to replace mechanical methods of simulating the dynamic model. By the late 1940's the Link company had incorporated an analog computer into their C-11 jet trainer shown in Figure 1.3. [1.2] Another significant change which occurred at around the same time was the shift in emphasis from trying to simulate the motion and control of an aircraft to realistically simulating the flight characteristics and instrumentation on a fixed base system without any motion.

1.1.3 Introduction of the Digital Computer

The next significant development in flight simulation technology was the introduction of the digital computer. In 1960 the Universal Digital Operational Flight Trainer, UDOFT, was completed at the University of Pennsylvania and was the first digital computer to simulate the aircraft's equations of motion. Link also developed a special purpose digital computer, the Link Mark I, specifically for real time simulation. [1.2]

1.1.4 Full Flight Simulators and Flight Training Devices

Up to this point, simulators were primarily procedure trainers used to teach the procedures for flying and navigation and lacked the true feel of flight. It was now becoming technically feasible to implement a reasonable motion system on the simulators. Flight simulation now began to develop two distinct objectives. The first was as a procedure trainer to teach navigation and instrument flying procedures. Since navigation and instrument procedures training do not require a sophisticated motion system, visual system or accurate flight model, a simple fixed base simulator without these features would suffice. This led to the category of flight simulators known today as Flight Training Devices, (FTD). The second objective, was to teach actual flying techniques. This is done most effectively by providing the student with the most realistic environment possible with motion, sound, visual, and instruments simulated as accurately as possible so that he may develop a feel for the actual aircraft being simulated. This led to the category of flight simulators known today as Full Flight Simulators (FFS). Modern flight simulator manufacturers often build FTDs with the option of a future upgrade to an FFS with the addition of a motion, visual, and enhanced flight model.

Another line of development which emerged in flight simulation was its use as a research tool in aircraft design, stability and control. This aspect will be discussed later.

1.1.5 Simulator Motion Systems

Motion is a key element of a full flight simulator. Early FFSs had few or even one degree of freedom. The Comet IV simulator shown in Figure 1.4 had a single degree of freedom in the pitch axis. Modern FFSs such as those developed at CAE Electronics, shown in Figure 1.5, employ six hydraulic actuators to achieve full six degree of freedom motion. Today these systems have developed to the point where they offer extremely realistic motion cues to the pilot, including such effects as turbulence, runway roughness, engine vibrations, and braking.

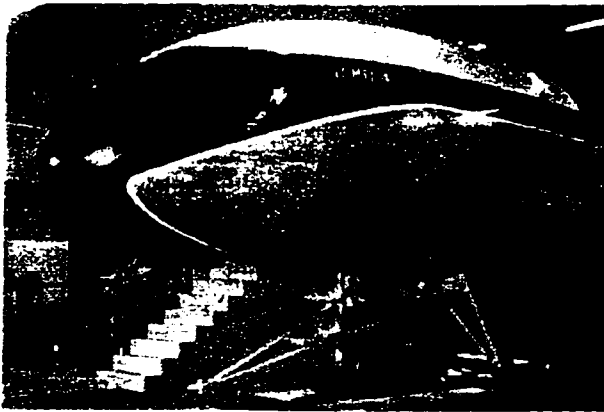


Figure 1.4 Comet IV FFS [1.2]



Figure 1.5 Modern FFS [1.4]

1.1.6 Simulator Visual Systems

A realistic visual system is another key component of a full flight simulator. By offering key visual cues, it provides the final sensory input to the pilot to create a truly realistic training environment. Considerable computing power is necessary to generate digital pictures in real time, and so this technique developed at a slower pace than the digital computation of the flight models.

Early visual systems were based on actual physical models, or pictures. During World War II, the German bomber crews trained on the apparatus shown in Figure 1.6. A continuous picture of a ground scene was rolled under the trainee who looked through the sight from a similar position to that in the actual bomber.

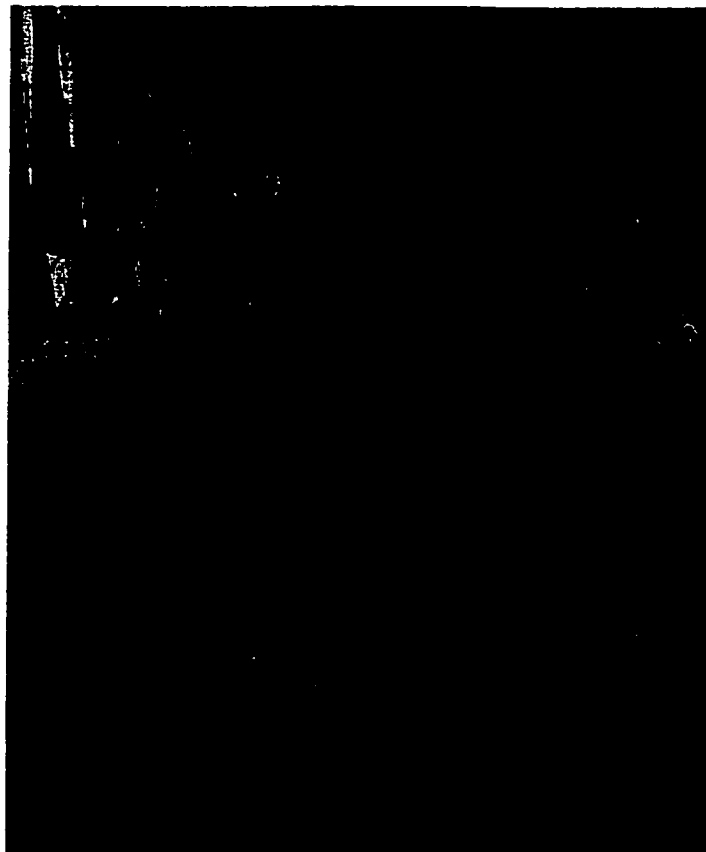


Figure 1.6 World War II bomber trainer. [1.5]

In later simulators, a miniature model of a scene around an airport would be built and a video camera would travel over it based on the aircraft's position as calculated by the flight model. An example of such a system was used on a Concorde simulator as shown in Figure 1.7. There were a number of significant disadvantages of this system, which eventually lead to its abandonment in pursuit of computer generated images. Firstly, the simulation was physically large, and the area in which the aircraft could fly was limited to the size of the scene, and could not be easily changed. Secondly, due to the irregularity of buildings, mountains, and other obstacles in the scene it was difficult to prevent and protect the camera from impacting the model when the aircraft crashes. This would lead to costly repairs to the model and camera system.

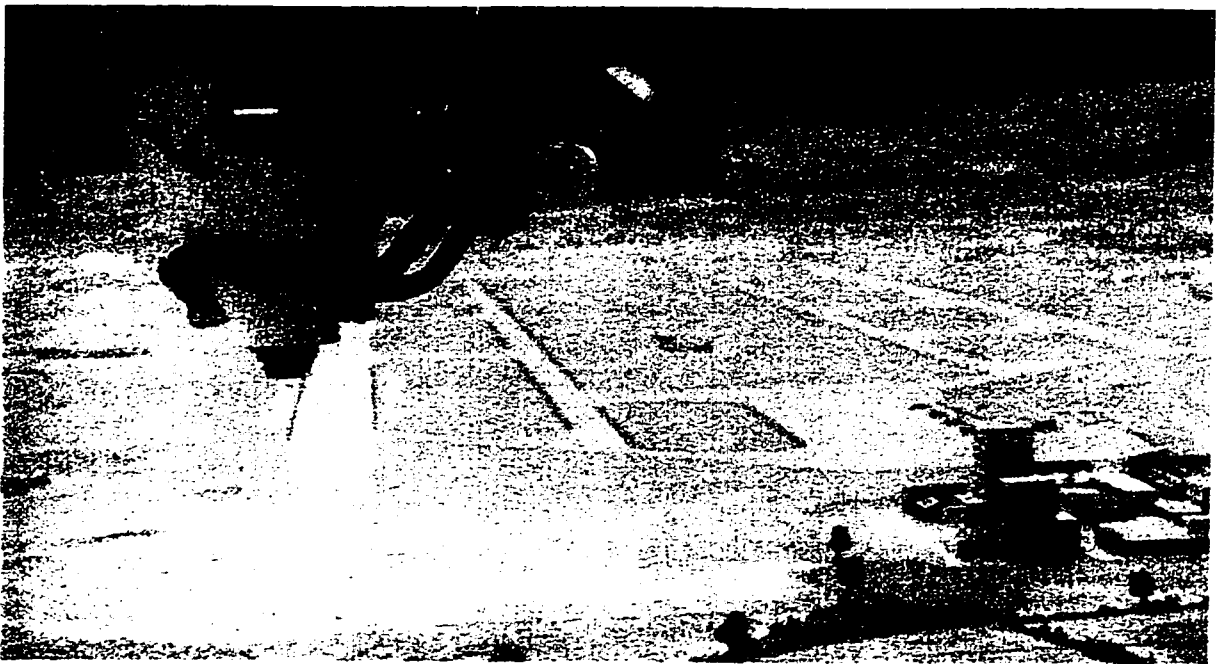


Figure 1.7 Early visual system airport model. [1.6]

The first digitally generated visual scenes systems offered a patterned 'ground plane' to represent the ground and sky. As systems improved, night scenes were introduced. Night scenes were developed first due to their simplicity, requiring only projections of points of light rather than the detailed three dimensional features of buildings and terrain. For day scenes, only simple block buildings were possible initially, but as computing power increased, detailed features including ground texturing, clouds, water, other traffic, and reduced visibility effects due to weather conditions became possible. Figure 1.8 show a typical visual scene available today.

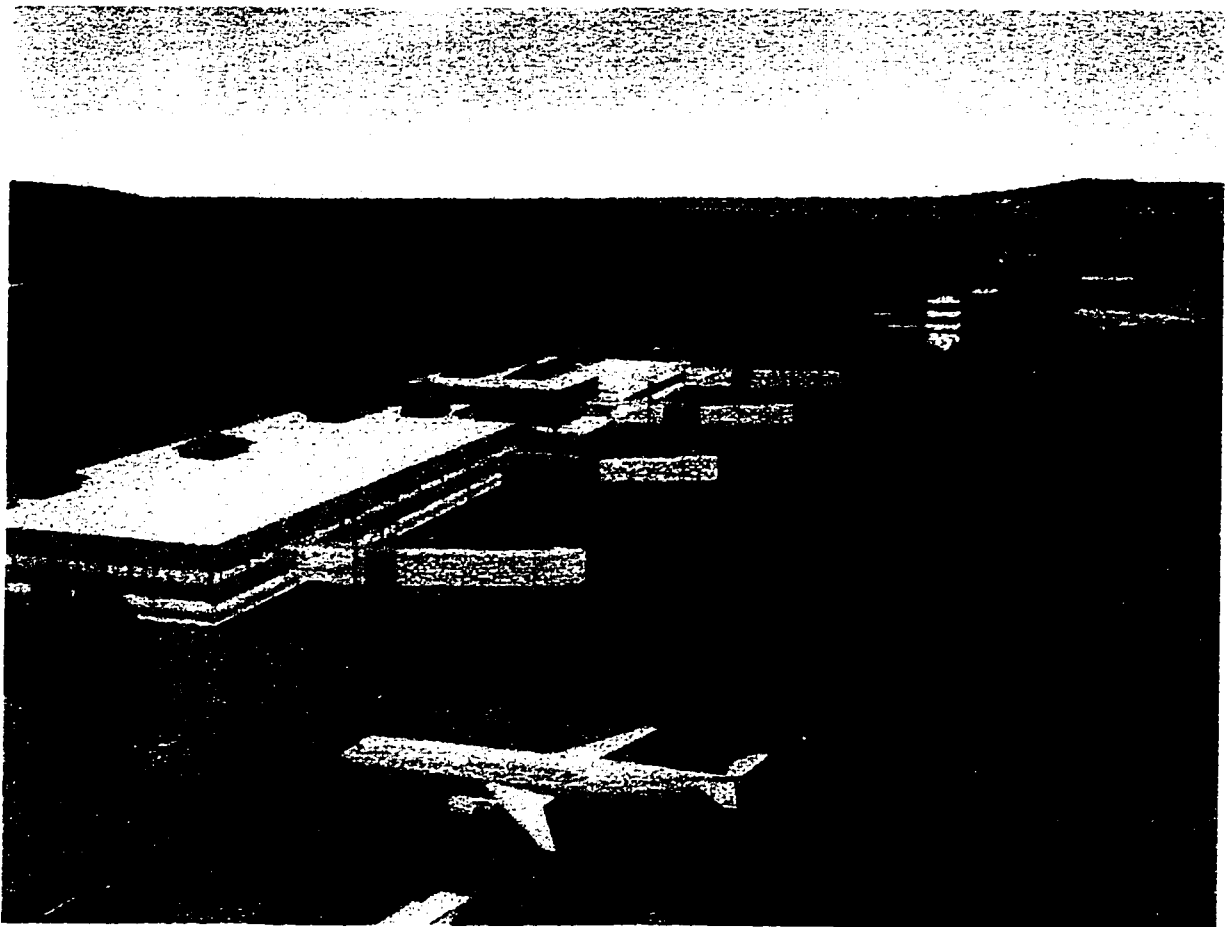


Figure 1.8 Modern computer generated visual scene. [1.7]

1.2 Flight Simulators as Research and Design Tools

Throughout the development of flight simulation, the emphasis has been on training pilots. Simulators used for studies into flight stability and control have generally been adapted from commercial flight simulators, or are custom built for a specific task. Today sufficient analytical tools exist to predict the stability and control characteristics of an aircraft design with a high level of accuracy before the first prototype is built. With this a research simulator can become an invaluable tool when studying the handling characteristics and when designing the aircraft's control systems. Depending on the specific requirements, such a simulator could take on any form from an FFS, an FTD or a flight model running on a stand alone computer. An example of such a simulator is shown in Figure 1.9, the Reconfigurable Engineering Flight Simulator REFS at Canadair. used to study the stability and control characteristics of such aircraft as the CL65 Regional Jet and the Global Express.



Figure 1.9 Canadair Reconfigurable Engineering Flight Simulator (REFS) [1.8]

Until the early 1980s, a powerful digital computer was required to run a reasonable real time flight dynamic model. It was not until the rapid growth of the microprocessor industry, which lead to inexpensive but powerful computers that the once impossible task of building a low cost flight simulator became possible. This is what lead to the development of the Concordia Flight Simulator.

1.3 The Original Concordia Flight Simulator System

In the early 1980s, the Fluid Power Laboratory at Concordia University embarked on a joint project with CAE Electronics to study the feasibility of building a low cost light twin engine flight simulator.

The result was an FTD built from the cockpit of an actual Beech Dutchess 76 aircraft. All the instruments and controls in the cockpit were retrofitted such that they could be realistically driven or read by the flight model computer. Most of the flight and navigational instruments were driven by either brushless dc or stepper motors. The variable controls were fitted with potentiometers to provide a voltage proportional to their position. Discrete digital input and output lines are used to drive simple on / off displays such as lamps, and read on / off devices such as switches and circuit breakers. The computer chosen to run the simulation was an Intel 8088 system with 128 kb of memory, an eight bit bus, and four RS-232 serial communication ports to communicate with external devices such a an instructor station or visual system. All interface cards used to read and drive the simulated instruments and controls were custom built. The flight model program was coded in Fortran specifically to model the Dutchess, and the interface routines, which communicate with the instruments and controls, were coded in assembly language.

Although unquestionably successful in demonstrating the feasibility of building a low cost FTD, CAE chose not to pursue this project further and the simulator has remained at Concordia where it has since been continuously improved over the years. Later enhancements would include the development of a simple computer generated visual system which would project the forward view from the aircraft on a screen mounted in front of the windshield. A motion system was considered but studies showed that an affordable limited capability motion systems would actually detract from the simulator's value. A decision was made not to pursue a motion system. [1.9]

1.4 Thesis Objectives

Compared to the computers of the early 1980s, modern commercially available computers are significantly more powerful and versatile, smaller in size and available at a lower cost. It became apparent that upgrading the existing computer to modern standards would allow significant enhancements to the capabilities of Concordia's flight simulator. A decision was made to upgrade the simulator's hardware and to rewrite the simulation code. The objective was to create a useful academic tool for teaching flight dynamic and simulation topics, and to provide an effective tool for research into topics of aircraft dynamics, stability, control, as well as studies into human factors aspects of flying. This thesis documents the development of the simulation program, and details of the development of the flight model and the supporting systems, which are comprised of the atmospheric, ground, flight and navigational instruments models. It is also the intention of the author that this thesis offer the necessary insight to the simulation model, and as such provide future students with the necessary information to continue in this field.

1.4.1 Hardware Upgrade

To be useful as a teaching and research tool, the simulation code should not be constrained to run in the laboratory with the simulated cockpit. It should be portable, able to run on a computer without the cockpit, and be able to use whatever hardware is connected to it to run the simulation. If achieved, this would allow the simulation to be run in a classroom environment on a portable computer with a projector to display the results on a screen. This would also permit the simulation to be run on most personal computers, allowing students to experiment with the system on their own time, and to undertake projects such as the investigation of autopilot control systems on an individual basis. From these basic objectives, it became clear that the simulation should be written to run in the Intel 80X86 PC family of computers. In its most common form, the PC computer can be used with a joystick, a mouse, and a keyboard. These devices offer sufficient options to reasonably control a flight simulation. The joystick could be considered analogous to the aircraft's control column, a forward motion of the mouse could be used to advance the simulated engine throttles, and the keyboard may be used to invoke specific commands such as tuning navigational beacons or freezing the simulation. If the simulation is to be run on a stand alone PC without the cockpit and its instruments, then the program should be able display a graphical representation of the flight and navigational instruments on the computers display.

Selecting the 80X86 family of computers would also ensure future compatibility as this line of computers develops since they would always be downward compatible.

When the simulation is run on the computer which drives the simulated cockpit, it must be equipped with the necessary interface hardware to send and read the necessary signals to and from the simulated instruments and controls. Unfortunately the interface cards which were built for the original system

are not compatible with the new computer, and commercially available interface cards had to be selected and installed. The instrumentation and controls in the cockpit did not need to be modified. Over the years they have performed reliably and their technology has not advanced at the same rate as that of computers. The systems used in the instruments and controls are still representative of technology used in modern commercial flight simulators.

1.4.2 Flight Simulator as a Teaching Aid

The aerospace industry is rapidly growing. As a result, there is an increasing demand from industry for qualified personnel, and in addition, the interest in this field has also increased at the university level. In response, academic aerospace programs are growing and adapting to meet these demands. The simulation has been written to act as a teaching aid, with the ability to completely define the stability and control coefficients which describe an aircraft's flight characteristics. The effect of modifying these coefficients and their effects on aircraft stability can be demonstrated in real time. To support these demonstrations it is necessary to monitor various variables, such as speed and angle of attack, during the simulation. The simulation has been written to allow the simultaneous real time graphical display of up to eight variables during the execution of the simulation on the computer screen. This takes the form of a strip chart recorder. It is not reasonable to expect anyone to be sufficiently skilled to fly the simulation accurately and repeatably to demonstrate flight characteristics. To demonstrate predefined repeatable tests, provisions are included to set up the aircraft in a specific flight configuration and then, using a predefined file, to execute precise repeatable flight commands to demonstrate the effects of these tests.

1.4.3 Flight Simulator as a Research Tool

The research applications are an extension of the teaching applications. The flight model is a full six degree of freedom model running in real time, and it is written in C, a high level language familiar to most students with programming experience. This will allow the simulation to serve as a test bed for research into aircraft stability and control, as well as being easily modified for research into areas including automatic flight control and unusual attitudes such as stalls and spins. The ability to store simulation variables is included for logging. The is saved in a format that is easily imported to other programs for graphing and analysis.

Another equally important area in the aerospace industry is the study of human factors. Flying an aircraft requires not only the skill, but also the ability to assimilate large amounts of presented via the instruments about aircraft and engine performance, navigation, and communication. This information is necessary to make intelligent safe decisions. Modern aircraft manufacturers have invested large resources to determine the most efficient format in which to present information to the pilot. To make the simulation a useful tool in this area, it should be capable of providing information to the pilot from all the systems on the actual aircraft. This project begins a move in this direction by implementing, not only the flight model, but the flight instruments and navigation systems as well. The standard flight instruments and navigational systems are described both in their operation and their simulation. The presentation options discussed above would be equally applicable to monitoring a pilot's response to the available information.

1.4.4 Estimation of Stability and Control Coefficient

In using the simulation software, students will undoubtedly develop an appreciation of how the stability and control coefficients affect the performance of the aircraft. To offer the students an insight to how the configuration of an aircraft determines its stability and control coefficients, an established method developed by the US Air Force, DATCOM, will be implemented in a computer program. This program will allow the geometry of a simple subsonic aircraft to be modified, and the solution of the longitudinal coefficients to be presented in a step by step presentation of all the calculations and graphs used.

1.4.5 Summary of Objectives

In summary, the availability of higher performance computers has offered the opportunity to upgrade Concordia's flight simulator. Taking advantage of the increased computing power will help achieve the following goals in the new program.

1. Run on a standard 80X86 PC computer,
2. Be capable of running with a full simulated cockpit or limited input devices such as a mouse and keyboard, or a control file,
3. Simulate a six degree of freedom flight model including a ground model, and instrument simulation,
4. Offer display options to represent the flight and navigation instruments, monitor simulation variables in real time, or track the aircraft's geographical position,
5. Offer the capability to store simulation variables to a file for later analysis,
6. Be written in C to allow for modifications by students for future research and projects,
7. Present the development of the simulation in this thesis as a supporting document for students and professors using this simulation.

To complement the simulation, develop a program to implement an established method to estimate an aircraft's stability and control coefficients, providing a step by step display of the solution.

2.0 AIRCRAFT DYNAMIC MODELLING

In this chapter, the development of a six degree of freedom dynamic model for the aircraft is presented. It is based on work by Connelly [2.1] as presented by Nelson [2.2] and Roskam [2.3].

2.1 Coordinate Systems

To develop these equations, the axis systems being used must be clearly defined. When describing the flight of an aircraft, three axis systems are used: body, stability, and fixed axis.

2.1.1 The Body Axis System

Figure 2.1 shows that the body axis system's origin is fixed to the aircraft centre of gravity, with the X axis along the centre line of the aircraft, the Y axis to the right, and the Z axis pointing downwards. The quantities associated with each axis are tabulated in Table 2.1.

	Roll Axis X_b	Pitch Axis Y_b	Yaw Axis Z_b
Angular rates	p	q	r
Velocity	u	v	w
Force	X	Y	Z
Moment	L	M	N
Moment of Inertia	I_x	I_y	I_z
Product of Inertia	I_{yz}	I_{xz}	I_{xy}

Table 2.1 Velocity, force, and moment notation in the body axis.

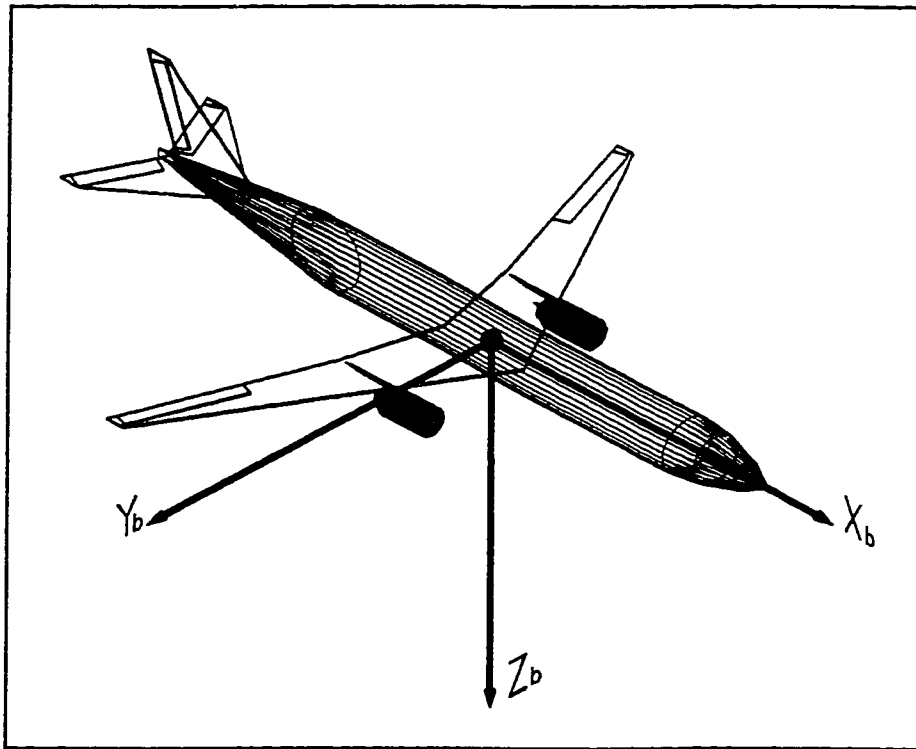


Figure 2.1 The body axis system .

From the velocities in the body axis system, the angle of attack (α) of the aircraft may be defined as shown in Figure 2.2 and Equation 2.1:

$$\alpha = \tan^{-1} \frac{w}{u} \quad (2.1)$$

If the aircraft has a lateral component of the velocity (v), then it is said to be sideslipping. This is described by a sideslip angle (β) defined as shown in figure 2.2 defined by Equation 2.2:

$$\beta = \tan^{-1} \frac{v}{\sqrt{u^2 + w^2}} \quad (2.2)$$

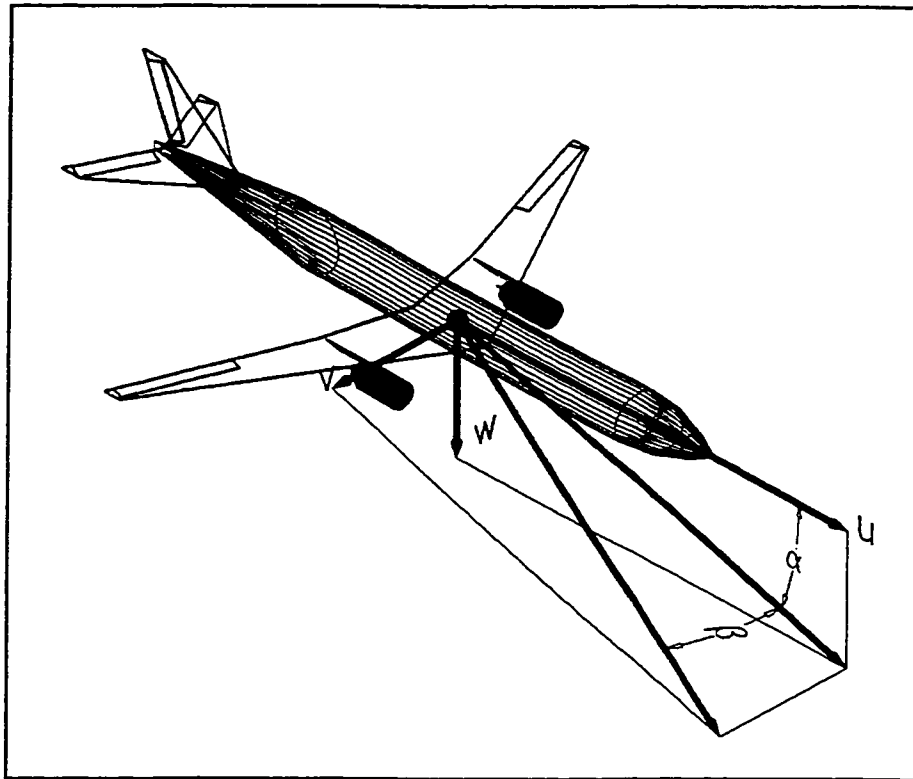


Figure 2.2 Velocity in the body axis system.

As the aircraft flies and rotates, the body axis system remains fixed to the aircraft, rotating with it. It is within this coordinate system that the equations of motion shall be derived and applied.

2.1.2 The Stability Axis System

It will be seen later that the direction of the forces acting on an aircraft are defined relative to its velocity. Since an aircraft may be flying at some angle of attack, another coordinate system, called the stability axis system, is defined in which the aerodynamic forces and moments are calculated. Again, like the body axis system, the origin of the stability axis system is fixed to the aircraft's centre of gravity but with the X axis aligned with the longitudinal component of the velocity in

the X-Z plane of the body axis system as shown in Figure 2.3. To convert a point from the body axis to the stability axis, it must be pitched in the negative direction by the angle of attack. The stability axis system will be the system in which the aerodynamic forces and moments are calculated. To apply them to the equations of motion, they will be transformed into the body axis system.

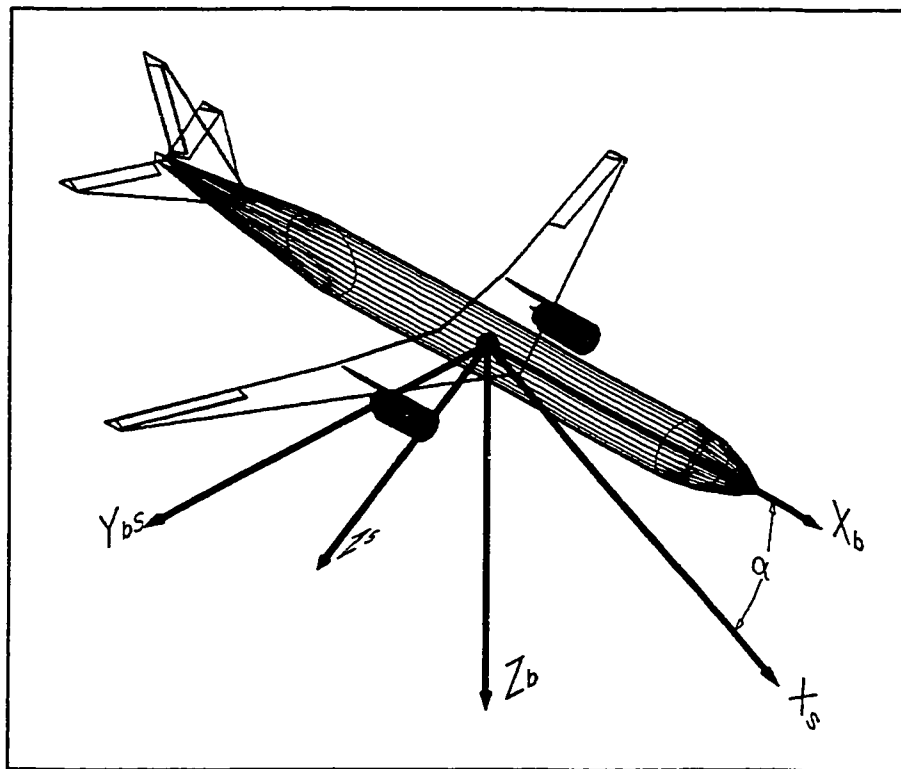


Figure 2.3 Stability axis system.

2.1.3 The Fixed Axis System

The fixed axis system is the coordinate system in which the aircraft flies. It is fixed to the earth and is assumed to be an inertial reference frame. The position and orientation of an aircraft in the fixed axis system is given in Table 2.2. Neglecting the motion of the earth, the fixed axis system is assumed to be an

inertial reference frame. It is in this coordinate system that the position and orientation of the aircraft is calculated. The altitude (z) is equal to zero at sea level.

West to East	x
North to South	y
Altitude (+ down)	z
Roll	ϕ
Pitch	θ
Yaw	ψ

Table 2.2 Position and attitude in the fixed axis system.

2.1.4 Euler Angles

The Euler angles between the different coordinate systems define the orientation of one coordinate system within another. Given the Euler angles in Table 2.2 describing the orientation of the body axis system in the fixed axis system, the transformation from the body fixed axis system to the fixed axis system is achieved with Equations 2.3 and 2.4.

$$\begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \\ \frac{dz}{dt} \end{bmatrix} = \begin{bmatrix} C\theta C\psi & S\phi S\theta C\psi - C\phi S\psi & C\phi S\theta C\psi + S\phi S\psi \\ C\theta S\psi & S\phi S\theta S\psi + C\phi C\psi & C\phi S\theta S\psi - S\phi C\psi \\ -S\theta & S\phi C\theta & C\phi C\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.3)$$

$$\begin{bmatrix} \frac{d\phi}{dt} \\ \frac{d\theta}{dt} \\ \frac{d\psi}{dt} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi\sec\theta & \cos\phi\sec\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (2.4)$$

2.2 Equations of Motion

The equations of motion presented are based on work by Connelly, [2.1] Nelson [2.2], and Roskam [2.3], in which the rotational and translational equations are considered separately.

2.2.1 Rotational Equations of Motion

The moment equations are based on Newton's second law given in Equation 2.5.

$$\frac{d\bar{P}}{dt} = \bar{T} \quad (2.5)$$

If an aircraft is considered to be a collection of discrete particles of mass m_i at a location r_i from the centre of gravity in the body axis system, then the total angular momentum of the aircraft is given by Equation 2.6.

$$\bar{P} = \sum \bar{r}_i \times m_i (\bar{\omega}_b \times \bar{r}_i) \quad (2.6)$$

Breaking the position and angular velocity down into their individual components yields Equations 2.7 to 2.8.

$$\bar{r}_i = x_i \hat{i} + y_i \hat{j} + z_i \hat{k} \quad (2.7)$$

$$\bar{\omega}_b = p\hat{i} + q\hat{j} + r\hat{k} \quad (2.8)$$

Substituting these into equation 2.6 and separating the three components yields Equations 2.9 to 2.11.

$$P_x = I_x p - I_{xy} q - I_{xz} r \quad (2.9)$$

$$P_y = I_y q - I_{yx} p - I_{yz} r \quad (2.10)$$

$$P_z = I_z r - I_{zx} p - I_{zy} q \quad (2.11)$$

Equations 2.12 to 2.17 are the products and moments of inertia which are obtained by integrating over the entire volume of the aircraft.

$$I_{xy} = I_{yx} = \iiint xy \rho(x, y, z) dx_b dy_b dz_b \quad (2.12)$$

$$I_{yz} = I_{zy} = \iiint yz \rho(x, y, z) dx_b dy_b dz_b \quad (2.13)$$

$$I_{zx} = I_{xz} = \iiint zx \rho(x, y, z) dx_b dy_b dz_b \quad (2.14)$$

$$I_x = \iiint (y^2 + z^2) \rho(x, y, z) dx_b dy_b dz_b \quad (2.15)$$

$$I_y = \iiint (x^2 + z^2) \rho(x, y, z) dx_b dy_b dz_b \quad (2.16)$$

$$I_z = \iiint (x^2 + y^2) \rho(x, y, z) dx_b dy_b dz_b \quad (2.17)$$

With few exceptions, aircraft are symmetrical about the x-z plane, and as such it follow that I_{xy} , I_{yx} , I_{yz} , I_{zy} will all be equal to zero. A rare example of an aircraft which is not symmetrical about the x-z plane is Burt Rutan's Boomerang shown in Figure 2.4. When these products of inertia are equal to zero, the momentum Equations reduce to equation 2.18.

$$\bar{P}=(I_x p-I_{xz} r)\hat{i}+(I_y q)\hat{j}+(I_z r-I_{xz} p)\hat{k} \quad (2.18)$$

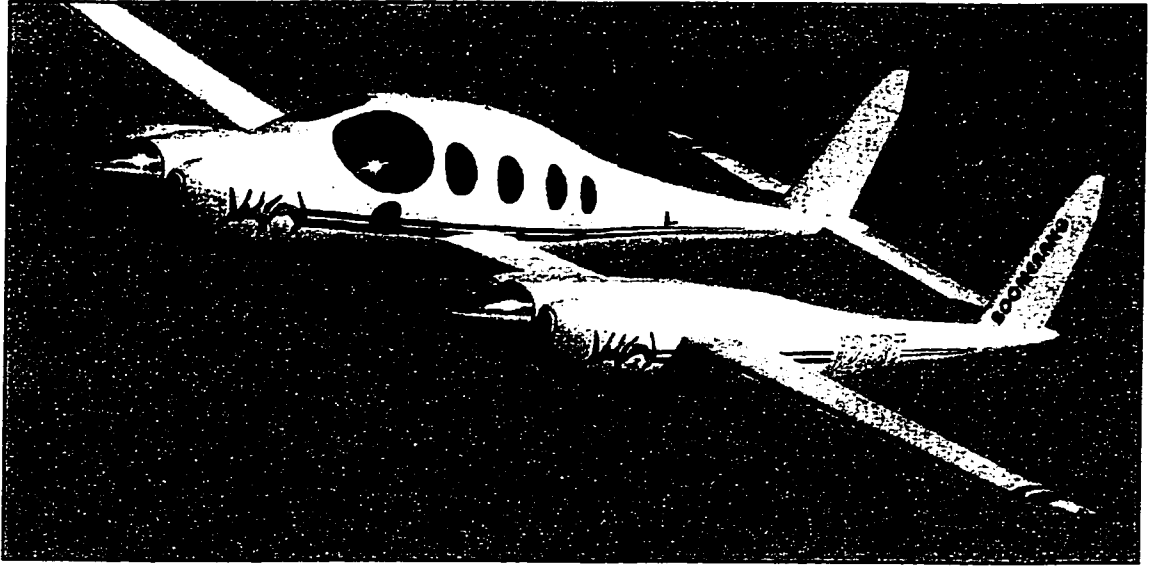


Figure 2.4 Burt Rutan's asymmetrical Boomerang [2.4].

Differentiating the momentum with respect to time gives equation 2.19.

$$\begin{aligned} \frac{d\bar{P}}{dt} = & (I_x \dot{p} - I_{xz} \dot{r})\hat{i} + (I_x p - I_{xz} r) \frac{d\hat{i}}{dt} \\ & + (I_y \dot{q})\hat{j} + (I_y q) \frac{d\hat{j}}{dt} \\ & + (I_z \dot{r} - I_{xz} \dot{p})\hat{k} + (I_z r - I_{xz} p) \frac{d\hat{k}}{dt} \end{aligned} \quad (2.19)$$

The derivatives of the unit vectors along each axis is given by the cross product of the angular velocity with the unit vectors, given by Equations 2.20 to 2.22.

$$\frac{d\hat{i}}{dt} = \bar{\omega}_b \times \hat{i} = r\hat{j} - q\hat{k} \quad (2.20)$$

$$\frac{d\hat{j}}{dt} = \bar{\omega}_b \times \hat{i} = -r\hat{i} + p\hat{k} \quad (2.21)$$

$$\frac{d\hat{k}}{dt} = \bar{\omega}_b \times \hat{k} = q\hat{i} - p\hat{j} \quad (2.22)$$

Substituting these equations into Equation 2.19, collecting the terms, along each axis, and equating them to the moment along each axis gives the final result presented by Equations 2.23 to 2.25.

$$L = I_x \dot{p} + (I_z - I_y) r q - I_{xz} (p q + \dot{r}) \quad (2.23)$$

$$M = I_y \dot{q} + (I_x - I_z) p r - I_{xz} (r^2 - p^2) \quad (2.24)$$

$$N = I_z \dot{r} + (I_y - I_x) p q - I_{xz} (\dot{p} - r q) \quad (2.25)$$

2.2.2 Translational Equations of Motion

The force equations are based on Newton's second law:

$$\bar{F}_b = m \frac{d\bar{V}_b}{dt} = m \left(\dot{u}\hat{i} + u \frac{d\hat{i}}{dt} + \dot{v}\hat{j} + v \frac{d\hat{j}}{dt} + \dot{w}\hat{k} + w \frac{d\hat{k}}{dt} \right) \quad (2.26)$$

Substituting Equation 2.18 into this equation, collecting the terms, along each axis, and equating them to the force along each axis gives the final result presented by Equations 2.27 to 2.29.

$$X = m(\dot{u} - rv + qw) \quad (2.27)$$

$$Y = m(v + ru - pw) \quad (2.28)$$

$$Z = m(w - qu + pv) \quad (2.29)$$

2.3 Aerodynamic Forces and Moments Modelling

The aerodynamic forces and moments acting on an aircraft in flight result from the air flowing over it. They are expressed in the stability axis as the product of a coefficient, the dynamic pressure Q , the wing area S , and in the case of moments, a reference velocity and a characteristic length. The dynamic pressure is given by Equation 2.30.

$$Q = \frac{1}{2} \rho V_{TRUE}^2 \quad (2.30)$$

Grouping the force and moment equations into longitudinal and lateral equations, longitudinal lift, drag, and pitch are computed. These are presented by Equations 2.31 to 2.33.

$$Lift = C_L QS \quad (2.31)$$

$$Drag = C_D QS \quad (2.32)$$

$$Pitching\ moment = C_m QS \bar{c} \quad (2.33)$$

Similarly, the lateral side force, roll, and yaw are obtained and presented by Equations 2.34 to 2.36.

$$\text{Side force} = C_y QS \quad (2.34)$$

$$\text{Rolling moment} = C_l QSb \quad (2.35)$$

$$\text{Yawing moment} = C_n QSb \quad (2.36)$$

The coefficients are not constant. They depend on the aircraft's geometry, the conditions of flight, and the position of the control surfaces. Assuming that the aircraft is rigid and does not have a variable geometry such as variable sweep wings, then the coefficients will depend only on the condition of flight and the position of the control surfaces. The symbols and sign conventions used to define the position of the control surfaces are given in Figure 2.5.

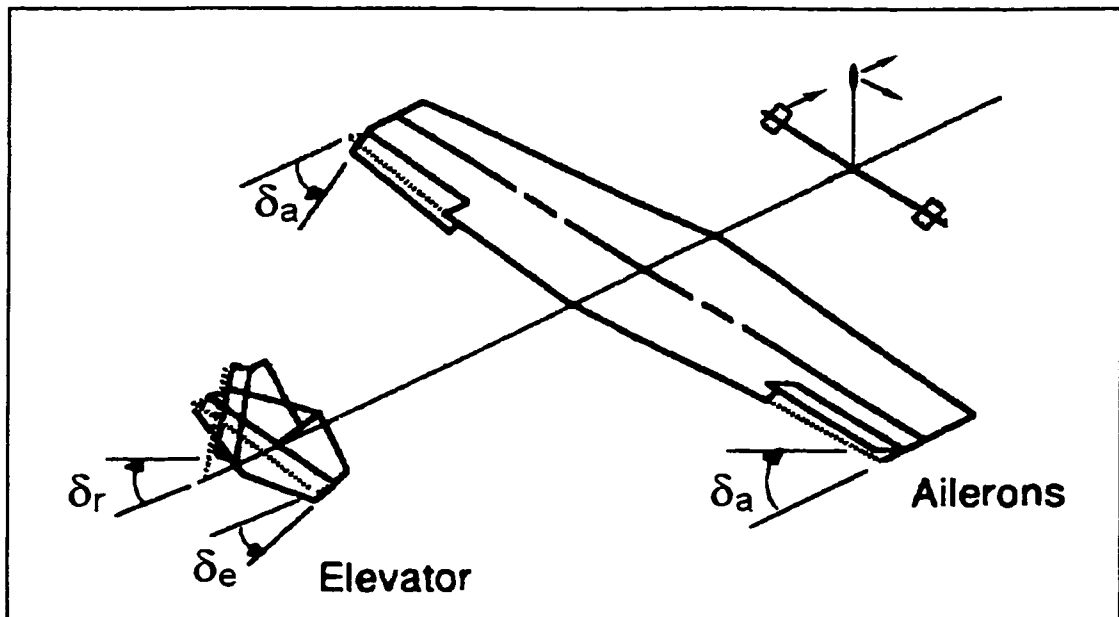


Figure 2.5 Control surface deflection. [2.2]

The coefficients do not vary linearly, especially at velocities above mach 0.2 [2.5], where the effects of compressibility start to become significant. To deal with the nonlinearities, the aircraft is considered to be operating close to specified flight condition about which the coefficients are approximated by linear equations. This approximation allows the coefficients to be expressed as the sum of components representing various variables for the given condition of flight. The longitudinal coefficients can now be expressed by Equations 2.37 to 2.39, and the lateral coefficients by Equations 2.40 to 2.42.

$$C_L = C_{L0} + C_{L\alpha}\alpha + C_{L\dot{\alpha}}\left(\frac{\dot{\alpha}\bar{c}}{2U_0}\right) + C_{Lq}\left(\frac{q\bar{c}}{2U_0}\right) + C_{L\delta_e}\delta_e \quad (2.37)$$

$$C_D = C_{D0} + C_{D\alpha}\alpha + C_{D\delta_e}\delta_e \quad (2.38)$$

$$C_m = C_{m0} + C_{m\alpha}\alpha + C_{m\dot{\alpha}}\left(\frac{\dot{\alpha}\bar{c}}{2U_0}\right) + C_{mq}\left(\frac{q\bar{c}}{2U_0}\right) + C_{m\delta_e}\delta_e \quad (2.39)$$

$$C_Y = C_{Y0} + C_{Y\beta}\beta + C_{Yp}\left(\frac{pb}{2U_0}\right) + C_{Yr}\left(\frac{rb}{2U_0}\right) + C_{Y\delta_r}\delta_r \quad (2.40)$$

$$C_l = C_{l0} + C_{l\beta}\beta + C_{lp}\left(\frac{pb}{2U_0}\right) + C_{lr}\left(\frac{rb}{2U_0}\right) + C_{l\delta_a}\delta_a + C_{l\delta_r}\delta_r \quad (2.41)$$

$$C_n = C_{n0} + C_{n\beta}\beta + C_{np}\left(\frac{pb}{2U_0}\right) + C_{nr}\left(\frac{rb}{2U_0}\right) + C_{n\delta_a}\delta_a + C_{n\delta_r}\delta_r \quad (2.42)$$

2.4 Thrust Model

In this work the aircraft being modelled is a twin engine aircraft with the engines mounted symmetrically about the $X_B Z_B$ plane with the lines of actions parallel to the X_B axis. The effects of prop, or jet wash over the control surfaces would normally be represented in the control coefficients as an increase in their efficiency, and are therefore not addressed here. The thrust model shown in Figure 2.6 then becomes very simple. Given that the engines' line of action is located Z_E below the centre of gravity, Y_E on either side of the $X_B Z_B$ plane, and the thrust from the left and right engines are respectively T_L , and T_R , then, the force, pitching, and yawing moments in the body axis are found using Equations 2.43 to 2.45. [2.2]

$$X_T = T_L + T_R \quad (2.43)$$

$$M_T = Z_E (T_L + T_R) \quad (2.44)$$

$$N_T = Y_E (T_L - T_R) \quad (2.45)$$

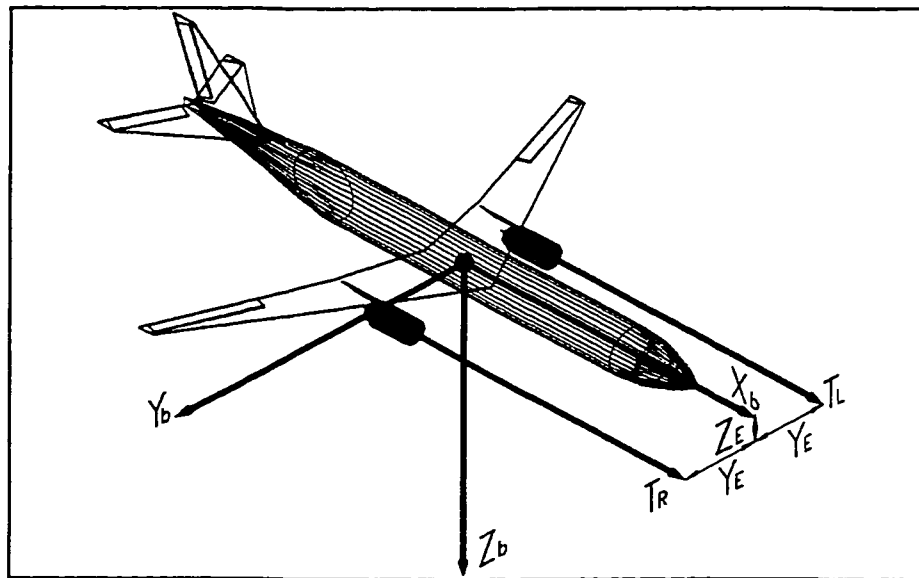


Figure 2.6 Thrust vector point of application.

2.5 Gravity Model

In the fixed axis system, gravity always acts in the positive Z direction. Referring to Figure 2.7, the gravity is transformed to the body axis by Equations 2.46 to 2.48. [2.2]

$$X_g = -mg \sin \theta \quad (2.46)$$

$$Y_g = mg \cos \theta \sin \phi \quad (2.47)$$

$$Z_g = mg \cos \theta \cos \phi \quad (2.48)$$

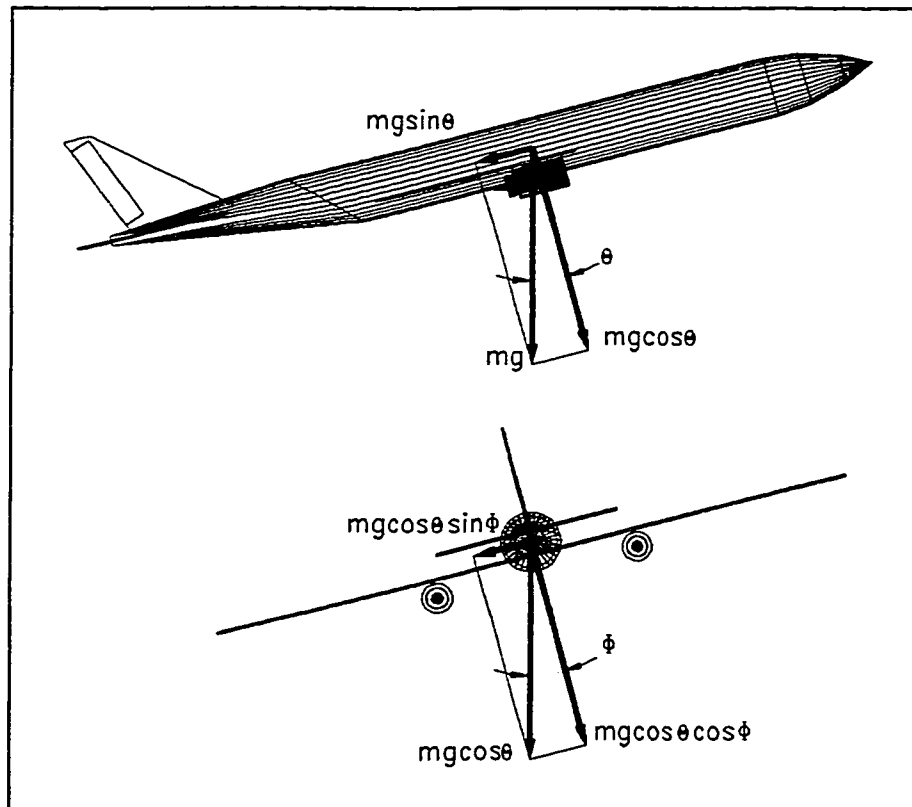


Figure 2.7 Gravity in the body axis system.

2.6 Ground Model

Although the primary goal of the simulation portion of this work is to implement a reconfigurable flight simulation program for teaching purposes, a ground model is needed so that the aircraft being simulated may be taxied, taken off and landed. Current ground models such as that described in Boeing Report D6-44062 [2.6] are complex beyond the requirements of this work, and require a thorough understanding of the interaction between the rubber tire and different landing surfaces under different loads. The parameters needed to describe this interaction are not readily available. This level of complexity would require unnecessary detail when defining a new aircraft. For this reason a hybrid kinematic dynamic model was developed requiring only geometric configuration and the struts spring and damping coefficients to be defined.

The hybrid ground model assumes that the landing gear can only generate forces in the Z body axis and no side forces. The side forces normally needed to turn the aircraft on the ground are avoided by switching to a kinematic model for the yaw, X and Y velocities. The model is based on the geometry of the landing gear, the steering angle of the nose gear, and the assumptions that a point contact with no slipping exists between the tires and the ground. This is shown in Figure 2.7.

2.6.1 Ground Kinematic Model

When all landing gear is in contact with the ground, then by using a kinematic model and referring to Figure 2.8, the radius of rotation of the centre line at the main landing gear can be obtained. It is given by Equation 2.49.

$$R_{main} = \frac{X_{nose} - X_{main}}{\tan(e)} \quad (2.49)$$

The difference in heading between the centre of gravity and the centre line of the aircraft is given by Equation 2.50.

$$T = \tan^{-1} \left(\frac{-X_{main}}{R_{main}} \right) \quad (2.50)$$

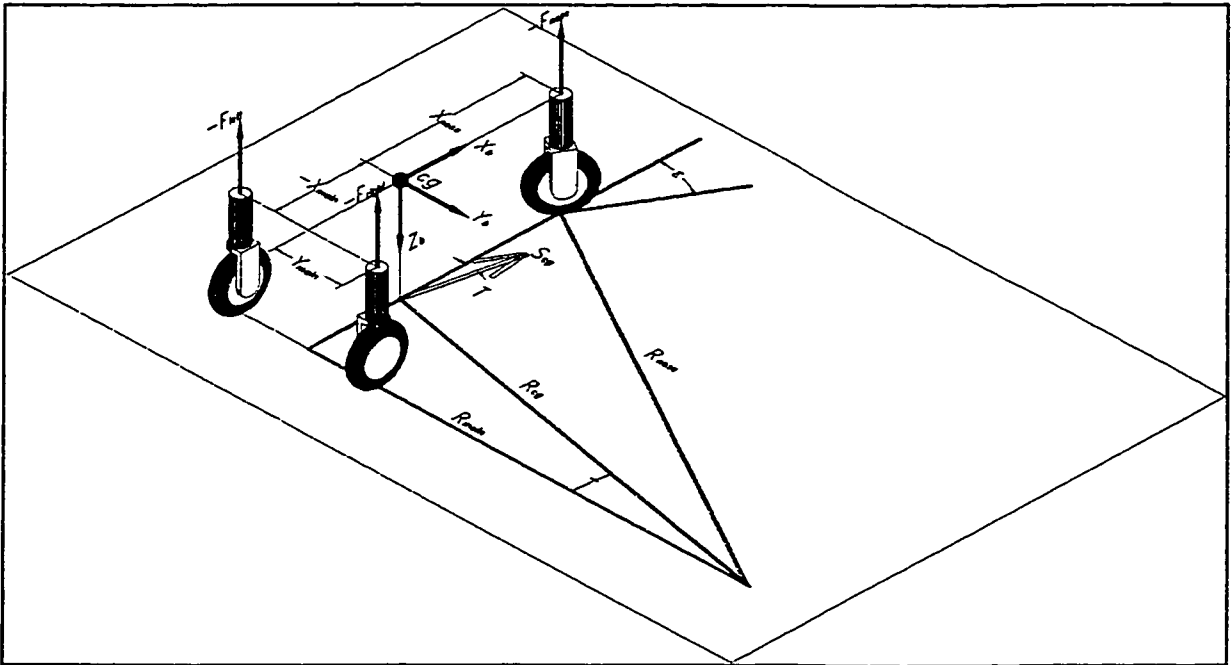


Figure 2.8 Ground kinematic model.

The radius of rotation of the centre of gravity is then found using Equation 2.51.

$$R_{cg} = \frac{R_{main}}{\cos(T)} \quad (2.51)$$

The speed of the centre of gravity in the $X_F Y_F$ plane is given by Equation 2.52.

$$S_{cg} = \sqrt{V_{XF}^2 + V_{YF}^2} \quad (2.52)$$

Combining these expressions yields the new yaw, X, and Y velocities in the fixed axis system as a result of the nose wheel steering angle. These are given in Equations 2.53 to 2.55.

$$\frac{d\psi}{dt} = \frac{S_{CG}}{R_{CG}} \quad (2.53)$$

$$V_{XF} = S_{CG} \cos(\psi + T) \quad (2.54)$$

$$V_{YF} = S_{CG} \sin(\psi + T) \quad (2.55)$$

2.6.2 Ground Dynamic Model

The ground dynamic model is used to model the forces which act in the Z body axis as a result of the compression and expansion of the landing gear. Each strut is modelled as a spring damper system. All landing gear is assumed to be of equal length, mounted on the XY body plane. The two main struts are assumed to have identical spring and damping constants which are not necessarily equal to those of the nose gear. The method of determining the forces from each strut is identical.

Firstly, the compression of the strut is determined. Referring to figure 2.9, by calculating the location of the top (aircraft mount) and bottom (wheel) of a fully extended strut, which may be below the ground in the fixed axis system, the compression of the strut is found with Equation 2.56.

$$C_g = \frac{-Z_{F_{top}}}{Z_{F_{ext}} - Z_{F_{top}}} L_g \quad (2.56)$$

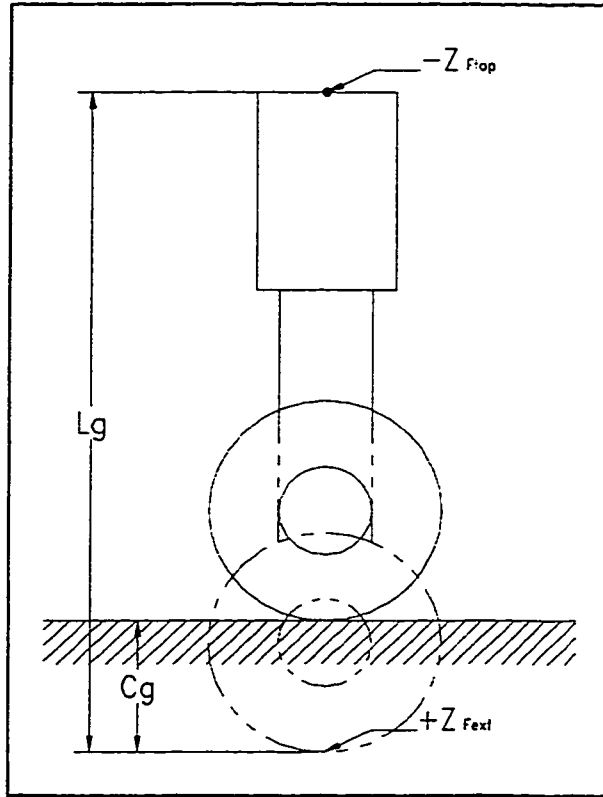


Figure 2.9 Landing strut compression.

The rate of compression given by Equation 2.57 is the difference between the previous and present compression divided by the elapsed time.

$$\dot{C}_g = \frac{C_{g(j)} - C_{g(j-1)}}{\Delta t} \quad (2.57)$$

Knowing the compression and rate of compression of the strut, the resulting force on the strut is found with Equation 2.58.

$$F_g = -C_g K_{S_g} - \dot{C}_g K_{d_g} \quad (2.58)$$

Since the landing gear cannot “pull” the aircraft towards the ground, any positive force resulting from the landing gear should be ignored. Once the force in each strut is known, then referring to figure 2.8, the net vertical force in the body axis as well

and the rolling and pitching moments may be determined using Equations 2.59 to 2.61.

$$Z_{gear} = Z_{g_n} + Z_{g_1} + Z_{g_r} \quad (2.59)$$

$$L_g = (Z_{g_r} - Z_{g_1}) Y_{main} \quad (2.60)$$

$$M_g = -Z_{g_n} X_{nose} - (Z_{g_1} + Z_{g_r}) X_{main} \quad (2.61)$$

This model does not address braking. To include braking, a simple braking model considers the left and right landing gear braking equally with a force X_{brk} . This will also cause a negative pitching moment given by Equation 2.62.

$$M_{brk} = X_{brk} Z \quad (2.62)$$

These force and moments are added to the aerodynamic forces and moments acting on the aircraft to give the total forces and moments on the aircraft for the dynamic equations.

2.7 Atmospheric Model

Atmospheric conditions depend on altitude. Significant variations in temperature and pressure can affect an aircraft's engines as well as its aerodynamic characteristics. To quantify atmospheric conditions, a reference temperature and pressure was adopted by the International Civil Aviation Organization, ICAO, to describe the atmosphere at sea level on a standard day as shown in table 2.3.

Pressure P_o	101.3 kPa
Density ρ_o	1.225 kg/m ³
Temperature T_o	288.16 °K
Absolute viscosity μ_o	1.789x10 ⁻³ Ns/m ²
Kinematic viscosity ν_o	1.460x10 ⁻³ m ² /s
Temperature lapse rate a	-6.5 °C/1000m
Gas constant R	29.26 m ² /K

Table 2.3 Standard atmosphere at sea level.

Figure 2.10 illustrates the temperature of the standard atmosphere consisting of gradient and isothermal regions. Most aircraft operate at altitudes below 11 000 meters (36089 feet) or slightly higher. For this reason, only the lowest gradient region from sea level to 11 000 meters and the first isothermal region above it will be considered.

2.7.1 Standard Atmospheric Conditions Below 11 000 Meters

Below 11 000 meters, the standard atmospheric temperature decreases linearly with a lapse rate of $a = -6.50 \text{ deg K/1000 m}$. Taking air to be an ideal gas and recalling that increasing altitude is in the negative Z_F direction, the temperature [°K], pressure [kPA], and density [kg/m³] may be found with Equations 2.63 to 2.65. [2.7]

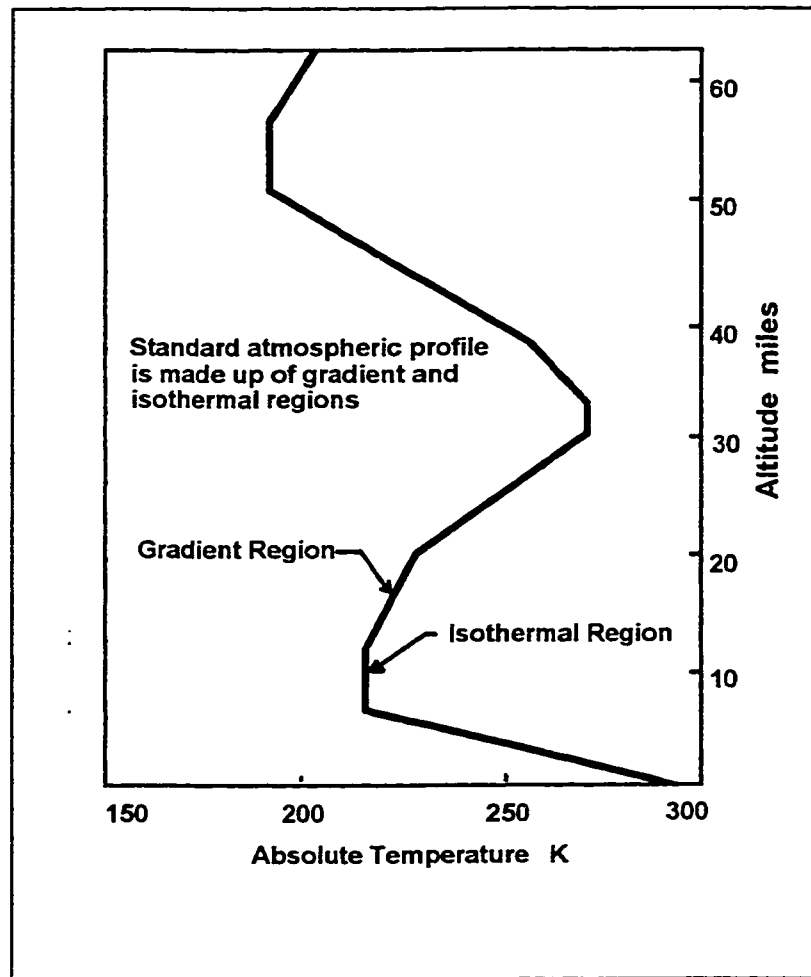


Figure 2.10 Standard ICAO atmosphere. [2.2]

$$T = 288.16 + 6.5 \left(\frac{z}{1000} \right) \quad (2.63)$$

$$P = P_o \left(\frac{T}{T_o} \right)^{-\frac{1}{aR}} = 101.3 \left(\frac{T}{288.16} \right)^{5.2561} \quad (2.64)$$

$$\rho = \rho_o \left(\frac{T}{T_o} \right)^{-\left(\frac{1}{aR} + 1 \right)} = 1.225 \left(\frac{T}{288.16} \right)^{4.2561} \quad (2.65)$$

At 11 000 meters, the conditions would be as shown in Table 2.4. These conditions become the reference point for determining the atmospheric conditions above 11 000 meters.

Pressure P_{11000}	22.6 kPa
Density ρ_{11000}	0.3639 kg/m ³
Temperature T_{11000}	216.66 °K

Table 2.4 Standard atmosphere at 11 000 meters.

2.7.2 Standard Atmospheric Conditions Above 11 000 Meters

Above 11 000 meters, the temperature remains constant at 215.7 °K. In this region the pressure [kPa] and density [kg/m³] are given by Equations 2.66 and 2.67. [2.7]

$$P = 26.6 \exp\left(\frac{z + 11000}{6339.5} \right) \quad (2.66)$$

$$\rho = 0.3639 \exp\left(\frac{z + 11000}{6339.5} \right) \quad (2.67)$$

The results of this model in both regions are as calculated in the simulation program, and are plotted in Figure 2.11.

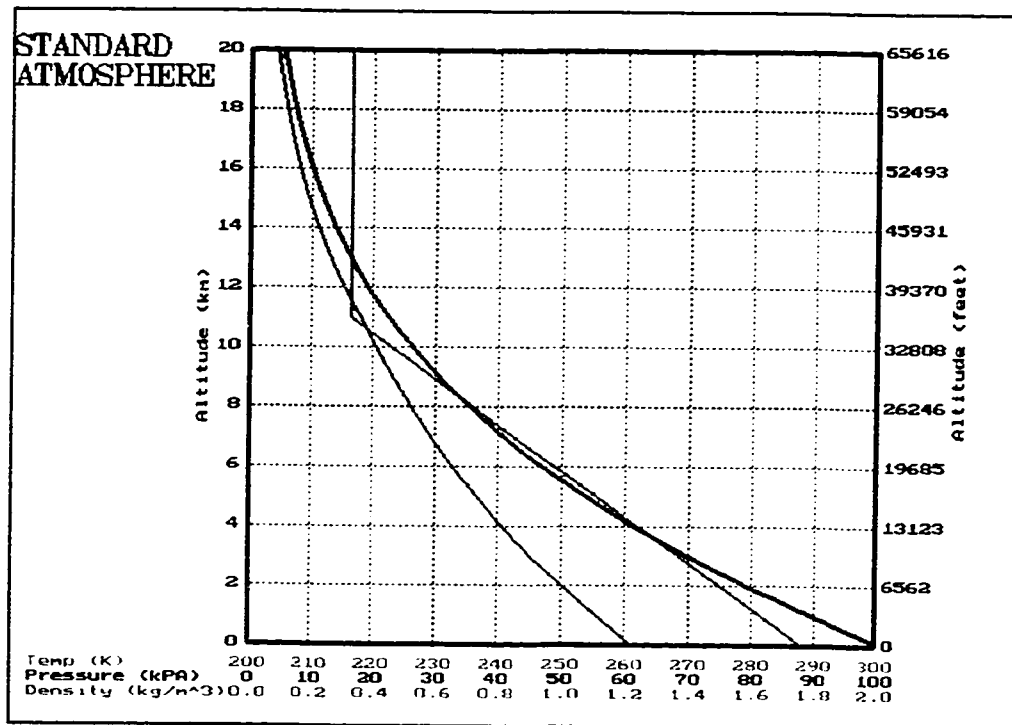


Figure 2.11 Simulated ICAO standard atmosphere.

3.0 COMPUTER CODE IMPLEMENTATION OF FLIGHT MODEL

The six degree of freedom model presented in Chapter 2 has been implemented as a C program with features to completely define the stability and control coefficients, to graphically display the instruments, to monitor the performance, and to run in a number of configurations ranging from a stand alone personal computer, to one interfacing with a fully instrumented aircraft cockpit and visual system.

The reason for the 80X86 operating platform is one of cost and upgradability. As a teaching tool this program should be able to run on machines which are readily available to students and teachers. The PC family of computers easily fits this criterion while more powerful work stations are less available. The second reason for choosing the 80X86 platform is because the 80X86 family is being driven very hard to constantly develop higher performance models which are always downward compatible. As faster 80X86 computers become available, the code will run without modification. In its current form, the code is not portable to other computers because of special calls to specific PC functions such as the joystick, mouse, and interface cards. For best performance, this program should not be implemented on less than a 80486DX-33 with a VGA display.

3.1 Implementation of Flight Model Simulation

Figure 3.1 outlines the relationships between the components of the flight simulation program. Figure 3.2 is a flow chart used to implement the model, including a routine to modify the configuration of the aircraft and display or store various program parameters in real time. This makes the system a valuable tool to study flight dynamics and to examine the effects of varying the program parameters.

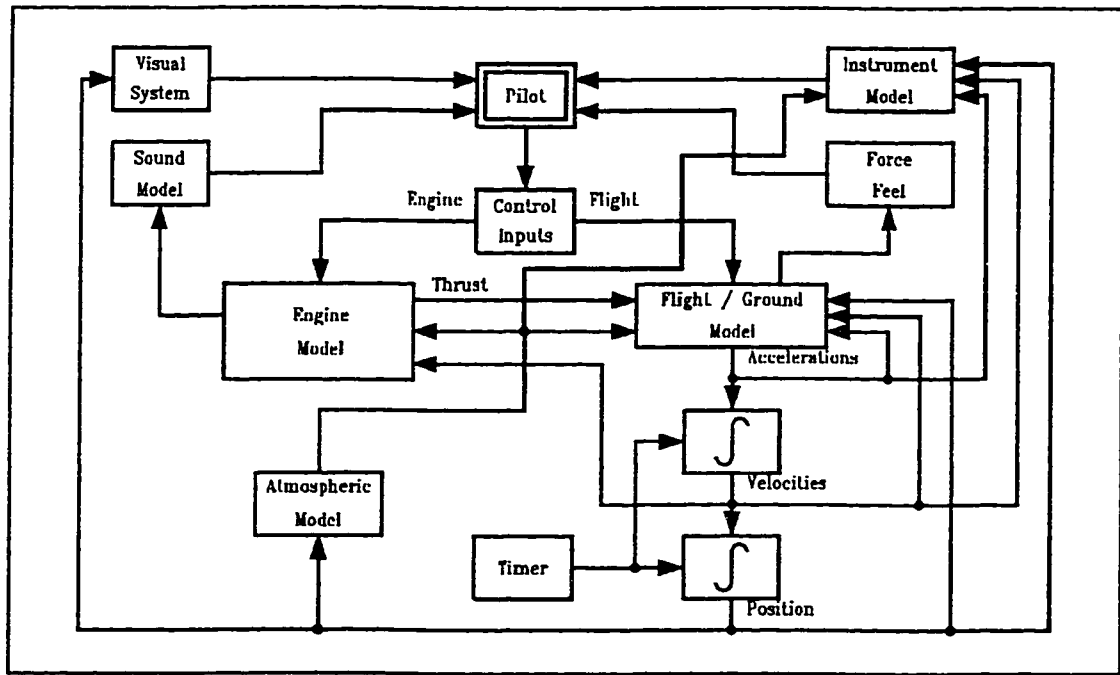


Figure 3.1 Flight model block diagram.

The flow chart presented in figure 3.2 shows the sequential steps taken to implement the flight model as a simulation code. Appendices 1 through 8 list the major C routines used in the simulation program [3.1][3.2][3.3].

3.1.1 Program Initial Conditions

The program has a set of initial conditions to define the aircraft's position, orientation, translational and rotational velocity. Initially, the aircraft is placed at the end of a runway, at an altitude equal to the length of its landing gear, ready for takeoff, and with zero velocity. The engines are off, the landing gear is extended and all control surfaces are in a neutral position. A default aircraft configuration is loaded consisting of all the stability and control coefficients to define a typical aircraft light twin engine aircraft. The user is free to modify the aircraft's characteristics and initial conditions.

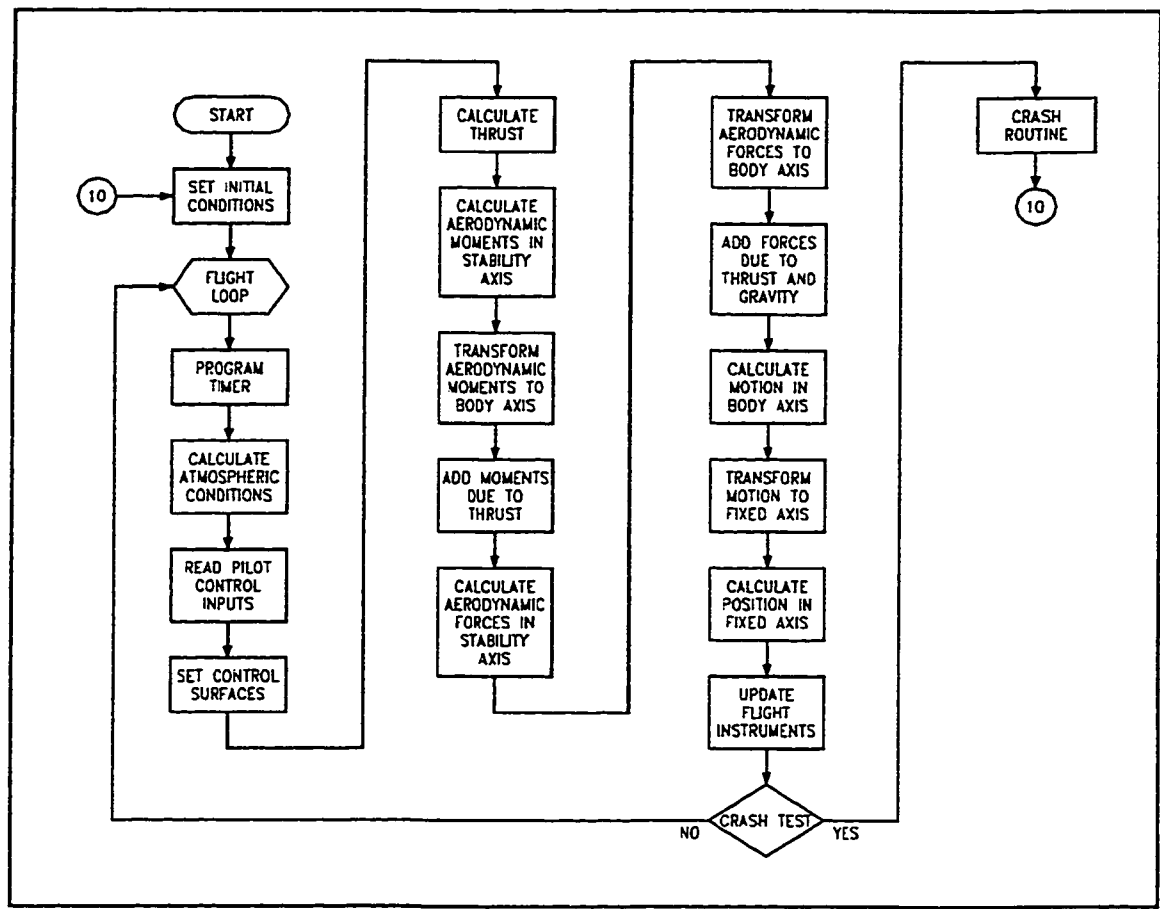


Figure 3.2 Flight dynamic model implementation flow chart.

3.1.2 Simulation Loop

The simulation loop is the main simulation routine, it recalculates all the flight variables at the maximum speed of the computer according to the flow chart in figure 3.2.

3.1.2.1 The Program Timer

For the program to run in real time, it is essential to keep track of the elapsed time and the period of each iteration Δt . The iteration period is used to calculate the velocities, attitude and position by numerically integrating

accelerations. The PC computer has an internal clock which returns the elapsed time since January 1, 1970. The accuracy of the returned time is to within 18 msec. Since a simulation cycling at 50 Hz would have a period of 20 msec, an unacceptable error would occur. This error is reduced by averaging the elapsed time over two seconds according to the current period. The program then compares the measured elapsed time to that of the iteration period and the number of cycles since the previous measurement and makes any necessary changes to the value of the period. In this way the program will always run as fast, and with the highest fidelity, as the PC can support. Since the elapsed time is always measured since the simulation began, there are no cumulative errors in calculating the period. The routine used to monitor and update the period is given in Appendix 2.

In cases where it is not necessary to run the simulation in real time, for example when it is being controlled by commands stored in a file, the period could be set to any value. This would increase the fidelity of the simulation, allowing slower computers to arrive at the same result as faster machines, but at the cost of running in real time.

3.1.2.2 Calculation of Atmospheric Condition

The atmospheric conditions are calculated according to the ICAO model presented in Section 2.6. Appendix 3 lists the routine used to calculate the atmospheric conditions. The user has the option to use this model or define the atmospheric conditions regardless of the altitude.

3.1.2.3 Reading of Control Inputs

When running in a stand alone PC configuration, the pilot's control inputs are read through the PC's joystick port. The joystick port is typically a slow interface and will have a significant effect on the performance of the simulation. This is because of the method used to read it. The variable resistive element in the joystick is the resistive element in a capacitor-resistor circuit. Changing the position of the joystick changes the circuit's time constant, and measuring the time constant is a slow process which slows down the entire simulation. [3.4] Another limitation is that not all joysticks and ports necessarily give the same result when centred. To overcome this, the centre position of the joystick and pedals is measured before the simulation begins and used as a reference for all subsequent readings of the joystick.

When connected to the cockpit section, the controls inputs are read by a high speed data acquisition card, increasing the iteration rate compared to reading the joystick. This results in a higher simulation fidelity. A more detailed description of this system is presented in Chapter 6.

3.1.2.4 Setting of Control Surfaces

This section is included for future development. At present, the control surfaces will move proportionally and simultaneously with the control inputs. In future work, this section could contain models describing effects such as friction and elasticity in the control cables as well as the behaviour of any actuators.

3.1.2.5 Calculation of the Aerodynamic Forces and Moments in Stability Axis

To calculate the aerodynamic forces in the stability axis, the stability and control coefficients are used in Equations 2.37 to 2.42 and subsequently in Equations 2.31 to 2.36 to find the forces and moments in the stability axis system. The angle of attack and sideslip are found using Equations 2.1 and 2.2 with the velocities of the previous iteration. In the case of the first iteration, the initial conditions are used.

3.1.2.6 Transformation of the Aerodynamic Forces and Moments to Body Axis

The dynamic Equations 2.23 to 2.25 and 2.27 to 2.29 require the forces and moments in the body axis. Since they are calculated in the stability axis, they must be transformed to the body axis by rotating them about the Y body axis (or stability) axis by the angle of attack. This is achieved with Equations 3.1 and 3.2.

$$\begin{bmatrix} Drag_B \\ Side_B \\ Lift_B \end{bmatrix} = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix} \begin{bmatrix} Drag_s \\ Side_s \\ Lift_s \end{bmatrix} \quad (3.1)$$

$$\begin{bmatrix} L_B \\ M_B \\ N_B \end{bmatrix} = \begin{bmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{bmatrix} \begin{bmatrix} L_s \\ M_s \\ N_s \end{bmatrix} \quad (3.2)$$

3.1.2.7 Calculation of the Forces and Moments Due to Thrust

The thrust is modelled as two forces acting forward, parallel to the aircraft's longitudinal axis at a user definable position in the lateral plane. Future work will explore the development of a piston and turbine engine model with both jet thrust or variable pitch propellers.

The resultant thrust would not only propel the aircraft forward but, would also cause a pitching moment dependant on the thrust and the engines' vertical position relative to the centre of gravity. Any asymmetric thrust would result in a yawing moment. These forces and moments are given by Equations 2.43 to 2.45.

3.1.2.8 Summation of Forces and Moments in Body Axis

By combining the aerodynamic, thrust and gravity forces and moments in the body axis, the net forces and moments acting on the aircraft are found. Note that both positive lift and drag are actually in the negative direction.

$$X = -Drag + X_T + X_g \quad (3.3)$$

$$Y = Y_A + Y_g \quad (3.4)$$

$$Z = -Lift + Z_g \quad (3.5)$$

$$L = L_A \quad (3.6)$$

$$M = M_A + M_T \quad (3.7)$$

$$N = N_A + N_T \quad (3.8)$$

3.7.2.9 Calculation of Motion in Body Axis

Using the dynamic Equations 2.23 to 2.25 and 2.27 to 2.29 the translational and rotational accelerations in the body axis are found from the results of Equations 3.3 to 3.8. These accelerations are numerically integrated according to Equations 3.9 to 3.14 to give the velocities.

$$\dot{u} = \dot{u} + \dot{u}\Delta t \quad (3.9)$$

$$\dot{v} = \dot{v} + \dot{v}\Delta t \quad (3.10)$$

$$\dot{w} = \dot{w} + \dot{w}\Delta t \quad (3.11)$$

$$\dot{p} = \dot{p} + \dot{p}\Delta t \quad (3.12)$$

$$\dot{q} = \dot{q} + \dot{q}\Delta t \quad (3.13)$$

$$\dot{r} = \dot{r} + \dot{r}\Delta t \quad (3.14)$$

3.1.2.10 Transformation of the Motion to the Fixed Axis System

The velocities in the body axis are transformed to the fixed axis system using Equation 2.3 and 2.4. These velocities are numerically integrated using Equations 3.15 to 3.20 to obtain the aircraft's position and attitude in the fixed axis.

$$x = x + \dot{x}\Delta t \quad (3.15)$$

$$y = y + \dot{y}\Delta t \quad (3.16)$$

$$z = z + \dot{z}\Delta t \quad (3.17)$$

$$\theta = \theta + \dot{\theta}\Delta t \quad (3.18)$$

$$\phi = \phi + \dot{\phi}\Delta t \quad (3.19)$$

$$\psi = \psi + \dot{\psi}\Delta t \quad (3.20)$$

3.2 Simulated Aircraft Configuration

3.2.1 Stability and Control Coefficients

The program offers the user full flexibility to redefine the stability and control coefficients, they are grouped by the axis on or about which they act. By altering these coefficients, the user quickly develops a feel of how to modify the coefficients to obtain a desired result.

3.2.2 Control Surface Deflection

The maximum control surface deflection corresponding to maximum travel on the control column can also be specified. However, one is cautioned that the model used by this program is a linear one and will not accurately model excessive control surface deflection in the nonlinear range. Although it is not a

control surface, the maximum nose wheel steering angle is included with the control surfaces since it is used for ground directional control.

3.2.3 Engine Configuration

The engines are presently modelled as two symmetrically located thrust vectors, not necessarily of equal magnitude. The user may specify the location as well as the maximum magnitude of the thrust vectors corresponding to the maximum throttle position.

3.2.4 Landing Gear Configuration

The landing gear modelled in Section 2.5 is a tricycle configuration with a steerable nose wheel located on the aircraft's $X_B Z_B$ plane. The user may specify the position and length of the nose and main gear in the body axis relative to centre of gravity, as well as the damping and spring constant of both the nose and main landing gear. Experience has shown that it is a tedious task to select appropriate damping and spring constant. For this reason an automatic tuning option was included that selects the spring constants such that when the aircraft is at rest, it will be level with all gears compressed twenty five percent. Using this feature, the landing gear may be completely defined by its geometrical arrangement. Referring to Figure 2.8, Equations 3.21 and 3.22 were derived from the static moment equations to give the spring constants for the main and nose struts. With the spring constants and the load on the struts, Equations 3.23 and 3.24 were derived to give the damping coefficient for the critically damped case.

$$K_{S_{nose}} = - \frac{mg X_{main}}{0.25 Z_g (X_{nose} - X_{main})} \quad (3.21)$$

$$K_{S_{main}} = \frac{mg X_{nose}}{0.5 Z_g (X_{nose} - X_{main})} \quad (3.22)$$

$$K_{b_{nose}} = 2.0 \sqrt{- \frac{mX_{main}}{X_{nose} - X_{main}} K_{S_{nose}}} \quad (3.23)$$

$$K_{b_{main}} = 2.0 \sqrt{\frac{mX_{nose}}{2 (X_{nose} - X_{main})} K_{S_{main}}} \quad (3.24)$$

3.2.5 Initial Position and Atmospheric Conditions

To specifically define the initial condition, the user has the ability to specify the position, orientation, velocities, control surface deflections, throttle position, and atmospheric conditions. The user may select either a standard atmosphere, which will have a varying temperature, pressure, and density as described in Section 2.6 or an atmosphere with constant conditions of a specified altitude.

Two common initial positions are at the end of the runway, and steady straight flight. These two initial positions are therefore programmed as a single key option. If the user selects to position the aircraft at the end of the runway, the aircraft will be placed there, facing down the runway with the engines off. If straight and level flight is selected, then an airspeed and a desired angle of climb must be specified. The program then calculates the power, angle of attack, and control surface deflections such that if undisturbed, the aircraft will fly in a straight line

indefinitely with the wings level along the specified angle of climb. This is achieved by solving the longitudinal equations simultaneously so that no net force or moment is acting on the aircraft. The equations for this are given by Equations 3.25 to 3.27.

$$mg\sin\theta = -(C_D + C_{D_\alpha}\alpha + C_{D_{\delta_e}}\delta_e)QS + T_T\cos\phi \quad (3.25)$$

$$-mg\cos\phi\cos\theta = (C_L + C_{L_\alpha}\alpha + C_{L_{\delta_e}}\delta_e)QS + T_T\sin\phi \quad (3.26)$$

$$0 = (C_m + C_{m_\alpha}\alpha + C_{m_{\delta_e}}\delta_e)QS\bar{c} - T_TZ_E \quad (3.27)$$

3.3 Simulation Control Configuration

This program was written to run in a number of different configurations, depending on the available control hardware.

Firstly, the control inputs which the pilot would normally provide may come from a number of sources. The available combinations are:

- 1) Joystick, keyboard, mouse
- 2) Joystick, pedals, keyboard, mouse
- 3) Control file
- 4) None, locked controls
- 5) Full cockpit simulation

The most common configuration is to use the computer's joystick port to control the elevator and ailerons, and the mouse to control the left or right engine depending which button is pressed. In this configuration, the rudder is locked in the neutral position, giving some control limitations, but the aircraft is 'flyable'. The logical progression is to include rudder pedals to the computer's joystick port, with

the joystick and mouse functioning as in the previous case.

It is not reasonable to expect someone to manually fly specific repeatable test control input. For this reason the simulation can run from a control file which can execute specified control inputs at predetermined times. The format of the control file is ASCII text where, each command is on a separate line made up of three parts separated by a single space, a two character code identifying the control and the type of change, the time to implement the control, and the magnitude of the control input. The options for the identification are given in Table 3.1 and the type of control change in Table 3.2.

Ailerons	A
Elevator	E
Rudder	R
Left Engine (Port)	P
Right Engine (Starboard)	S
Stop	X

Table 3.1 Control identification

Absolute	A
Incremental	I
Proportional	P

Table 3.2 Control changes

All control surface inputs vary from +1 to -1 for full deflection in either direction. An absolute change will set the control surface to the specified point. An incremental change will add or subtract the specified amount from the current position. A proportional change will multiply the current position by the specified amount. The program will limit all control surfaces to within their maximum

deflections. The engine controls work in exactly the same manner except that the inputs vary from 0 to +1. To stop the simulation, X is used as a control identification, followed by a dummy control change and then the time to stop. The following is an example of a program to execute a 50% absolute elevator deflection after 5 seconds, followed by a 50% proportional reduction after 10 second bringing the deflection to 25%, then a -45% incremental deflection bringing the deflection to -20% after 15 seconds and then, bringing the simulation to a stop at 20 seconds.

```
EA 5.0 0.5  
EP 10.0 -0.5  
EI 15.0 -0.45  
XX 20.0 0.0
```

3.4 Computer Display Configuration

A number of display options are available to present real time information. If the system is running on a stand alone computer, a graphic display of the flight instruments is available to facilitate flying and navigation. A map display offers the ability to monitor the aircraft's position and performance. A strip chart display is capable of monitoring up to eight simultaneous values in real time.

3.4.1 Flight Instruments Display

The flight instruments would be displayed on the screen as shown in Figure 3.3, which is a representation of an Electronic Flight Instrument System (EFIS) display. It includes an attitude indicator to represent the pitch and roll angles in the center, an air speed indicator tape on the left, an altimeter tape on the right above a numerical vertical speed display, and a magnetic heading indicator below.

To the left of the heading indicator is a representation of the control surfaces which deflects according to their setting. This is particularly useful when the simulation is being flown using the mouse as a control column since, unlike the joystick, the mouse does not naturally return a neutral position. An actual EFIS display is shown in Figure 3.4. Some liberties were taken in the implementation of the EFIS display to minimize time intensive graphic routines which would reduce the iteration rate and decrease the fidelity of the model. In the upper right hand corner are the navigational instruments. A detailed description of the navigation systems will be presented in Chapter 6. Below the navigation instruments is the throttle quadrant. For the linear thrust model and jet engines, only the throttles would be used. In future work, a constant speed propeller would also use the pitch control, and a piston engine would also use the mixture controls to control the air fuel ratio. Below the EFIS are switches to control the master power, left and right engines, landing gear, and flaps.

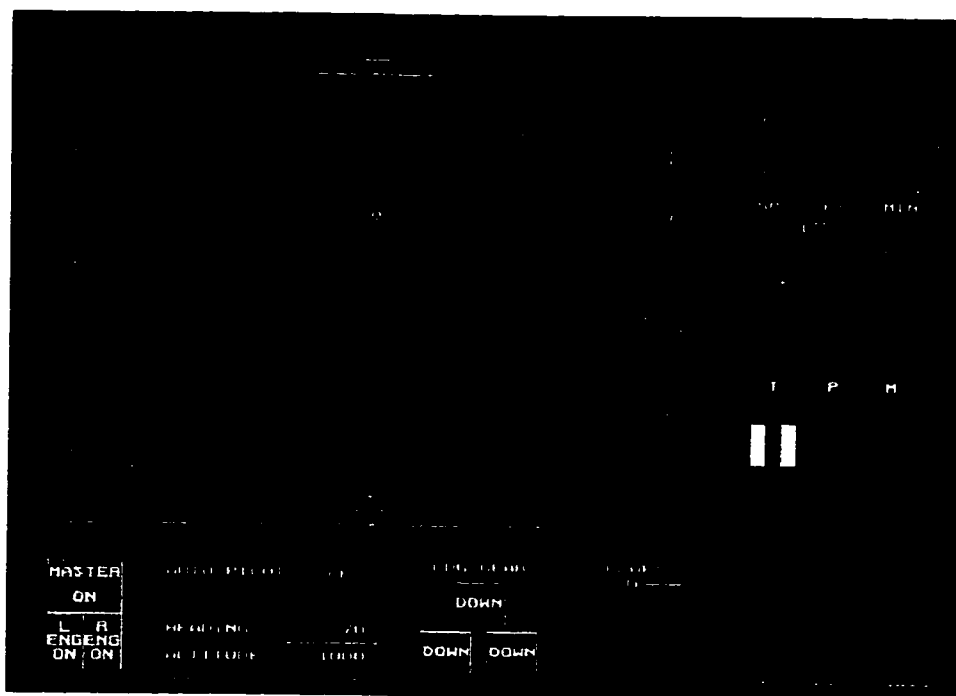


Figure 3.3 Simplified EFIS display.

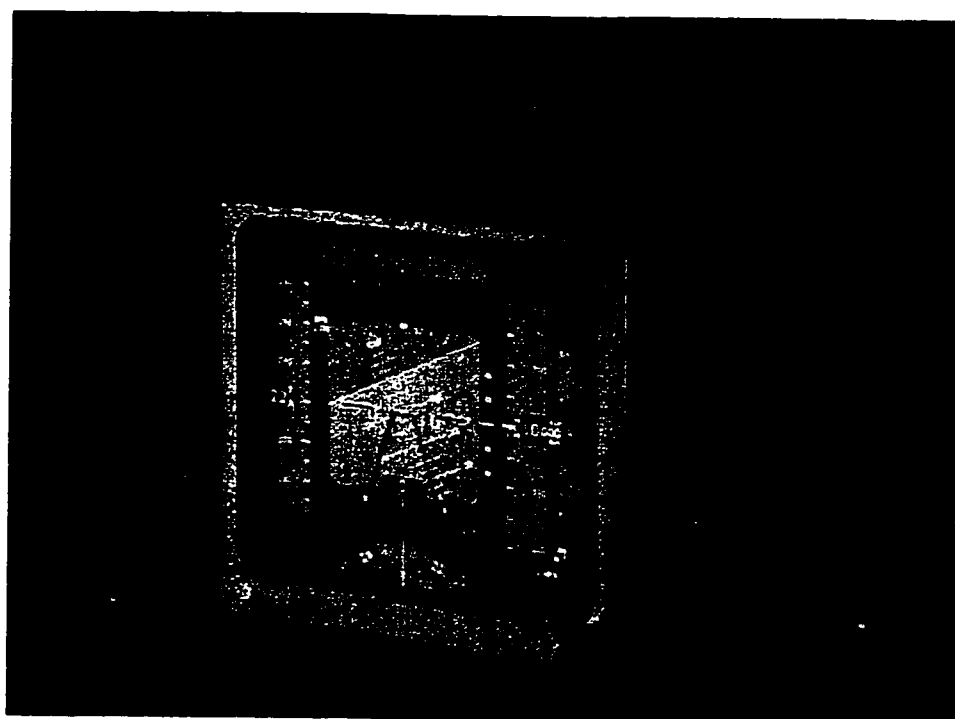


Figure 3.4 Actual EFIS display. [3.5]

3.4.2 Map Display

A graphical map of the surrounding area, shown in Figure 3.5, offers the means to geographically track the aircraft's position. In the main portion of the screen is the Montreal area including the local airports and navigational aids. The aircraft is represented by a yellow square which moves along the map as it flies. Along the side of the screen, aircraft control, performance, attitude, and position variables are constantly updated. Above this data, an isometric representation of the aircraft displays its attitude and heading. Since two possible perspectives could be seen in the isometric display, the left wing is coloured red and the right green to indicate the correct perspective.

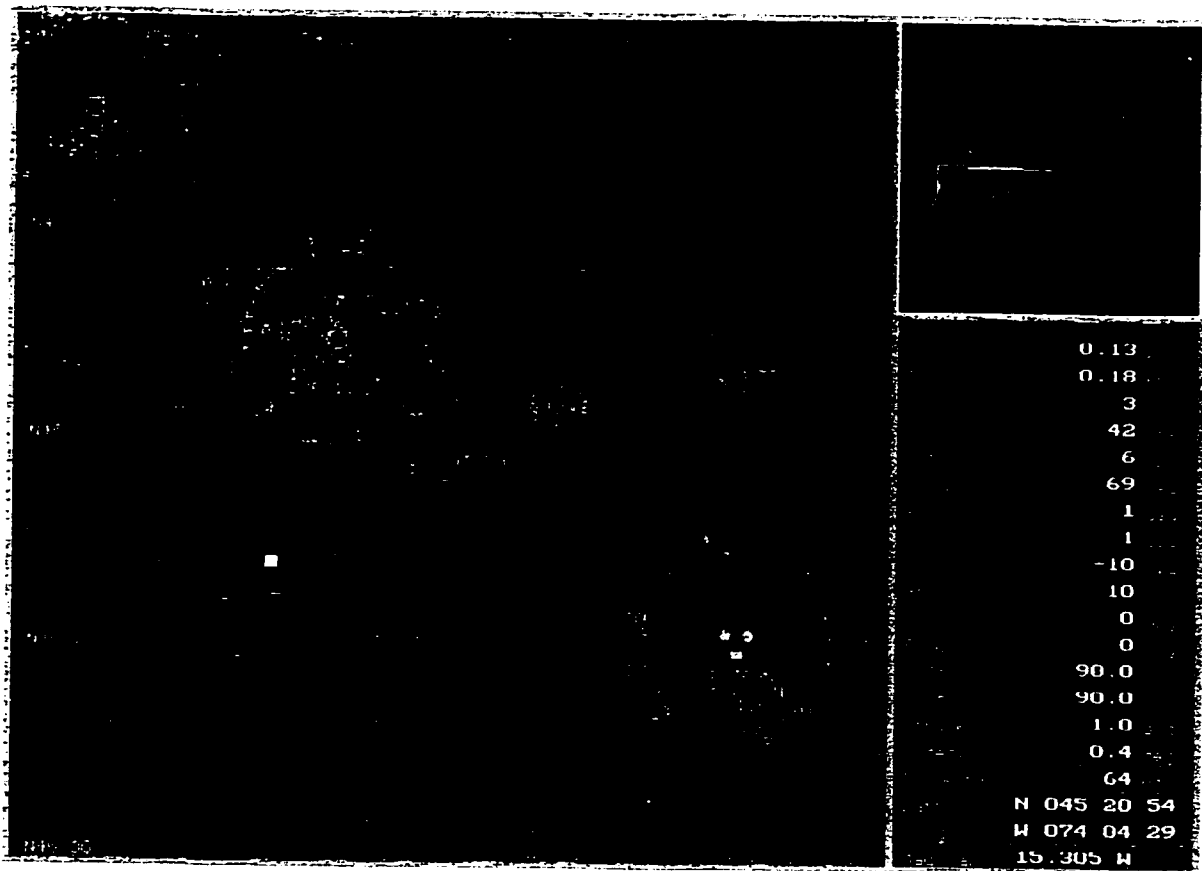


Figure 3.5 Map display.

3.4.3 Strip Chart Display

Another option displays two strip charts, shown in Figure 3.6. Up to eight parameters may be selected to display on either chart in a definable scale. The speed at which the charts are drawn can be set any time to draw a complete chart from left to right. When the graph has reached the end of the chart, it is erased and starts drawing again from the left. This does not store any data. If one wishes to store data, a data log file must be set up.

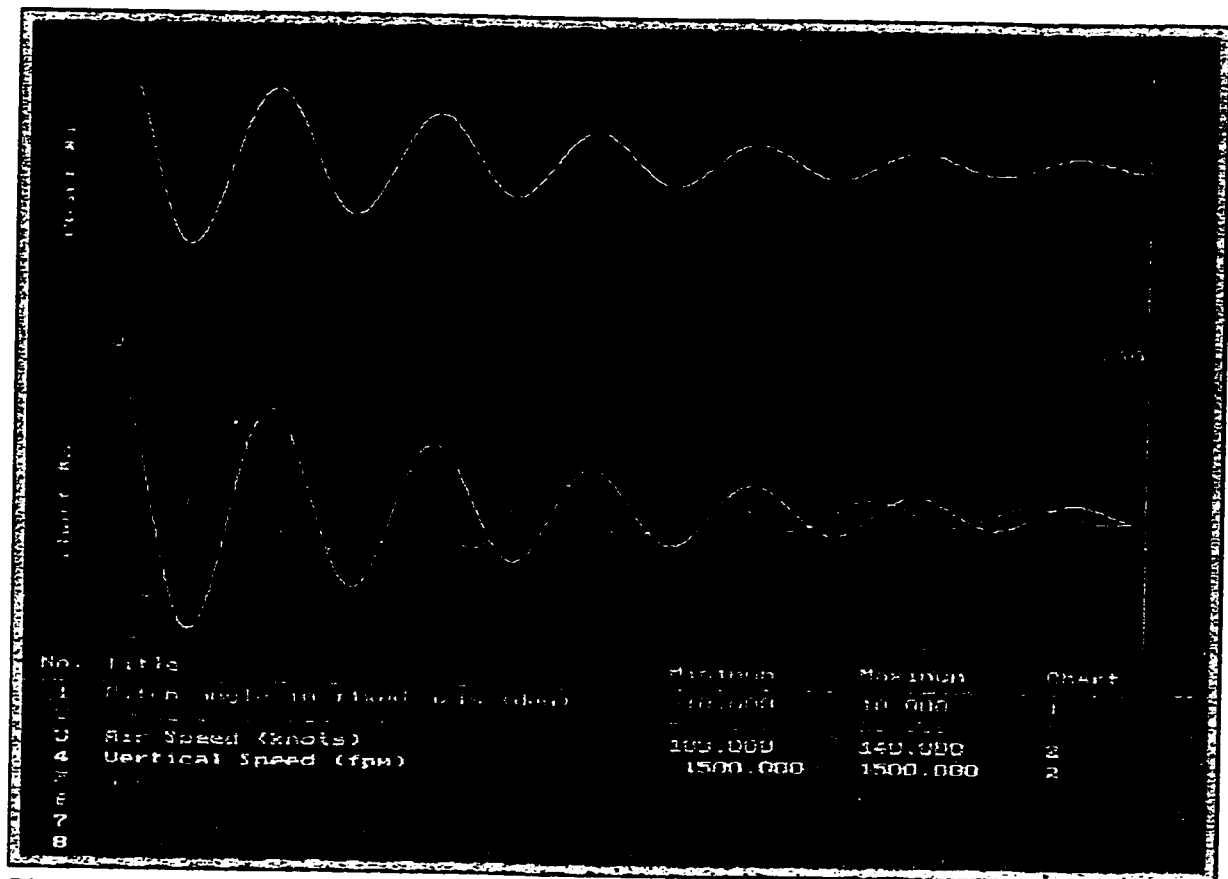


Figure 3.6 Strip chart recorder.

3.5 Optional Visual System

The program is capable of driving the current visual projection system which is based on a Silicon Graphics IRIS work station connected to the PC via an RS232 connection at 9600 baud. The visual system is selected by specifying the port to which it is connected: COM1, COM2, or not connected. Although no work was done on the visual system itself, the simulation code required to drive it was written.

3.6 Analysis Configuration

The program offers the option to study the flight model by setting up a data log file to record any variable at any frequency. The data is stored in a format which may be imported to most spreadsheet programs. In the past, this has been the simulator's primary use. However, modification and recompilation of the simulator code were required to encode the data in a 16 bit word which would be transmitted through a serial cable logged to an external terminal. The new systems offers greater performance because the setup is done through software, and as such, the file is saved directly on the system's own hard disk. The data is stored in its engineering units with appropriate headings and units, with little degradation of the iteration rate.

4.0 AIRCRAFT FLIGHT INSTRUMENTS, OPERATION AND SIMULATION

The cockpit section shown in figure 4.1, built from an actual Beech Duchess 76 aircraft is an accurate recreation of the cockpit environment. All the instruments and controls found in the cockpit section have been refitted with the necessary hardware to drive the instruments in a realistic manner and the software has been written to simulate the functioning of the actual instruments, including any errors or particular behaviour they might have.

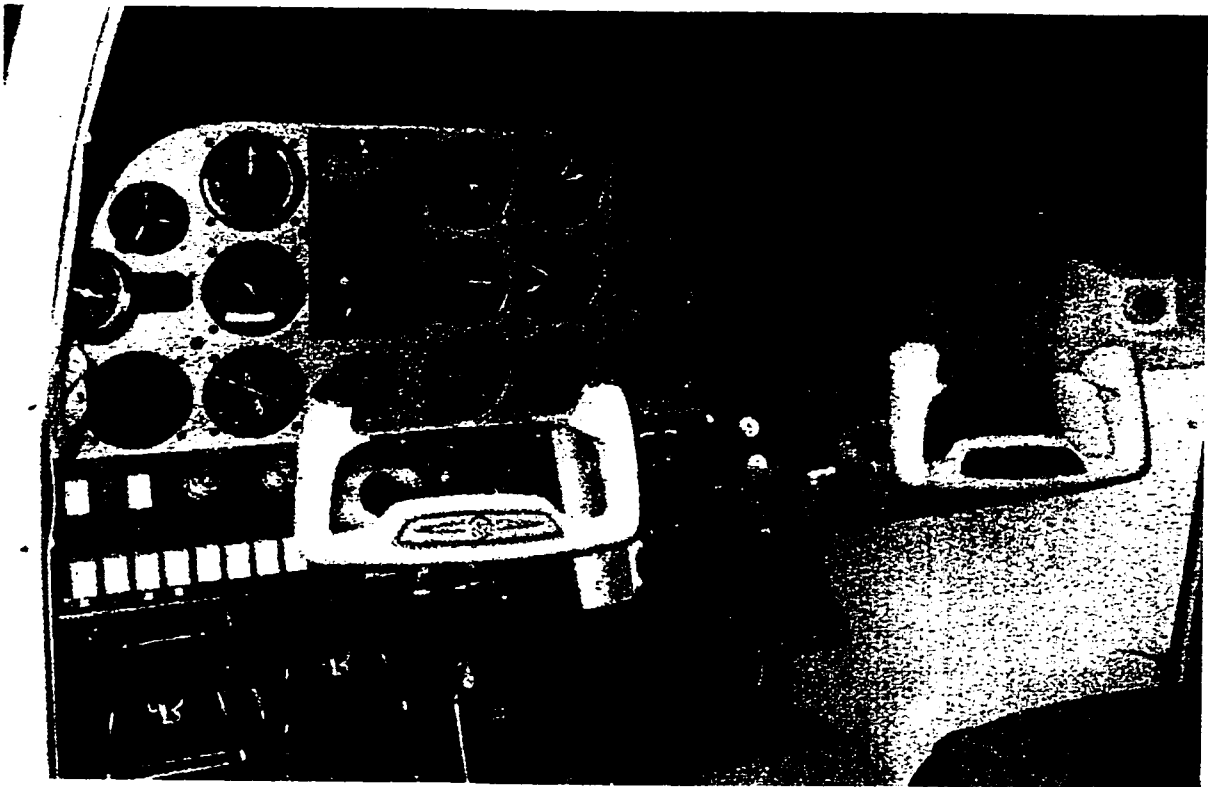


Figure 4.1 Simulated cockpit section.

4.1 The Magnetic Compass

The magnetic compass provides the pilot with a reliable self contained means of orientation by indicating the aircraft's heading relative to the Earth's magnetic north. This simplicity and reliability have made the magnetic compass a legal requirement on all aircraft in Canada [4.1][4.2].

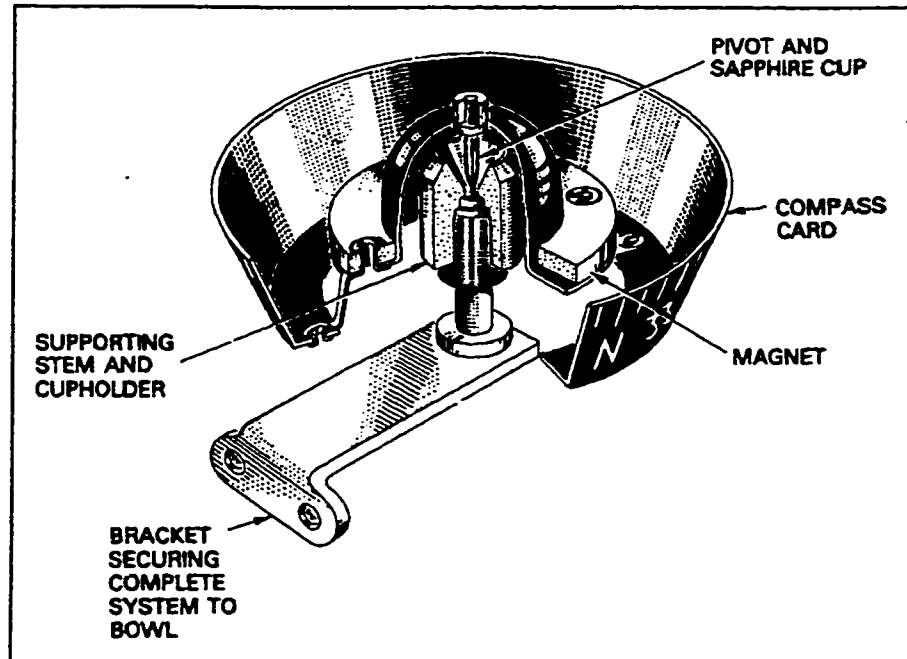


Figure 4.2 Magnetic compass construction. [4.3]

As shown in Figure 4.2, the magnetic compass consists of a permanent magnet which is suspended from a jewelled pivot and attached to a graduated card. The entire assembly is immersed in a sealed, fluid filled glass bowl. The fluid acts as a damping fluid and reduces the wear on the pivot by giving the magnet and compass card some buoyancy. Alcohol is generally used due to its low freezing point, low viscosity, resistance to corrosion, and its consistent clarity. [4.3]

4.1.1 Magnetic Compass Operation and Simulation

The magnetic compass provides the pilot with directional information by aligning itself with the Earth's magnetic north to indicate the aircraft's heading relative to the magnetic north, which does not coincide with the Earth's true north pole of rotation. Also, the Earth's magnetic field is constantly shifting due largely to changes in the Earth's conducting fluid outer core [4.4]. To understand the errors inherent to a magnetic compass, the Earth's magnetic field must be understood.

4.1.1.1 The Earth's Magnetic Field

Figure 4.3 depicts the Earth's magnetic field. Due mainly to variation in the Earth's composition, it does not follow a simple mathematically derivable path and must therefore be routinely measured and updated. The International Geomagnetic Reference Field (IGRF) is the mathematical model presently accepted by the International Union of Geodesy and Geophysics to describe the field potential. This model is updated every five years using the polar vector Equation 4.1 [4.5].

$$\bar{V}(r, \theta, \phi) = R_E \sum_{n=1}^N \left(\frac{R_E}{r} \right)^{n+1} \sum_{m=0}^n \left(g_n^m \cos(m \phi) + h_n^m \sin(m \phi) \right) P_n^m(\theta) \quad (4.1)$$

The notation used in this model is standard to this model and should not be confused with notation used to describe the aircraft's dynamics in later chapters.

r	radius of the point of interest from the centre of the earth
θ	longitude eastward from Greenwich
ϕ	colatitude ($\phi = 90^\circ - \text{latitude}$) 0° at true north, 180° at south pole
R_E	Earth's radius (6371.2 km)
N	number of coefficients g and h

- g Gauss coefficient
h Gauss coefficient
P Legendre polynomial of degree n and order m

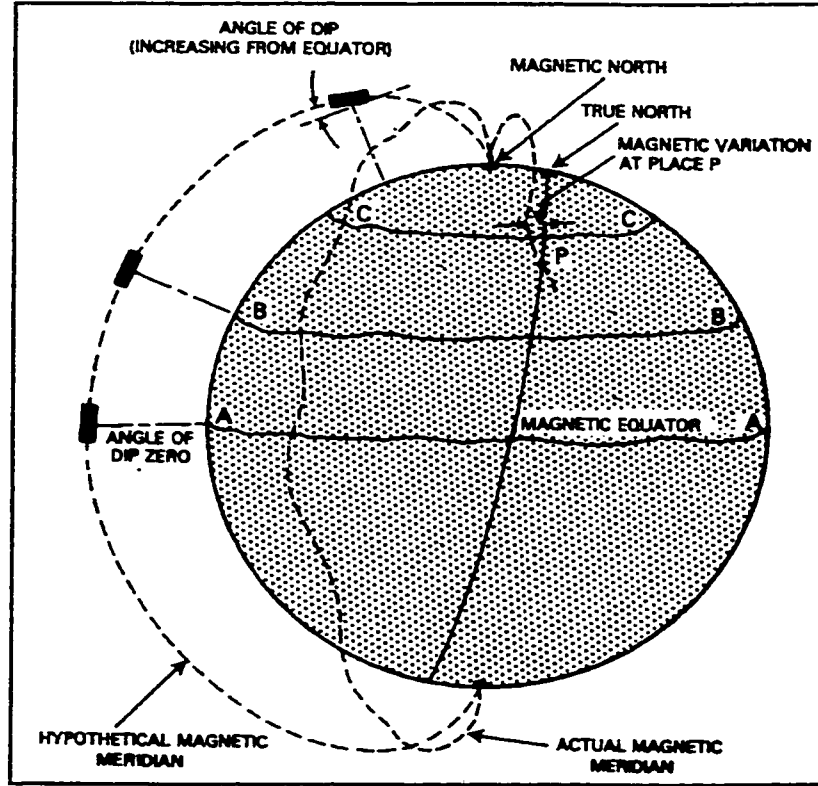


Figure 4.3 The Earth's magnetic field. [4.3]

Reference 4.4 gives a numerical solution to this equation for the field potential in three dimensions :

$$B_r(r, \theta, \phi) = -\frac{\partial V(r, \theta, \phi)}{\partial r} \quad (4.2)$$

$$B_r(r, \theta, \phi) = \sum_{n=1}^N (n+1) \left(\frac{R_E}{r} \right)^{n+2} \sum_{m=0}^n \left(g_n^m \cos(m\phi) + h_n^m \sin(m\phi) \right) P_n^m(\theta)$$

$$B_{\theta}(r, \theta, \phi) = -\frac{1}{r} \frac{\partial V(r, \theta, \phi)}{\partial \theta} \quad (4.3)$$

$$B_{\theta}(r, \theta, \phi) = \sum_{n=1}^N \left(\frac{R_E}{r} \right)^{n+2} \sum_{M=0}^n \left(g_n^m \cos(m \phi) + h_n^m \sin(m \phi) \right) \frac{dP_n^m(\theta)}{d\theta}$$

$$B_{\phi}(r, \theta, \phi) = -\frac{1}{r \sin \theta} \frac{\partial V(r, \theta, \phi)}{\partial \phi} \quad (4.4)$$

$$B_{\phi}(r, \theta, \phi) = \frac{1}{\sin \theta} \sum_{n=1}^N \left(\frac{R_E}{r} \right)^{n+2} \sum_{M=0}^n m \left(g_n^m \sin(m \phi) - h_n^m \cos(m \phi) \right) P_n^m(\theta)$$

The coefficients g and h are updated every five years and may be down-loaded from the Flight Dynamics Division of NASA's Goddard Space Flight Center [4.6]. The Legendre polynomial and its derivatives are computed using Equations 4.5 to 4.17.

$$P_n^m(\theta) = S^{nm} P^{nm}(\theta) \quad (4.5)$$

$$\frac{dP_n^m(\theta)}{d\theta} = S^{nm} \frac{dP^{nm}(\theta)}{d\theta} \quad (4.6)$$

$$P^{-10}(\theta) = \frac{dP^{-10}(\theta)}{d\theta} = 0 \quad (4.7)$$

$$P^{00}(\theta) = 1 \quad (4.8)$$

for $m=n \neq 0$

$$P^{nm}(\theta) = \sin(\theta) P^{(n-1)(m-1)}(\theta) \quad (4.9)$$

for $m \neq n, n > 0$

$$P^{nm}(\theta) = \cos(\theta) P^{(n-1)m} - k^{nm} P^{(n-2)m}(\theta) \quad (4.10)$$

for $m=n \neq 0$

$$\frac{dP^{nm}(\theta)}{d\theta} = \sin(\theta) \frac{dP^{(n-1)(m-1)}(\theta)}{d\theta} + \cos(\theta) P^{(n-1)(m-1)}(\theta) \quad (4.11)$$

for $m \neq n, n > 0$

$$\frac{dP^{nm}(\theta)}{d\theta} = \cos(\theta) \frac{dP^{(n-1)m}(\theta)}{d\theta} - \sin(\theta) P^{(n-1)m}(\theta) - k^{nm} \frac{dP^{(n-2)m}}{d\theta} \quad (4.12)$$

$$S^{00} = 1 \quad (4.13)$$

for $n > 0$

$$S^{n0} = \left(\frac{2n-1}{n} \right) S^{(n-1)0} \quad (4.14)$$

$$S^{n1} = \sqrt{\frac{2n}{n+1}} \quad (4.15)$$

for $m > 1$

$$S^{nm} = \sqrt{\frac{(n-m+1)J}{n+m}} S^{n(m-1)} \quad (4.16)$$

$$k^{nm} = \frac{(n - 1)^2 - m^2}{(2n - 1)(2n - 3)} \quad (4.17)$$

With the field potential in the longitudinal and lateral directions, it is possible to determine angular variation of the magnetic north relative to the true north. This quantity, called the magnetic variation, is found using Equation 4.18.

$$\text{Magnetic Variation} = \tan^{-1} \left(\frac{B_{\phi}}{B_{\theta}} \right) \quad (4.18)$$

The inclination of the field relative to a tangent to the Earth, call the magnetic dip, is found using Equation 4.19.

$$\text{Magnetic Dip} = \tan^{-1} \left(\frac{B_r}{\sqrt{B_{\phi}^2 + B_{\theta}^2}} \right) \quad (4.19)$$

Appendix 9 lists the routine used to calculate the magnetic variation.

4.1.1.2 Magnetic Compass Acceleration and Turning Errors

The radial component of the Earth's field will tend to tip the compass card off the horizontal plane. To minimize this effect and to give it added stability, the compass card is designed such that the centre of gravity is located below the pivot point. This gives rise to errors caused by the acceleration of the aircraft due to both changes in the forward velocity as well as lateral accelerations resulting primarily from aircraft turning [4.3]. Figure 4.4 is a schematic representation of a magnetic compass. The moment caused by the radial component of the Earth's field is balanced by the moment from the slight deflection of the centre of gravity.

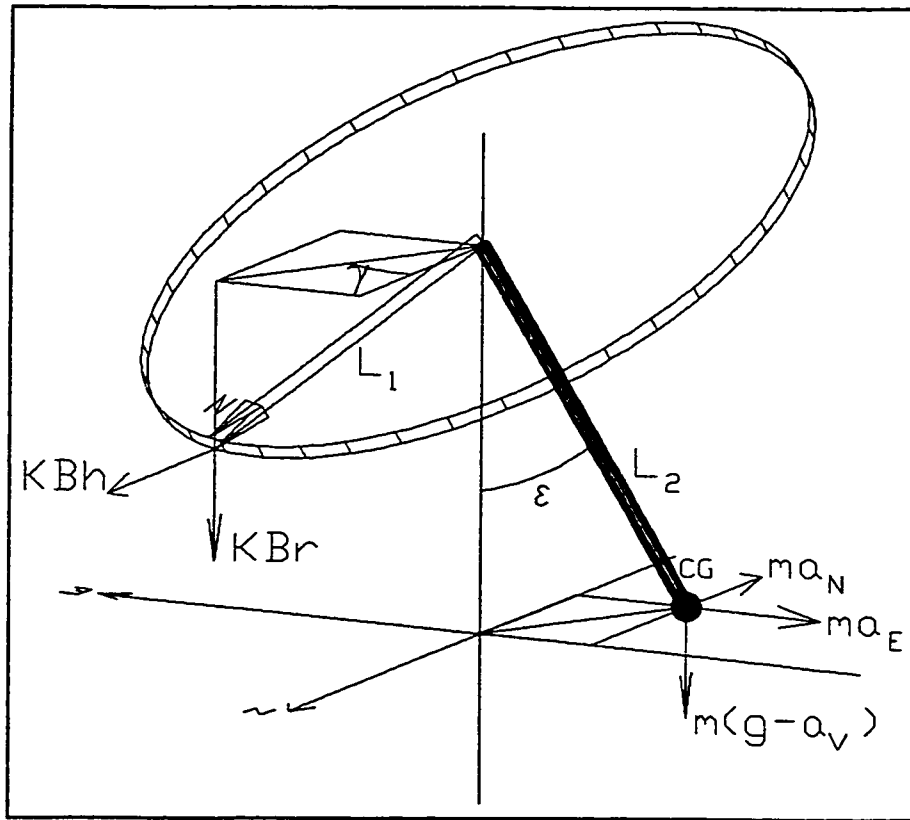


Figure 4.4 Schematic representation of a magnetic compass.

The disk represents the compass card with the centre of mass at the end of a pendulum fixed at its centre. The disk and pendulum assembly is free to pivot and dip about the point where they are joined. Acceleration forces act at the centre of gravity and the forces due to the Earth's magnetic field act at the North pointing point on the disk, causing a dip of angle ϵ and an heading error γ . Taking the moments about the pivot gives Equations 4.20 to 4.22.

$$KB_r L_1 \cos \epsilon \cos \gamma - KB_h L_1 \sin \epsilon = m(g - a_v) L_2 \sin \epsilon \cos \gamma - m a_N L_2 \cos \epsilon \quad (4.20)$$

$$KB_h L_1 \cos \epsilon \sin \gamma = m a_E L_2 \sin \epsilon \cos \gamma - m a_N L_2 \sin \epsilon \sin \gamma \quad (4.21)$$

$$KB_r L_1 \cos \epsilon \sin \gamma = m(g - a_v) L_2 \sin \epsilon \sin \gamma - m a_E L_2 \cos \epsilon \quad (4.22)$$

These equation are not solvable by a closed form solution and would require

iterative numerical techniques. Since acceleration errors are not quantitatively read by pilots, but rather qualitatively taken into account when reading the compass, it was decided that an exact solution was not required. Rather than implementing an iterative solution into the simulation, a rough approximation, tuned by qualitative observations in flight, was implemented. This model breaks the error into a component due to acceleration and a component due to turning. The result is in degrees and the acceleration is in knots/second.

$$\text{Mag Compass Indication} = \text{Heading}_{\text{Mag}} + \text{Err}_{\text{Acc}} + \text{Err}_{\text{Turn}} \quad (4.23)$$

$$\text{Err}_{\text{Acc}} = -5 \sin(\psi_{\text{Mag}}) \sin(\text{Dip}) \text{ Acceleration} \quad (4.24)$$

$$\text{Err}_{\text{Turn}} = -5g \cos(\psi_{\text{Mag}}) \sin(\text{Dip}) \sin(\phi) \quad (4.25)$$

4.2 The Air Speed Indicator

The air speed indicator (ASI) is a critical instrument since the safe operation of the aircraft depends on specific maximum and minimum indicated air speeds. Some examples of this are the maximum landing gear extension speed, maximum flap extension speed, speed for best rate of climb and best angle of climb.

4.2.1 ASI Operation

The air speed indicator (ASI) shown in Figure 4.5 is a typical representation of one found in a general aviation aircraft. It actually provides an indication of the dynamic pressure acting on the aircraft. This is more useful in determining the aircraft's performance since, as was presented in Chapter 2, the forces and moments acting on the aircraft depend directly on the dynamic pressure.

The ASI operates by indicating the pressure difference between the total pressure measured by a pitot probe mounted in the free stream airflow around the aircraft and the static pressure measured at a static port, usually located on the fuselage, as shown in Figure 4.6. The scale on the instrument is calibrated to indicate the airspeed at standard conditions (101.3 kPa and 15°C), and to take into account the compressibility of air at the stagnation point of the pitot tube at higher velocities. Equation 4.26 is used [4.3]

$$P_{tot} - P_{stat} = \frac{1}{2} \rho_o V_{cal}^2 \left(1 + \frac{V_{cal}^2}{4 a_o^2} \right) \quad (4.26)$$

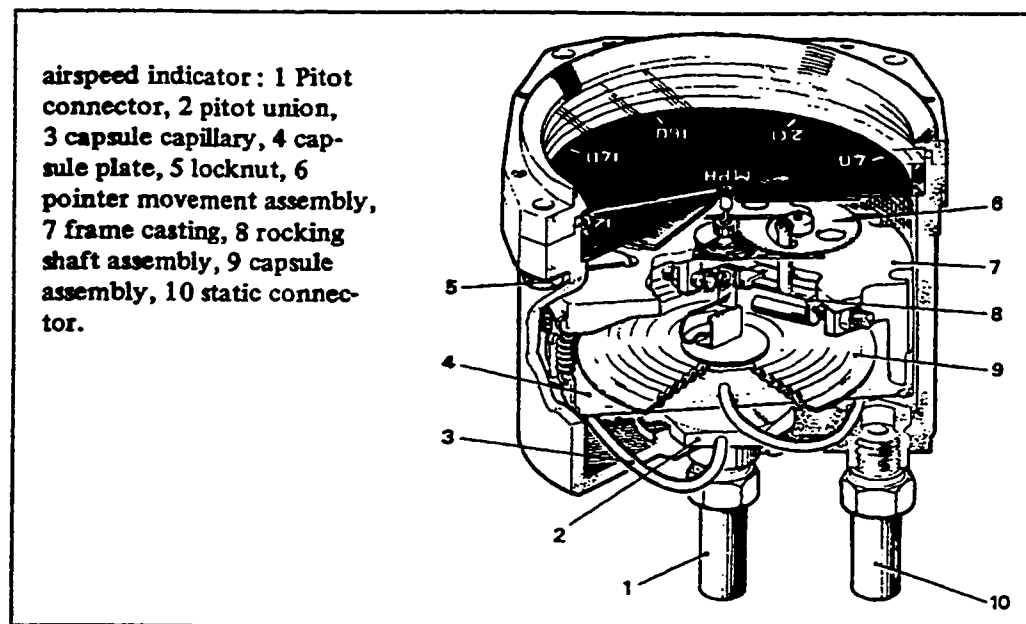


Figure 4.5 Air speed indicator, cutaway view. [4.3]

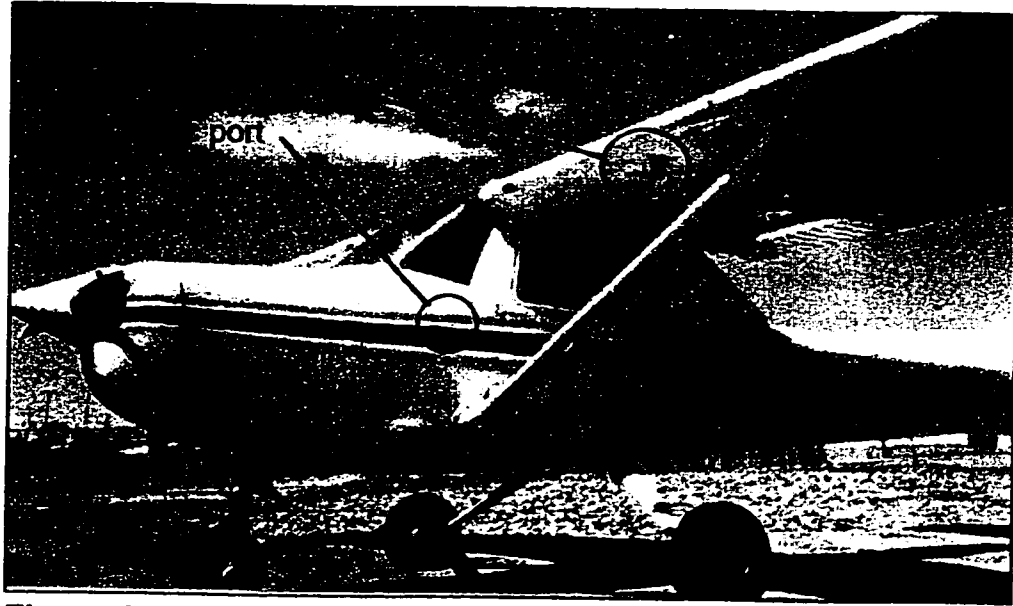


Figure 4.6 Pitot tube and static port location.

At low airspeed, the generally accepted assumption is that air is incompressible, and as such Equation 4.26 reduces to Equation 4.27.

$$V_{cal} = \sqrt{\frac{2}{\rho_o}(P_{tot} - P_{stat})} \quad (4.27)$$

If the aircraft is operating in an air density other than standard, then this calibrated airspeed would differ from the true airspeed. To determine the calibrated airspeed, the corresponding pressure difference between the pitot and static ports are found using Equation 4.28 with the true airspeed, local air density and speed of sound at the aircraft altitude. Equation 4.27 is then used to determine the calibrated airspeed.

$$P_{tot} - P_{stat} = \frac{1}{2}\rho V_{true}^2 \left(1 + \frac{V_{true}^2}{4a_0^2} \right) \quad (4.28)$$

If the assumption is made that air is incompressible then the true and calibrated airspeed are related by the Equation 4.29.

$$V_{true} = V_{cal} \sqrt{\frac{\rho_o}{\rho}} \quad (4.29)$$

Due to the installation of the pitot static system on the aircraft, it is affected by disturbances in the flow due to effects such as flap deployment and propeller wash. These effects are not easily predicted, and are typically small. Experimental results are often used to create a graph or table relating the indicated airspeed to the calibrated airspeed. Figure 4.7 is a graph relating the Duchess' indicated to calibrated airspeed.

4.2.2 ASI Simulation

In the simulation, the flight model calculates the aircraft's true velocity. The indicated airspeed is determined in three steps. First the difference between the pitot and static pressures is found using the true velocity and local air density. Secondly, this pressure difference is used to calculate the calibrated airspeed. The experimental results are then used to obtain the indicated airspeed from the calibrated air speed.

Having determined the indicated airspeed, a final look up table is used to determine angular position of the airspeed indicator to show the correct indicated airspeed. Note that a minimum airspeed is required to initially move the needle. The position of the simulated needle is driven by a brushless DC motor as discussed in Section 6.1.4.1.1. Table 4.1 gives the angular displacement from the 40 knot position as a function of the indicated airspeed. The motor was mounted in the simulated instrument such that a zero degree displacement of the pointer corresponds to a 40 knot indicated airspeed. This data is used in the simulation to determine the required position of the needle.

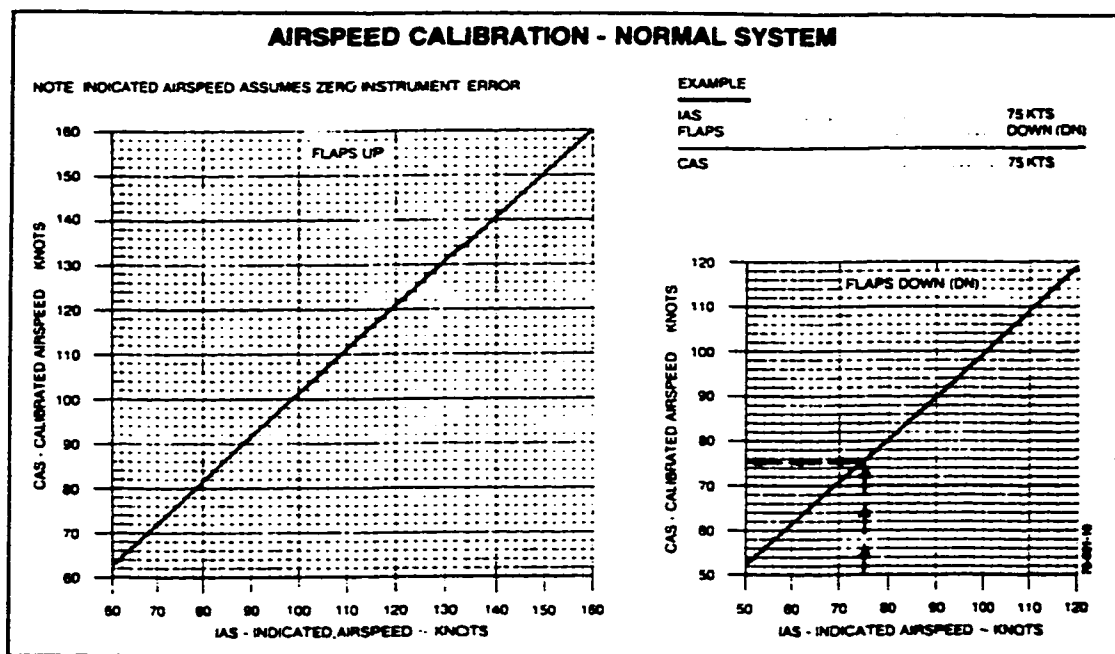


Figure 4.7 Beech Duchess indicated to calibrated airspeed conversion. [4.7]

CIAS	deg	CIAS	deg	CIAS	deg	CIAS	deg	CIAS	deg
40	0.0	90	70.7	140	168.2	190	254.0	240	305.9
45	4.7	95	80.2	145	178.3	195	258.9	245	310.4
50	9.5	100	89.9	150	189.0	200	264.1	250	314.9
55	15.8	105	99.5	155	198.9	205	269.1	255	319.8
60	22.5	110	109.2	160	209.9	210	275.4	260	324.7
65	30.4	115	119.2	165	218.9	215	280.7		
70	37.5	120	129.1	170	229.6	220	286.0		
75	45.9	125	139.3	175	236.5	225	291.5		
80	52.5	130	148.5	180	243.7	230	296.5		
85	61.6	135	158.4	185	248.0	235	301.2		

Table 4.1 Airspeed indicator indicated airspeed vs angular position.

4.3 The Attitude Indicator

The attitude indicator, shown in Figure 4.9 provides the pilot with a representation of the aircraft's pitch and roll attitude. The two fixed horizontal lines to the left and right of the instrument represent the aircraft's wings. Behind them is a region with the lower half coloured brown, and the upper half coloured blue representing the ground and sky respectively.

As the aircraft rolls about its longitudinal axis, the background of the indicator rolls by the same amount, thus representing the aircraft roll. From the pilots point of view, the horizontal line which divides the brown and blue sections will always remain parallel to the horizon as seen out of the aircraft's windshield. The amount by which the aircraft rolls is indicated by an arrow at the top of the instruments, which rotates with the background, pointing to

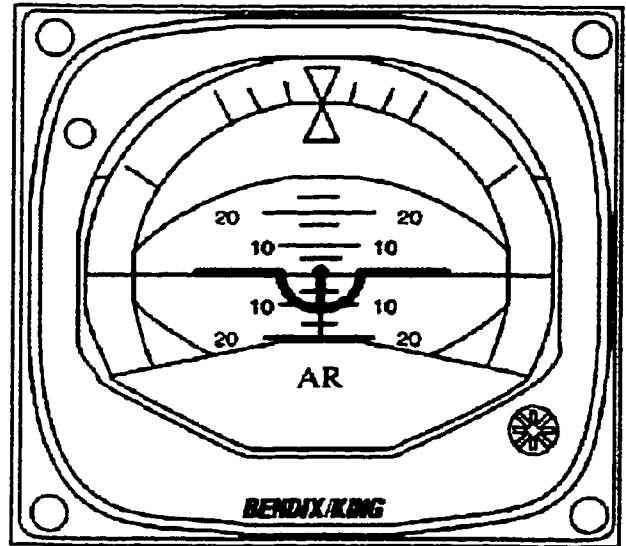


Figure 4.9 Attitude indicator. [4.8]

graduations along the perimeter at plus and minus 10, 20, 30, and 60 degrees of roll. As the aircraft pitches about its lateral axis, the background rises and descends in a manner similar to the horizon seen beyond the windshield. The degree to which the aircraft pitches is indicated by graduations on the background, which is read at the dot located in the centre of instrument.

4.3.1 Attitude Indicator Operation

The attitude indicator shown in a cutaway view in Figure 4.10 is driven by a gyroscope. Through a series of linkages, the gyroscope, which tends to maintain its orientation in space, drives the background in both pitch and roll according to the movements of the aircraft.

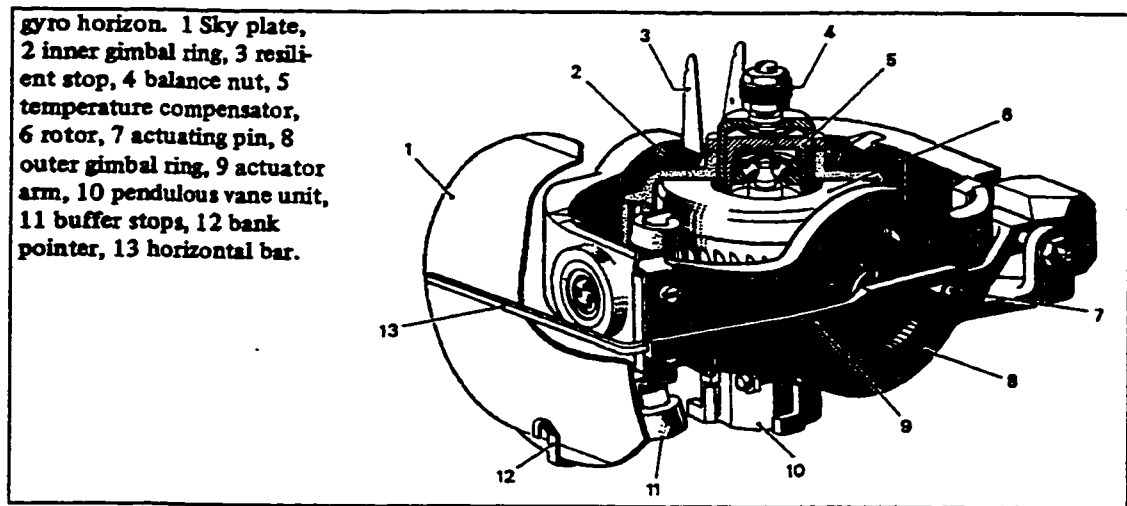


Figure 4.10 Attitude indicator construction.

4.3.2 Attitude Indicator Simulation

The construction of the simulated attitude indicator is shown in Figure 4.11. Since the brushless DC motors used in the other instruments lack the torque to reliably drive the mechanisms, stronger stepper motors are used. Both stepper motors have a 7.5 degree step and are mounted with a reduction gear head to reduce the step size. To gain a greater resolution between steps, micro stepping techniques are employed as described in Section 6.1.4.1.3.

The roll axis is driven by its own dedicated stepper motor. Since the output from the gear head on the roll motor is 1 degree per step which drives a 1 to 6 gear ratio to roll the background, the net effect is a 1/6 degree indicated roll per step. Due to the differential mechanism within the roll axis, the indicated pitch is a

combination of the output from both the pitch and the roll motors. Considering at first pure pitching motion, the relationship between the pitch stepper motor and the indicated pitch is 20/56 degree indicated pitch per step. The effect of the roll through the differential must also be taken into account by subtracting four steps for each degree of roll.

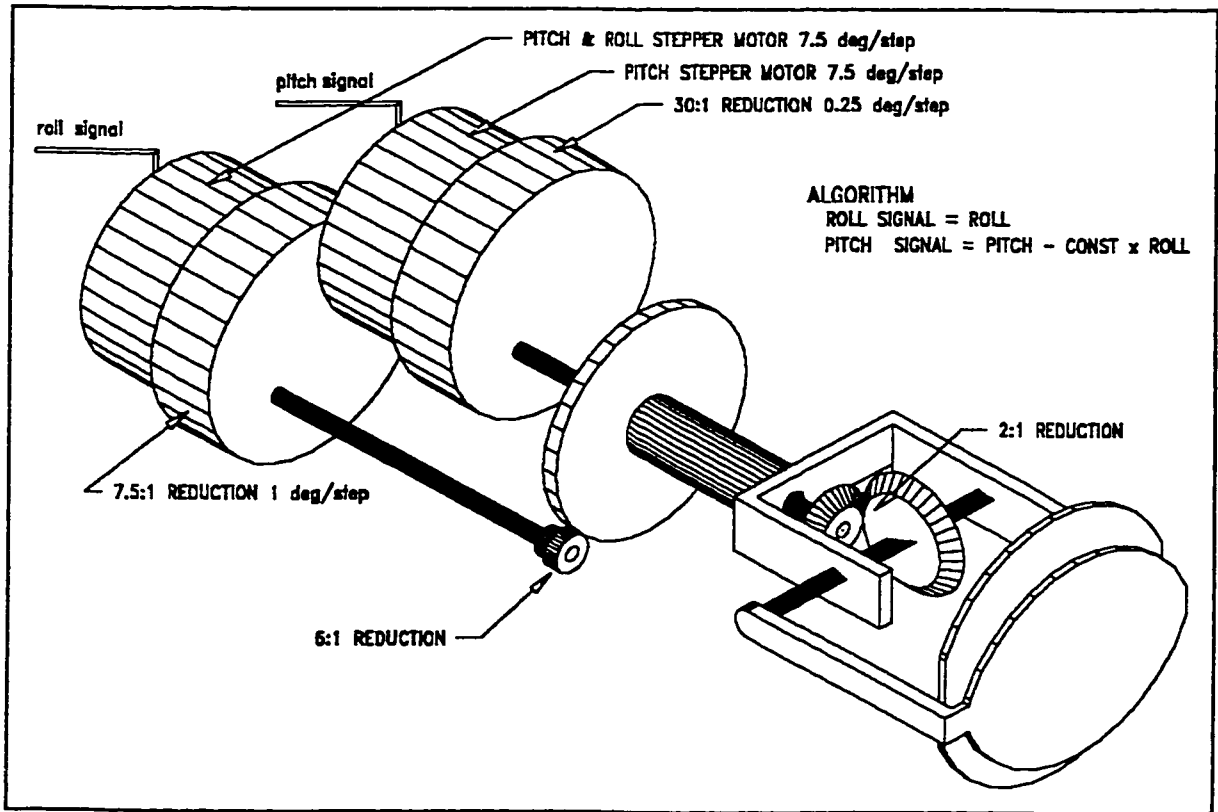


Figure 4.11 Simulated attitude indicator construction.

4.4 The Altimeter

The altimeter provides the pilot with an indication of the aircraft's altitude above sea level. This is useful not only from the point of view of maintaining the aircraft at a specific altitude for ground clearance and aircraft separation, but also to determine aircraft performance which can vary significantly due to the changing properties of the atmosphere with altitude.

4.4.1 Altimeter Operation

Shown as an exploded view in Figure 4.12, the altimeter consists of a series of aneroid capsules which expand and contract as a function of the atmospheric pressure. Through a series of linkages, this motion is transferred to a linear scale where hundreds of feet are indicated by the long pointer, thousands by the medium pointer and ten thousands by the short pointer. Table 4.2 lists the normal error limits for the indicated altitude. Since this instrument is pressure sensitive, a means to correct for varying atmospheric pressure due to the weather is provided. On the front of the instrument is a baroscale adjustment knob by which the pilot can set the sea level pressure. With this parameter set, the altimeter will indicate the correct altitude above sea level based on a standard atmosphere. Pilots refer to this altitude as the indicated altitude.

Test point (ft/1000)	0	2	4	6	8	10	12	14	16	18	20	22	24	28	32	35
Error limit (ft/100)	1	1	1	2	2	2	2	3	3	3	3	4	5	6	7	8

Table 4.2 Simple altimeter error limits. [4.9]

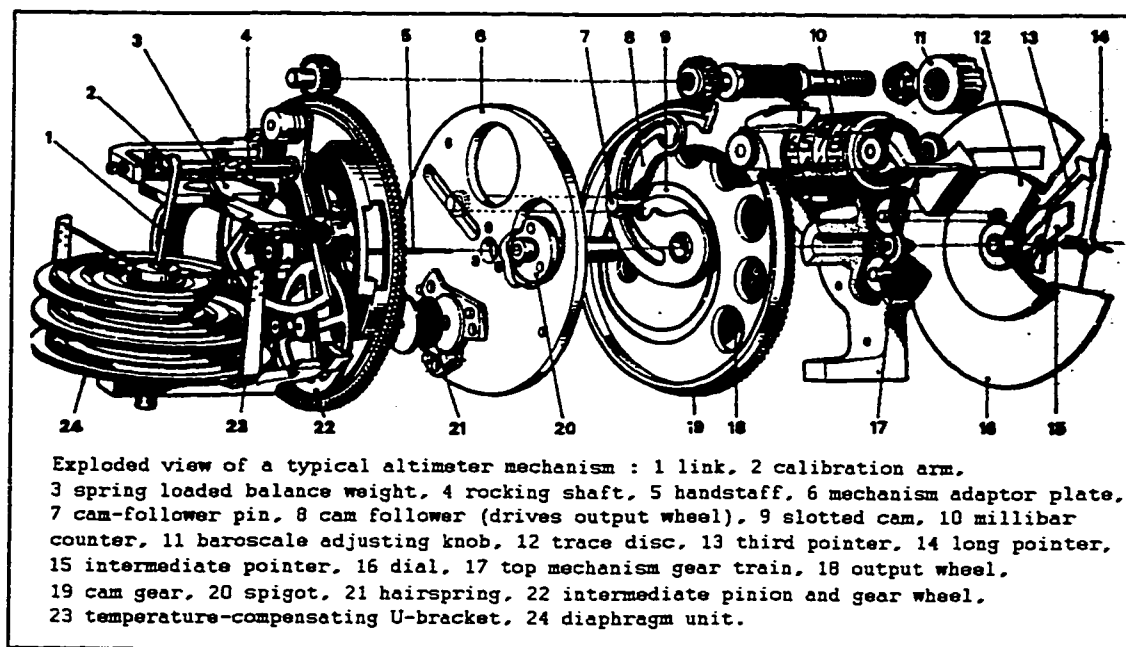


Figure 4.12 Exploded view of a typical altimeter. [4.3]

If the altimeter is set to the standard sea level pressure of 101.3 kPa, the altimeter would display the altitude in standard conditions. Pilots refer to this altitude as the pressure altitude [4.1]. The altimeter is designed to display the correct altitude assuming the temperature decreases linearly at the standard lapse rate of 6.5 °C/1000 m (1.98 °C/1000 ft). The indicated altitude corrected to sea level pressure, may be found by referring to the equations describing the standard atmosphere in Section 2.6, and solving for the altitude. The result is Equation 4.30.

$$Z_{ind} = \frac{T_{sl} \left(\frac{P}{P_{sl}} - 1 \right)^{\frac{1}{5.2561}}}{0.0065^{\circ}K/m} \quad (4.30)$$

Recall that according to convention, a negative altitude is in the upward direction.

Another interpretation of the density altitude is a pressure altitude which has the same density as the air in which the aircraft is flying.

4.4.2 Altimeter Simulation

The pointers on the simulated altimeter are driven through three coaxially mounted brushless DC motors as shown in Figure 4.13. With the result of Equation 4.30, angle to which each pointer turns is given by Equations 4.31 to 4.33.

$$P_{10000} = -\frac{Z_{ind}}{10000} 360^{\circ} \quad (4.31)$$

$$P_{1000} = -\frac{Z_{ind}}{1000} 360^{\circ} \quad (4.32)$$

$$P_{100} = -\frac{Z_{ind}}{100} 360^\circ$$

(4.33)

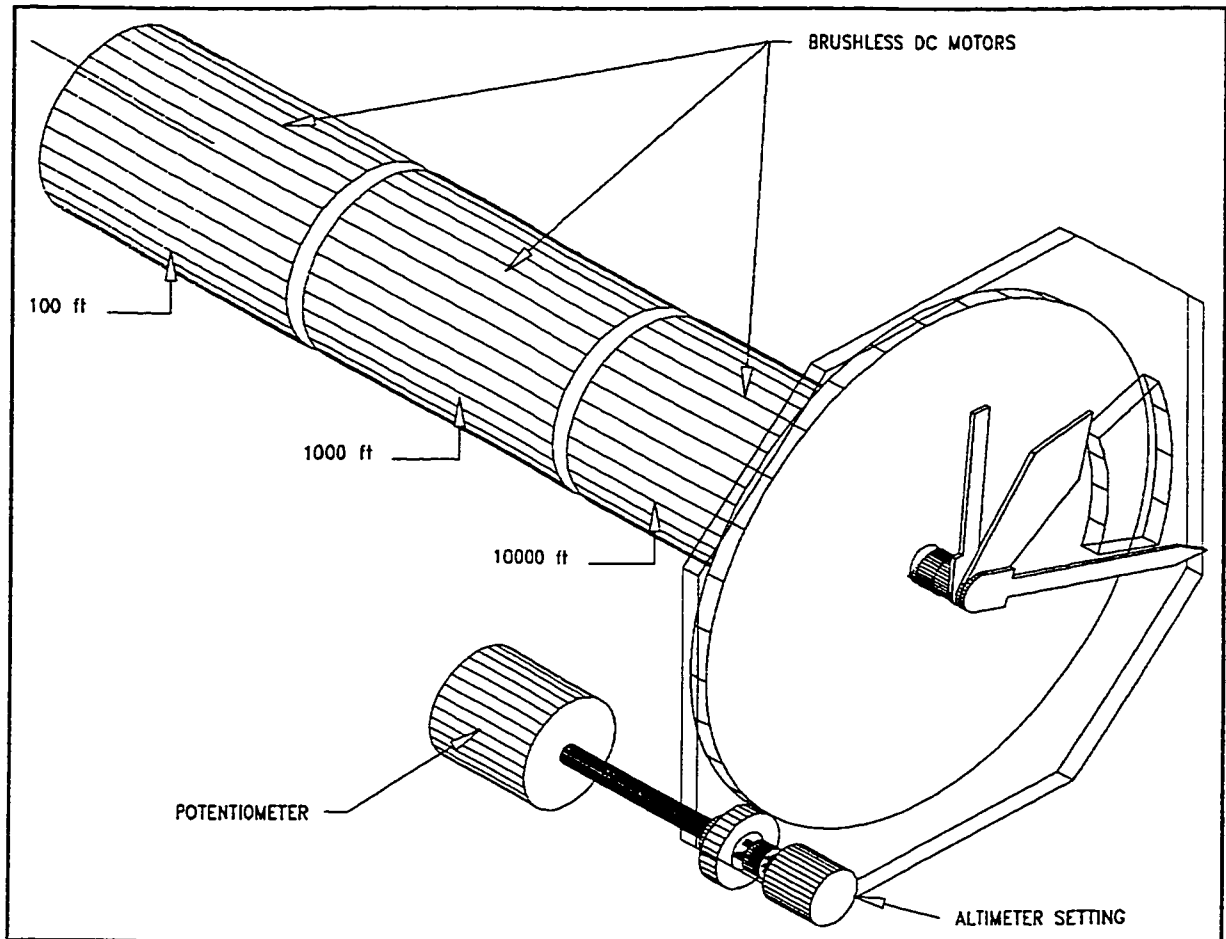


Figure 4.13 Simulated altimeter construction.

4.5 The Turn Coordinator

The turn coordinator, shown in figure 4.14 provides the pilot with an indication of the aircraft's turning yaw rate as well as a slip or skid indication. The display consists of a representation of an aircraft which banks in the direction of the turn. The greater the rate of turn, the greater the angle of bank. Horizontal markings on the side of the instrument line up with the wings when the aircraft is not turning. A second set of marking align with the left or right wing when the aircraft is turning at a rate that would come full circle in two minutes. Below the aircraft is a black ball mounted in a curved tube against a white background. If the aircraft is performing a coordinated turn then the net force due to centripetal and gravitational acceleration would be in the body axis vertical direction, centering the ball. If the aircraft is skidding out of the turn or slipping into the turn then the ball will slide to one side or the other.

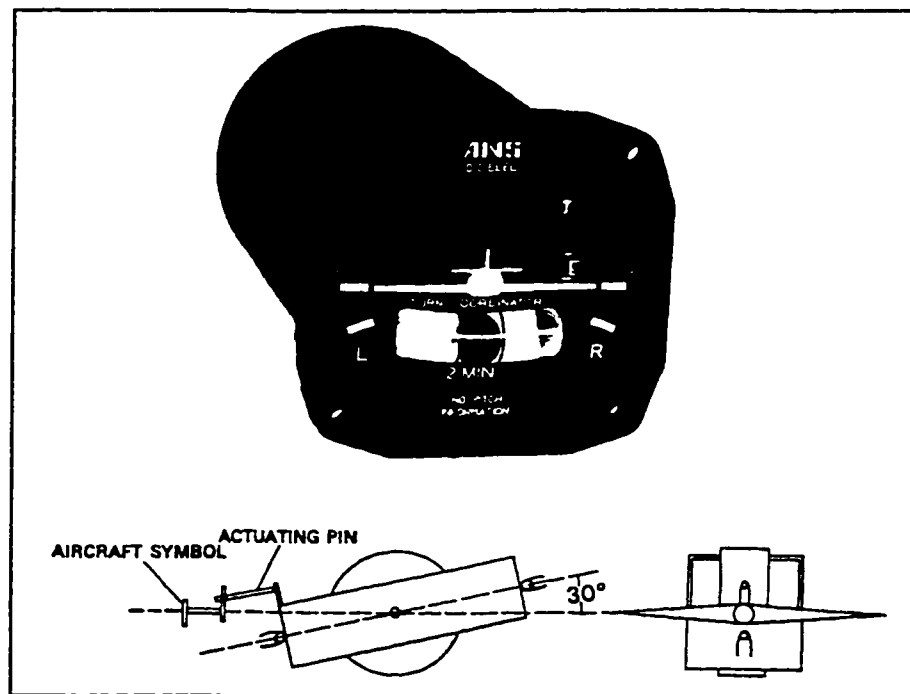


Figure 4.14 Turn coordinator. [4.3]

4.5.1 Turn Coordinator Operation

The turn coordinator contains a gyroscope with its axis of rotation mounted in the aircraft's lateral direction, in a frame which can rotate about an axis 30 degrees from the aircraft's longitudinal axis in the vertical plane. By mounting the frame at a 30 degree angle, the precession of the gyroscope will result in a torque about the frame's axis when subjected to a rotational velocity about an axis normal to the frame's axis. The rotational velocity about the frame's normal axis is due to the aircraft's roll and pitch velocity [4.3]. Respecting the sign conventions, the frames normal angular velocity is given by equation 4.34.

$$\omega_{TC_N} = p \sin(30) + q \cos(30) \quad (4.34)$$

A spring is used to apply a restoring moment to the frame's centre position against the gyroscope's precession. With the spring and precession torques in equilibrium, the representative aircraft will rotate to a position proportional to the aircraft's rate of turn by the amount given by Equation 4.35.

$$\phi_{TC_T} = K_{TC} \omega_{TC_N} \quad (4.35)$$

The value of K_{TC} is chosen such that, for a two minute turn, the indicator will line up with the two minute turn marks.

By tilting the frame's axis, the turn coordinator senses not only the yaw motion of the aircraft, but also the banking of the aircraft as it enters the turn, and as such provides an indication of this bank in the direction of the impending turn.

4.5.2 Turn Coordinator Simulation

The turn coordinator is simulated by implementing the equations which describe its motion. The deflection of the ball in the tube is proportional to the

inverse tangent of the aircraft's vertical and lateral acceleration. The amount by which it moves from side to side is proportional to the curvature of the tube. This deflection is given by the Equation 4.36.

$$\phi_{TC_s} = \tan^{-1}\left(\frac{Y}{Z}\right) \quad (4.36)$$

In the simulated instrument, both the aircraft symbol and ball are driven by a brushless DC motor.

4.6 The Directional Gyrocompass

The gyroscopic compass shown in Figure 4.15 provides the pilot with an indication of the aircraft's heading without the acceleration and turning errors inherent in the magnetic compass.

4.6.1 Gyroscopic Compass Operation

Based on a gyroscope which resists changes in its orientation, a gyroscopic compass can drift over time due to friction on the bearings and gimbal imbalance which can cause unwanted precession. This drift must be corrected at regular intervals. The correction can be made manually by referencing the magnetic compass during straight and steady flight, or automatically and continuously by a flux valve which electronically measures the Earth's magnetic field.

4.6.2 Gyroscopic Compass Simulation

In the simulation, the directional gyroscopic compass, included in the horizontal situation indicator (HSI) indicates the magnetic heading and if not slaved to the flux valve, a user defined drift will be added cumulatively to the indicated heading taken directly from the flight model.

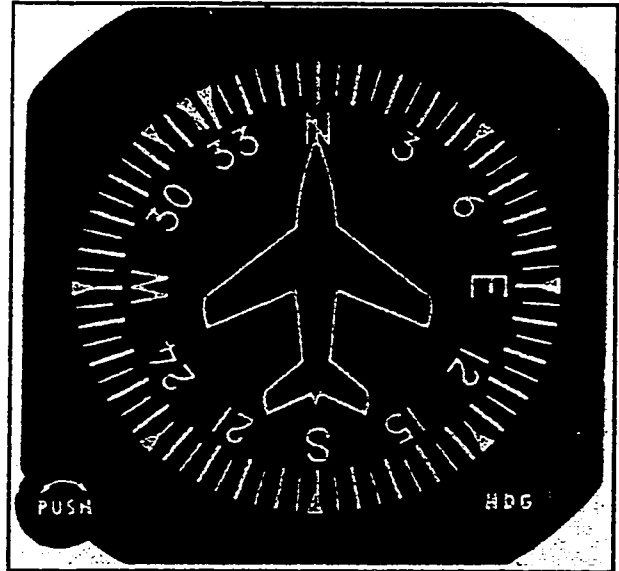


Figure 4.15 Gyroscopic compass. [4.10]

4.7 Vertical Speed Indicator

The vertical speed indicator (VSI) shown in Figure 4.16 provides the pilot with an indication of the aircraft vertical speed. It operates by measuring the rate of change of static pressure measured from the static port and translating it into a clockwise or counterclockwise rotation of the indicator.

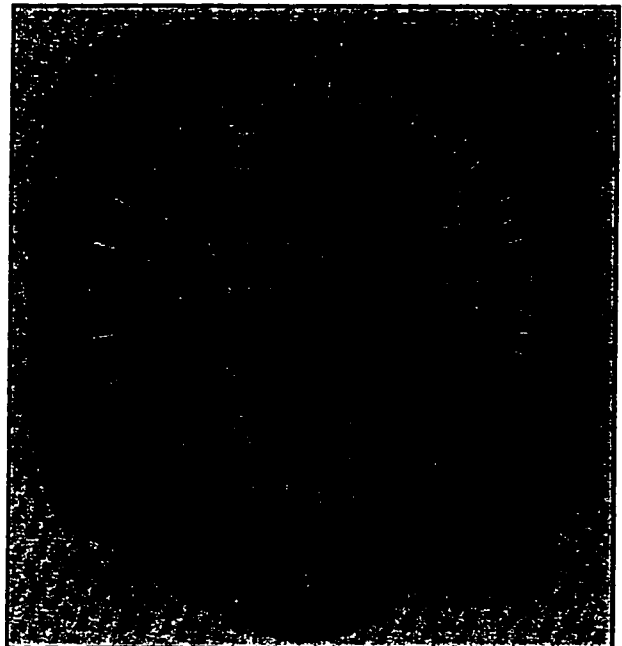


Figure 4.16 Vertical speed indicator. [4.10]

4.7.1 VSI Operation

Figure 4.17 illustrates the operation of the VSI. The needle is linked to the capsule which expands or contracts depending on the pressure differential which exists across it. The inside of the capsule is linked directly to the static port and is subjected to the static pressure. The area surrounding the capsule inside the sealed unit is also connected to static pressure, but the air must first pass through a restrictive metering unit. As a result, the pressure outside the capsule (inside the VSI) will lag the pressure inside the capsule and result in a pressure difference. The magnitude of the pressure differential is a function of the metering unit's construction as well as the rate and direction of vertical speed [4.3].

4.7.2 VSI Simulation

Referring again to Figure 4.17 again, the VSI can be modelled as a first order fluid system. As such it will take a few seconds for the VSI to give an accurate reading. Settling times of 6 to 9 seconds is not uncommon [4.1].

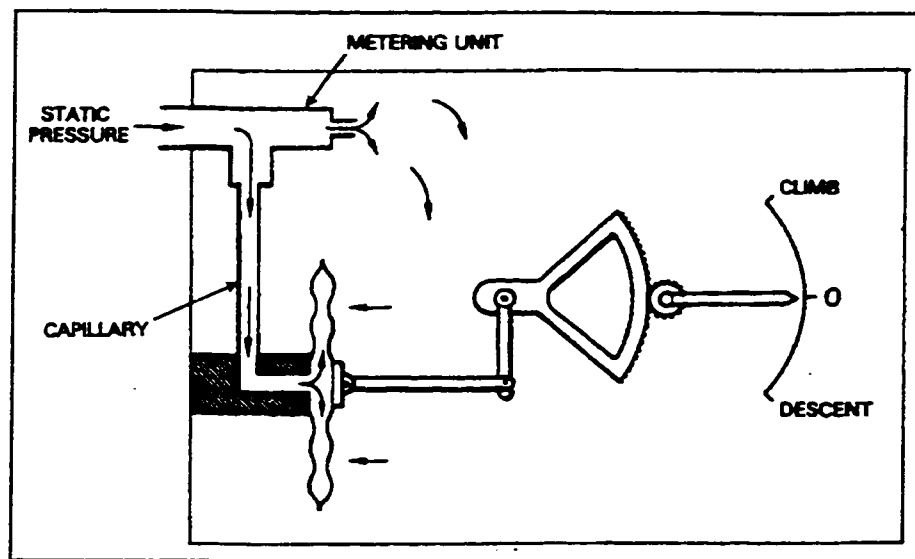


Figure 4.17 Vertical speed indicator operation. [4.3]

Table 4.3 lists the angular displacement of the pointer from the nine o'clock position as a function of the climbing indicated vertical speed. Note that the displacement of the pointer for the descending vertical speed is equal and opposite to that of the equal climbing vertical speed.

VSI	deg	VSI	deg	VSI	deg	VSI	deg
0	0.0	500	32.9	1000	68.9	3500	172.0
100	7.8	600	40.0	1500	88.4		
200	13.4	700	46.6	2000	108.8		
300	19.8	800	55.0	2500	128.6		
400	26.7	900	62.4	3000	148.8		

Table 4.3 Vertical speed indicator angular displacement.

5.0 RADIO NAVIGATION SYSTEMS, OPERATION AND SIMULATION

Early aviation soon realized the importance of accurate navigational techniques. Aviation pioneers initially employed dead reckoning in which their ground speed was calculated from the time required to reach a known checkpoint. This velocity was used then to estimate their position after a measured elapsed time. Dead reckoning techniques are still taught today to student pilots. Dead reckoning is obviously limited to clear days when landmarks are clearly visible from the air. However, these techniques depend on the pilot not getting lost. The need for more reliable methods lead to the development of modern navigation systems based on ground transmitters. The most common systems found in most general aviation aircraft consist of: the Automatic Direction Finders (ADF), distance measuring equipment (DME), very high frequency omni-range receiver (VOR), and the instrument landing system (ILS).

5.1 The Automatic Direction Finder

The simplest radio navigation system is the ADF, it provides the pilot with a relative bearing from the aircraft to a ground transmitter. The bearing is indicated to the pilot over a compass card as shown in Figure 5.1. The top of the compass card always indicates the forward direction of the aircraft thus the arrow always points to the transmitter. The compass card could be slaved to the gyroscopic compass to maintain a correct orientation or in less sophisticated system, rely on the pilot to manually adjust it. In either case the arrow is not affected by the orientation of the compass card. By following this bearing the pilot can direct the aircraft to the selected transmitter. The difficulty in this system is that since it only provides a relative bearing from the aircraft to the transmitter, any cross wind would tend to push the aircraft off the direct route and cause it to fly the path shown in figure 5.2. A pilot could correct for the cross wind by flying a heading in which the

indicated relative bearing remains constant but does not necessarily point toward the front of the aircraft. [5.1]

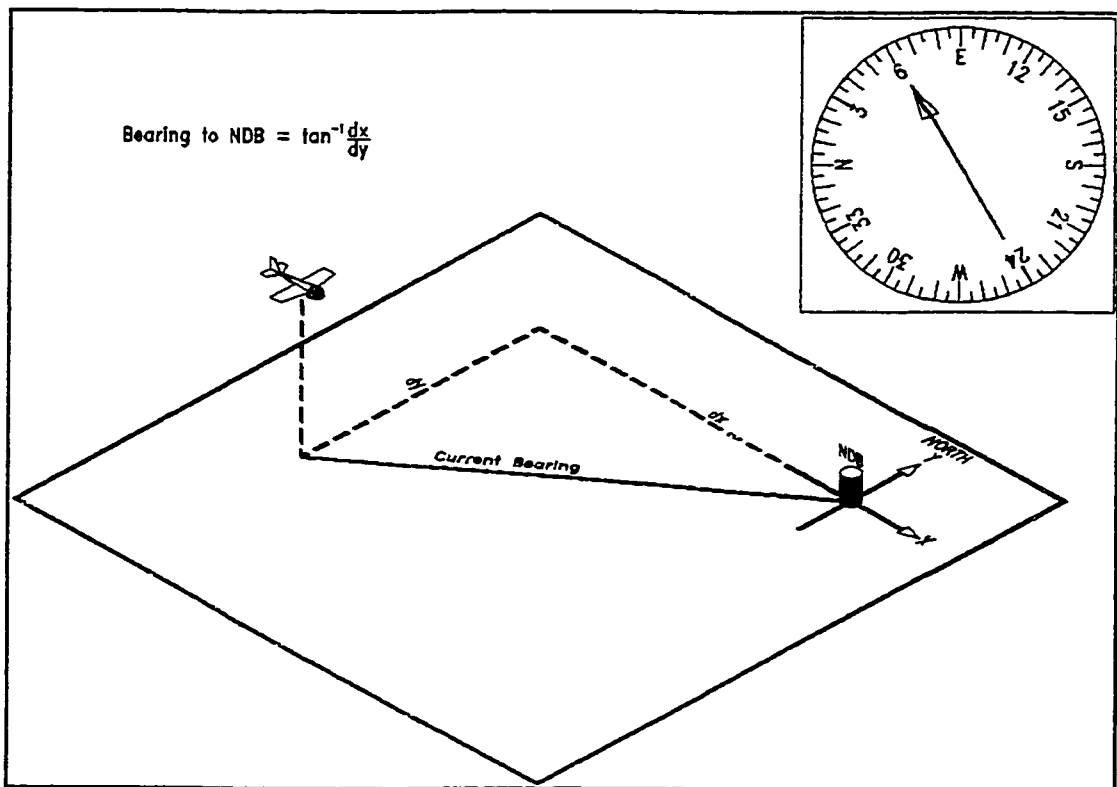


Figure 5.1 Aircraft bearing to a Non Directional Beacon (NDB).

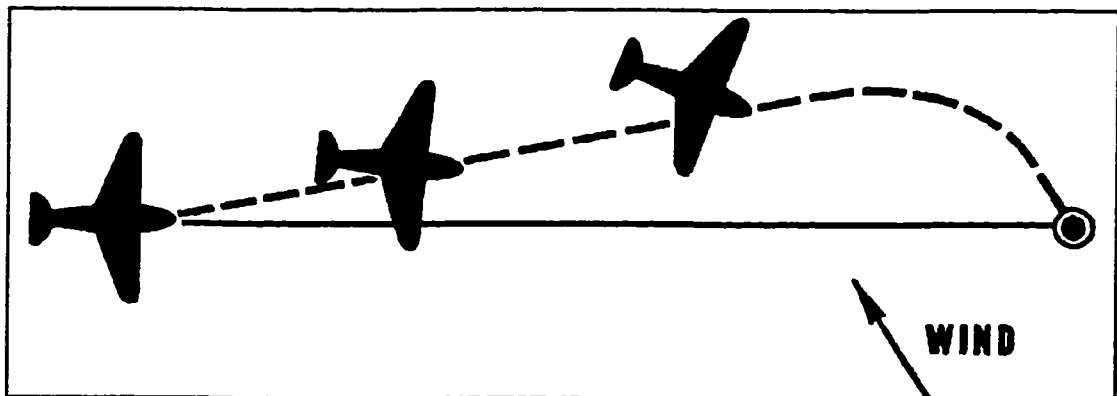


Figure 5.2 Aircraft tracking an NDB without compensating for cross wind. [4.1]

5.1.1 ADF Theory of Operation

A simple single antenna placed in the alternating electromagnetic field of a radio transmitter will have a small alternating voltage induced on it which is proportional to the radio signal and the distance from the transmitter. This is illustrated in figure 5.3.

Consider a pair of antennas, A and B each connected to one end of a transformer's primary winding. If the antenna arrangement is such that A is closer to the transmitter than B as shown in figure 5.4, the amplitude at A will be slightly greater than at B resulting in an oscillating

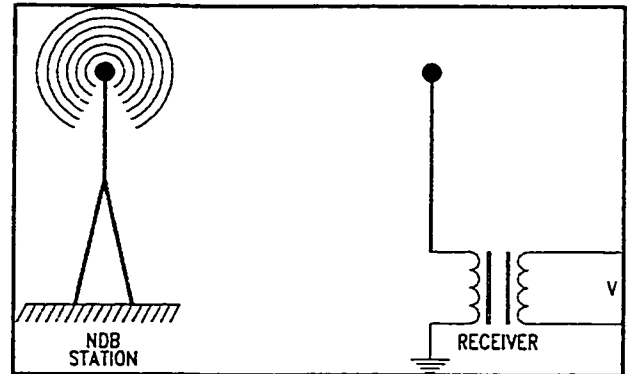


Figure 5.3 NDB station and ADF receiver.

voltage across the transformer of a given phase. As the antennae are rotated such that A and B are equidistant to the transmitter, both A and B will have an equal voltage induced on them, and thus no current will flow through the transformer resulting in a null. As the antennae continue to rotate past this point, B will be closer to the transmitter, which will again result in an oscillating voltage across the transformer but of the opposite phase. As mentioned earlier, the magnitude of the induced voltage is very small. To increase this voltage, a loop antenna is used, effectively increasing the length of A and B. The relative bearing to a transmitter could be determined by rotating the antenna until a null is detected at which point the transmitter would be in a direction perpendicular to the antenna loop. Depending on the direction of the phase shift as the antenna passes the null, the position of the transmitter with respect to the loop (in front of or behind) can be determined. [5.2]

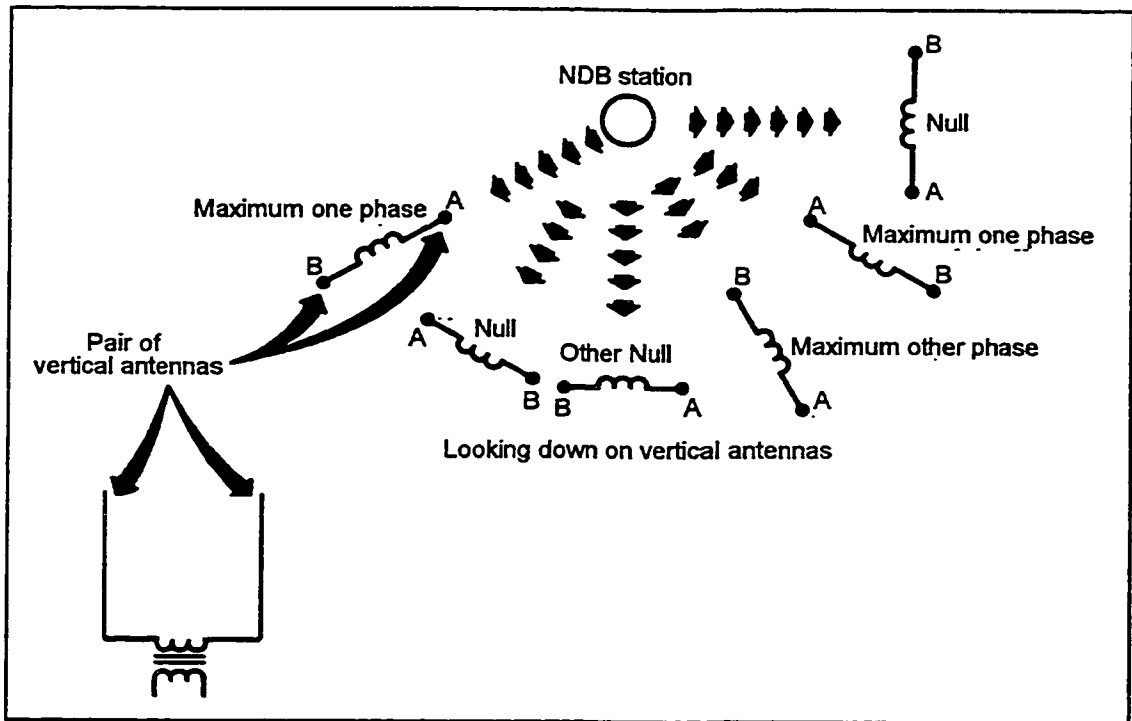


Figure 5.4 ADF operating principle. [5.2]

This technique of rotating the antenna was used on early systems, but it required a constant adjustment by the navigator. By mounting two mutually perpendicular loop antennas L1 and L2, the amplitude of the resulting signals from each antenna would be a function of the relative bearing to the transmitter θ_{ADF} and would vary by 90 degrees according to Equations 5.1 and 5.2.

$$A_1 = A \sin(\theta_{ADF}) \quad (5.1)$$

$$A_2 = A \sin(\theta_{ADF} + 90) = A \cos(\theta_{ADF}) \quad (5.2)$$

Solving these equations simultaneously yields a solution for the relative bearing to the transmitter in terms of the two resulting amplitudes from each antenna.

$$\theta_{ADF} = \text{atan}\left(\frac{A_1}{A_2}\right) \quad (5.3)$$

Dedicated non directional beacons (NDB) operate in the low to medium frequency range of 200-415 KHz and 510-535 KHz, however most ADF receivers are designed to tune signals from 190 up to 1750 KHz [5.3] allowing commercial AM broadcasting stations to be used as navigational aids. Figure 5.5 shows a typical ADF receiver control panel.

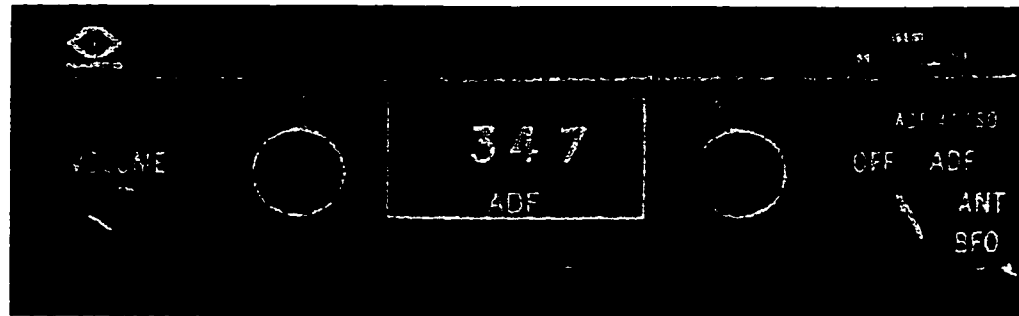


Figure 5.5 Typical ADF control panel. [5.4]

Being low to medium frequency signals, NDB signals travelling over land are less susceptible to ground attenuation than higher frequency signals. The degree of attenuation of a signal travelling over land depends on the frequency of the signal (lower signals suffer less attenuation), and the conductivity of the surface below it. NDB signals travelling skyward can be reflected back to Earth by the ionosphere. This can greatly increase the effective range of the beacon. However, the extent to which the signal is reflected and the increase in range is a function of both the frequency and the condition of the ionosphere. As a result, there is no simple means to determine the range of an NDB beacon, it can vary significantly based on several factors. [4.1]

5.1.2 ADF Simulation

The ADF is simulated by calculating the relative bearing from the aircraft to the transmitter. A more accurate polar approach, taking into account the curvature of the Earth, is used as opposed to the method described in Figure 5.1. For these calculations, the Earth centered fixed coordinate system is used. The location of the transmitters are defined in published documents such as the Canadian Flight Supplement (CFS) in terms of their latitude and longitude. [5.5]

Consider an aircraft located at A_1 and an NDB beacon located at N_1 on the Earth as shown in Figure 5.6. The calculation to determine the relative bearing to the aircraft is achieved through two transformations.

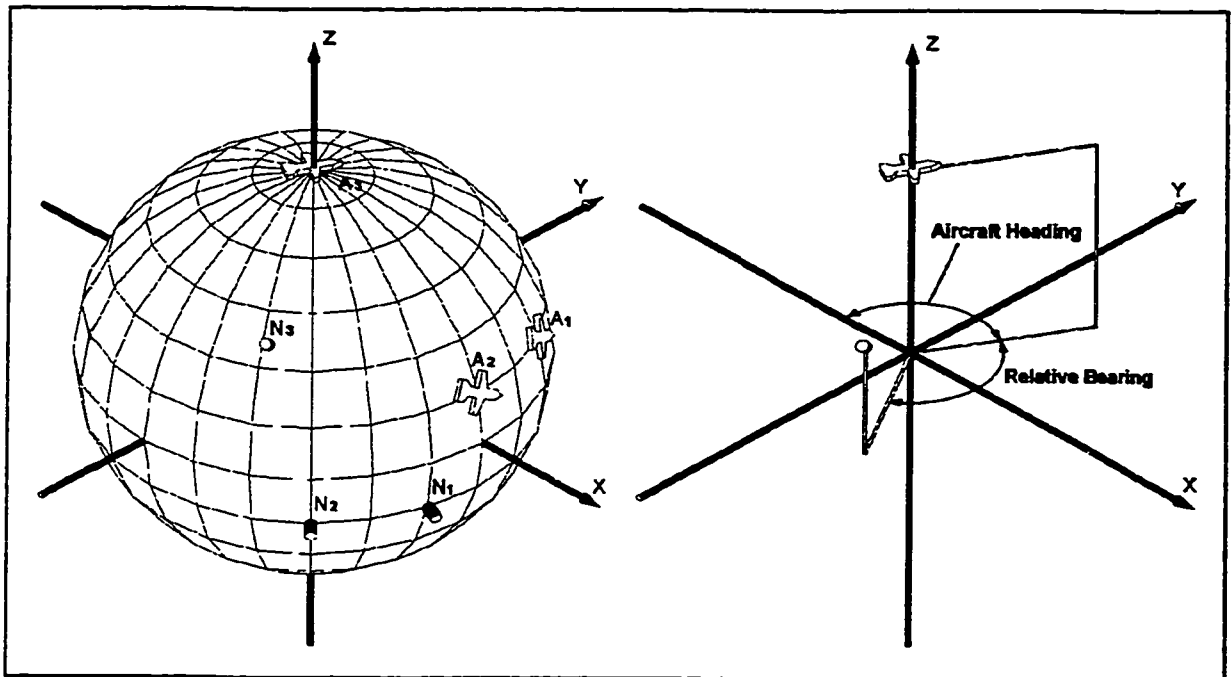


Figure 5.6 Relative bearing from aircraft to NDB beacon.

The aircraft and NDB are located respectively at coordinates $A_1(\text{Lat}_{A_1}, \text{Lon}_{A_1})$ and $N_1(\text{Lat}_{N_1}, \text{Lon}_{N_1})$. Firstly, the aircraft and NDB are equally rotated about the Z axis (Earth's axis) by the negative of the aircraft's longitude so that the aircraft is on the

zero longitude to positions A_2 and N_2 :

$$A_2 (Lat_{A1}, 0) \quad (5.4)$$

$$N_2 (Lat_{N1}, Lon_{N1} - Lon_{A1}) \quad (5.5)$$

$$\begin{aligned} N_2 (X_{N2}, Y_{N2}, Z_{N2}) = & (R_{earth} \cos(Lat_{N2}) \cos(Lon_{N2}), \\ & R_{earth} \cos(Lat_{N2}) \sin(Lon_{N2}), \\ & R_{earth} \sin(Lat_{N2})) \end{aligned} \quad (5.6)$$

Next the aircraft and NDB are equally rotated about the Y axis such that the aircraft is at the north pole at positions N_3 and A_3 , respectively.

$$\begin{bmatrix} X_{N3} \\ Y_{N3} \\ Z_{N3} \end{bmatrix} = \begin{bmatrix} \cos(90 - Lat_{A2}) & 0 & -\sin(90 - Lat_{A2}) \\ 0 & 1 & 0 \\ \sin(90 - Lat_{A2}) & 0 & \cos(90 - Lat_{A2}) \end{bmatrix} \begin{bmatrix} X_{N2} \\ Y_{N2} \\ Z_{N2} \end{bmatrix} \quad (5.7)$$

Finally the bearing to NDB is calculated from Equation 5.8.

$$Bearing_{NDB} = \tan^{-1} \left(\frac{-X_{N3}}{Y_{N3}} \right) \quad (5.8)$$

and the relative bearing to the NDB is given by Equation 5.9.

$$Relative \ Bearing = Bearing_{NDB} - True \ Heading_{AC} \quad (5.9)$$

Appendix 10 lists the C function used to calculate the bearing to the station. Figure 5.7 illustrates the flowchart used to implement the ADF simulation.

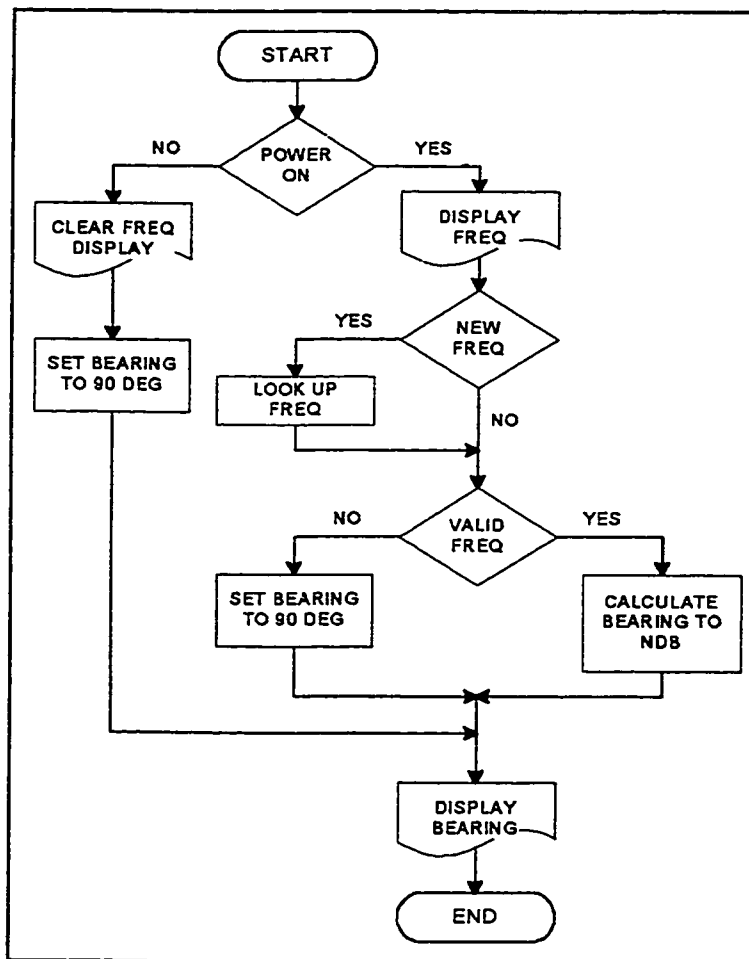


Figure 5.7 ADF simulation flowchart.

5.2 The Very High Frequency Omni-Range Receiver

The Very High Frequency Omni-Range Receiver (VOR) overcomes the limitations of the ADF by providing the pilot with an indication of any deviation from a selected course to the VOR station. This information is indicated either on a Horizontal Situation Indicator (HSI), as shown in Figures 5.8 or a Course Deviation Indicator (CDI), shown in Figure 5.9.

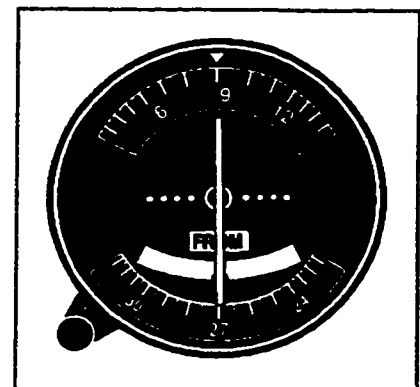


Figure 5.9 Course deviation indicator. [5.6]

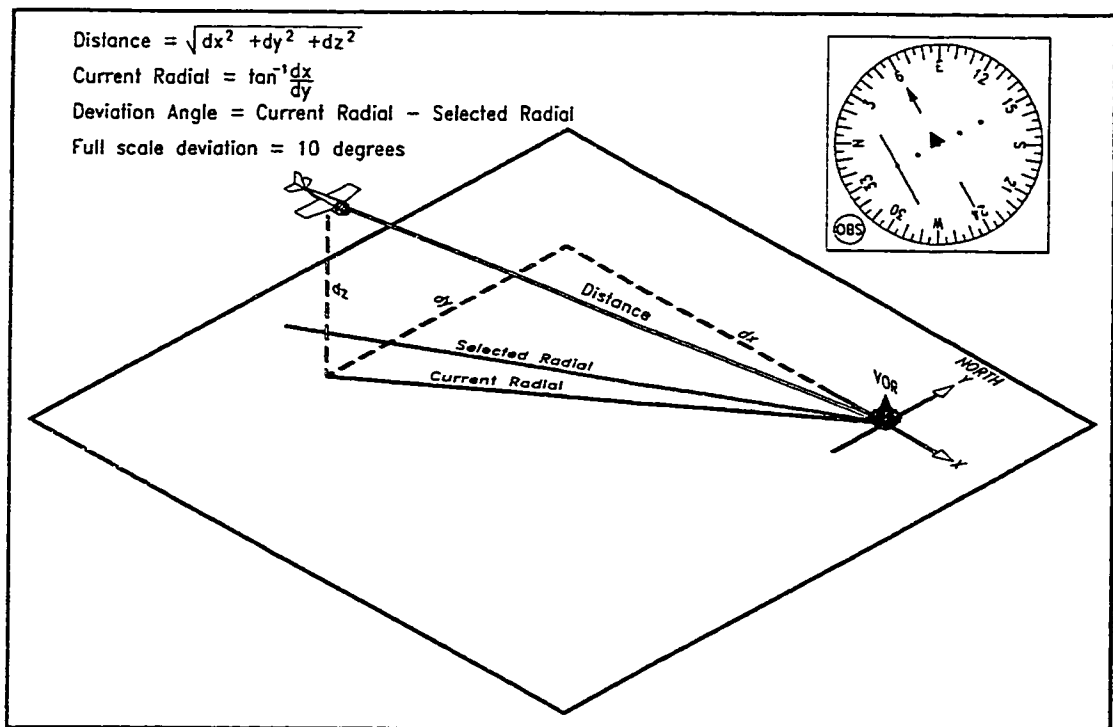


Figure 5.8 Aircraft course deviation, approaching a VOR.

An HSI is slaved to the gyroscopic compass to ensure that the compass card is always oriented correctly to indicate the aircraft's heading. A radial is defined as a straight path away from a station at a defined bearing. The pilot selects the desired radial to fly towards or away by orienting the arrow on the compass card using the omnibearing selector (OBS) knob. The deviation from the selected radial is indicated by the deviation of the centre portion of the arrow as shown in figure 5.8. Different indicators may have different amounts of dots but the full scale deviation is plus or minus 10 degrees from the selected track [5.1][5.2]. As a rule, if the selected track points to the VOR station as the aircraft flies towards it and away from the station as the aircraft flies away, then the pilot should correct the course in the direction of the needle's deviation.

Another indication on the HSI is the 'TO' and 'FROM' indicator. If the absolute value of the deviation from the selected radial is less than 90 degrees, then the indicator will indicate 'TO'. As the absolute value of the deviation exceeds

90 degrees, the indicator will indicate 'FROM'. This is illustrated in Figure 5.10.

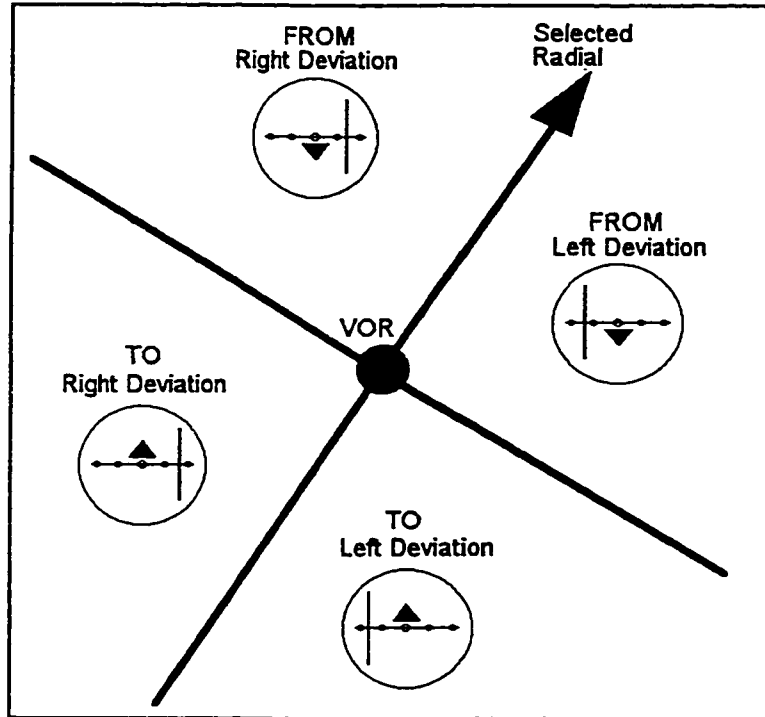


Figure 5.10 VOR indication.

Using the VOR, the pilot can fly a track directly to the VOR station even in the presence of a cross wind.

A VOR operates in the very high frequency range from 108.00 to 117.95 MHz. This is the same frequency range used by the instrument landing system (ILS) localizer beam. Appendix 11 gives a listing of the frequencies in this range, indicating those assigned to VORs and those to ILS systems. In this frequency range, the signal is not reflected by the ionosphere or subject to ground attenuation. The VOR operates along a line of sight and therefore requires an unobstructed path from the beacon to the aircraft. The range of reception for a VOR is estimated by Equation 5.10, where the altitude is in feet and the range is in nautical miles. [5.1]

$$Range = 1.23\sqrt{Altitude} \quad (5.10)$$

VOR stations are often paired with a Distance Measuring Equipment site (DME) to provide an indication on the distance remaining to the station, this is discussed further in the DME section.

5.3.1 VOR Theory of Operation

A VOR station transmits on a carrier wave in the frequency range from 108.00 to 117.95 MHz [5.7]. Placed on this carrier is a frequency modulated 9960 Hz subcarrier modulated from 9480 Hz to 10440 Hz at 30 Hz according Equation 5.11.

$$f = 9960 + 480\sin(30 \times 2\pi t) \quad (5.11)$$

The resulting signal is transmitted on a rotating directional antenna which also turns at 30 revolutions per second such that the signal is strongest in the northern direction when the frequency modulated component is at its maximum frequency. Since the antenna is rotating, a receiver would detect not only the carrier with its frequency modulated signal, but it would receive the signal as being amplitude modulated as the antenna is directed towards and away from it 30 times per second. [5.2] The amplitude modulated component would vary according to Equation 5.12, where A_{\max} is the maximum amplitude and θ is the bearing from the station to the receiver, measured clockwise from the north.

$$A = A_{\max}\sin(30 \times 2\pi t - \theta) \quad (5.12)$$

Figure 5.11 shows a block diagram of a typical VOR receiver. The 9960 subcarrier is removed from the carrier and fed into a 30 Hz amplitude modulation detector and a 30 Hz frequency modulation detector. The phase difference between the amplitude and frequency modulated signals is equal to the bearing of the transmitter

to the aircraft.

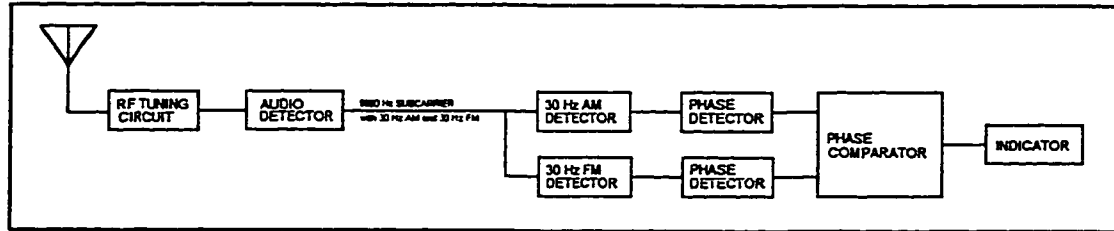


Figure 5.11 VOR receiver block diagram.[5.2]

5.2.2 VOR Simulation

The VOR is simulated in a manner similar to that of the ADF. The only exception being that the bearing from the station to the aircraft is calculated rather than the bearing from the aircraft to the station. Again, this approach is used as opposed to that illustrated in Figure 5.8 to take into account the curvature of the Earth. Since the radials from a VOR station are defined relative to the Earth's magnetic field, the true radial used in the calculations is determined by subtracting the magnetic variation at the VOR station from the selected magnetic bearing. First the coordinates of the aircraft and transmitter are rotated by an equal amount about the Earth's axis to positions A_2 and V_2 such that the station's longitude is at zero as shown in Figure 5.12. The coordinates are then converted to cartesian coordinates using equation 5.15, with the Z axis aligned with the earth's northerly axis and the station on the Y-Z plane. Next, using Equation 5.16, the aircraft and station are rotated by an equal amount about the Y axis to positions A_3 and V_3 such that the VOR station is at the north pole. The bearing from the station to the aircraft is calculated from the aircraft's X and Y coordinates. The selected true radial is subtracted from this bearing to give the deviation.

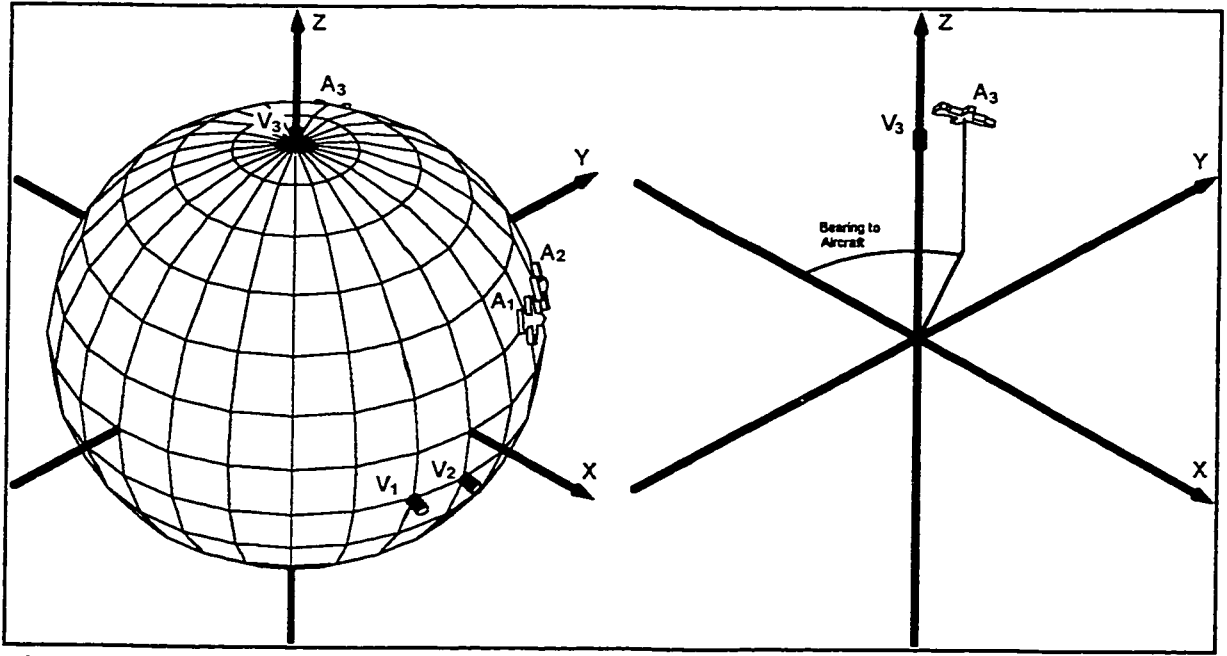


Figure 5.12 Bearing from VOR station to aircraft.

$$V_2 (Lat_{V1}, 0) \quad (5.13)$$

$$A_2 (Lat_{A1}, Lon_{A1} - Lon_{V1}) \quad (5.14)$$

$$A_2 (X_{A2}, Y_{A2}, Z_{A2}) = \begin{pmatrix} R_{earth} \cos(Lat_{A2}) \cos(Lon_{A2}) , \\ R_{earth} \cos(Lat_{A2}) \sin(Lon_{A2}) , \\ R_{earth} \sin(Lat_{A2}) \end{pmatrix} \quad (5.15)$$

$$\begin{bmatrix} X_{A3} \\ Y_{A3} \\ Z_{A3} \end{bmatrix} = \begin{bmatrix} \cos(90 - Lat_{V2}) & 0 & -\sin(90 - Lat_{V2}) \\ 0 & 1 & 0 \\ \sin(90 - Lat_{V2}) & 0 & \cos(90 - Lat_{V2}) \end{bmatrix} \begin{bmatrix} X_{A2} \\ Y_{A2} \\ Z_{A2} \end{bmatrix} \quad (5.16)$$

$$\text{Deviation} = \tan^{-1} \left(-\frac{X_{A3}}{Y_{A3}} \right) - \text{Selected Radial} \quad (5.17)$$

Appendix 10 lists the C function used to calculate the bearing to the station. Figure 5.13 illustrates the flowchart used to implement the VOR simulation.

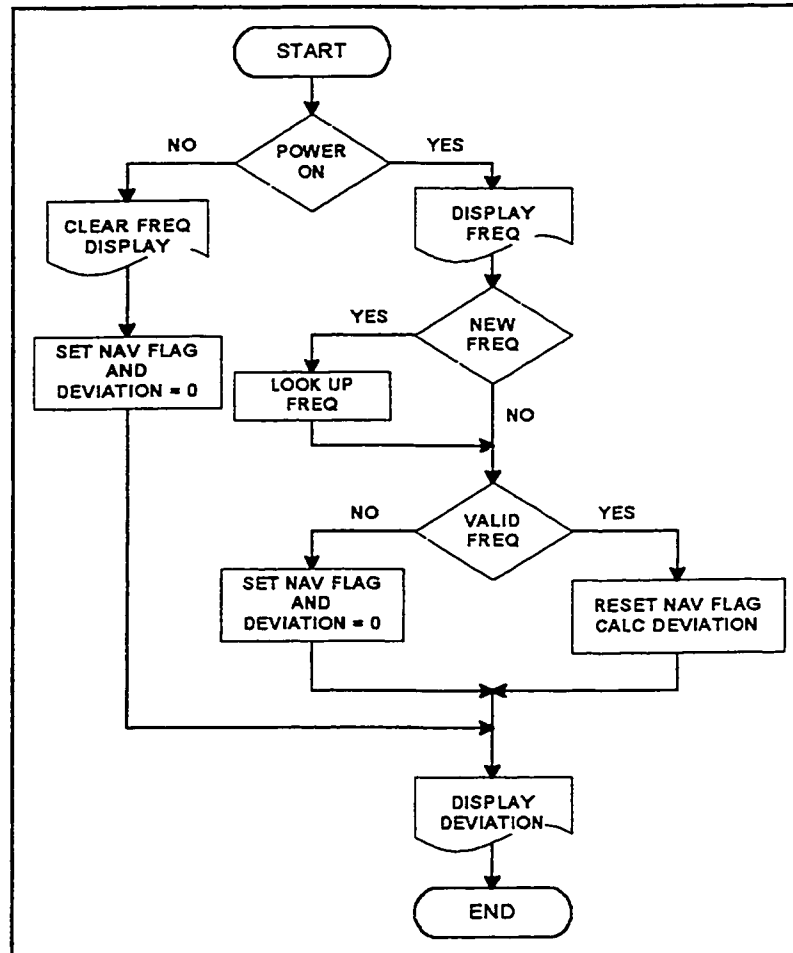


Figure 5.13 VOR simulation flowchart.

5.3 The Distance Measuring Equipment

The DME system provides the pilot with a digital display of the slant range defined as the straight line distance from the aircraft to a ground based station up to 199 nautical miles. DME stations are often collocated with a VOR station, providing the pilot with both range and bearing information. Although the ground distance is the information required by the pilot, over larger distances the difference between the slant and ground range is small. To obtain the ground distance would require additional calculations, as well as interfacing the unit to the altimeter, creating unnecessary complications. It was decided by the avionic manufacturers that, in the interest of simplicity and reliability, the ground distance would not be calculated. [5.8] Many DME systems will differentiate the distance to determine the velocity and estimated time to arrival.

5.3.1 DME Theory of Operation

The DME interrogator transmits a pair of randomly spaced pulses to the ground station on a specific carrier frequency. When the ground station receives the two pulses, it waits 50 microseconds and then retransmits a pair of pulses with the same spacing as the original signal. By measuring the time from the transmitted signal to the reply and subtracting the 50 microsecond delay, the distance to the station may be calculated. Since the time between the pulses is random, an interrogator can differentiate between replies to other aircraft and those replying to its own transmission. [5.2]

The DME's interrogation carrier frequency (from the aircraft) is in the range from 1025 to 1150 MHz, and the ground based DME station replies on a carrier frequency of 63 MHz above or below the interrogating signal.

5.3.2 DME Simulation

The DME system is simulated by calculating the straight line distance from the aircraft to the DME station. Both the coordinates of the aircraft and station are converted from latitude, longitude and altitude to cartesian coordinates using Equations 5.18 and 5.19, and used to calculate the distance between the two using Equation 5.20.

$$\begin{aligned}(X_{AC}, Y_{AC}, Z_{AC}) = & (R_{earth} \cos(Lat_{AC}) \cos(Lon_{AC}), \\ & R_{earth} \cos(Lat_{AC}) \sin(Lon_{AC}), \\ & R_{earth} \sin(Lat_{AC}))\end{aligned}\tag{5.18}$$

$$\begin{aligned}(X_{ST}, Y_{ST}, Z_{ST}) = & (R_{earth} \cos(Lat_{ST}) \cos(Lon_{ST}), \\ & R_{earth} \cos(Lat_{ST}) \sin(Lon_{ST}), \\ & R_{earth} \sin(Lat_{ST}))\end{aligned}\tag{5.19}$$

$$DIST = \sqrt{(X_{AC} - X_{ST})^2 + (Y_{AC} - Y_{ST})^2 + (Z_{AC} - Z_{ST})^2}\tag{5.20}$$

Appendix 10 lists the C function used to calculate the distance to the station.

5.4 The Instrument Landing System

The final phase of any successful flight is the final approach and landing at an airport. For safety as well as the economic importance of maintaining a schedule, an Instrument Landing System (ILS) was developed to guide a pilot during the final approach without a visual reference to the runway or ground. At a predefined altitude before touchdown, called the “decision height”, the landing must

be aborted if the runway is not in sight.

When approaching a runway equipped with an ILS system, the pilot tunes the frequency of the ILS for the intended runway. As with the VOR frequencies, the ILS frequencies range from 108.01 MHz to 119.90 MHz [5.9]. Appendix 11 lists the frequencies which are assigned to VORs and ILSs. When tuned, the pilot will receive information about the aircraft's horizontal alignment with the runway from the localizer, and the vertical deviation from an established

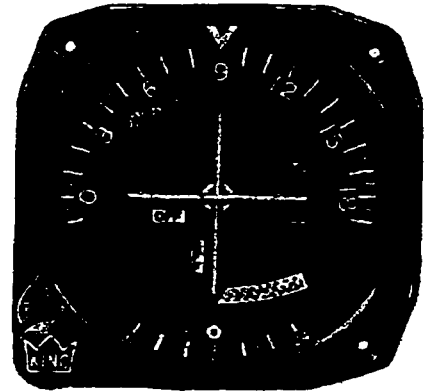


Figure 5.14 CDI with glide slope indication.

approach angle from the glide slope. The information is presented to the pilot on a Course Deviation Indicator (CDI) much like the one used for the VOR system, except that it also provides the glide slope deviation with the deflection of a horizontal bar, and does not require that course be set on the CDI. Figure 5.14 shows a typical CDI with glide slope indication. The same CDI is often used for both the VOR and ILS systems, with a red glide slope flag in view when in the VOR mode to indicate that there is no glide slope indication. The indications of the CDI are shown in Figures 5.15, and 5.16. If the aircraft drifts below the glide slope then the horizontal bar will shift up. If the aircraft drifts above the glide slope then the horizontal bar will shift down. A full scale deflection of the horizontal bar represents a deviation of plus or minus 0.7 degrees from the glide slope. A full scale deflection of the vertical bar represents a deviation of plus or minus 2 degrees from the localizer. [5.2]

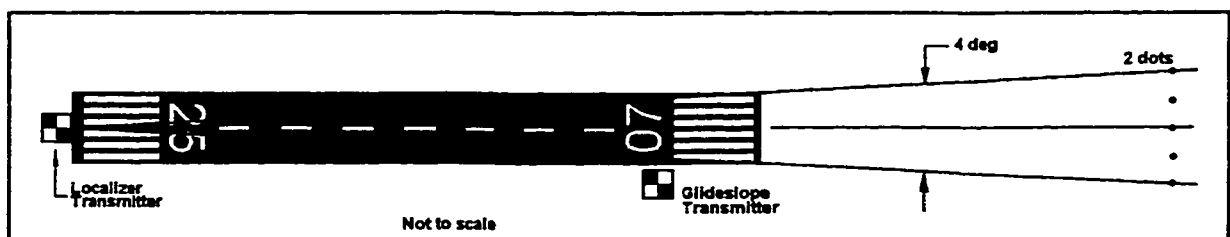


Figure 5.15 Localizer deviation indication.

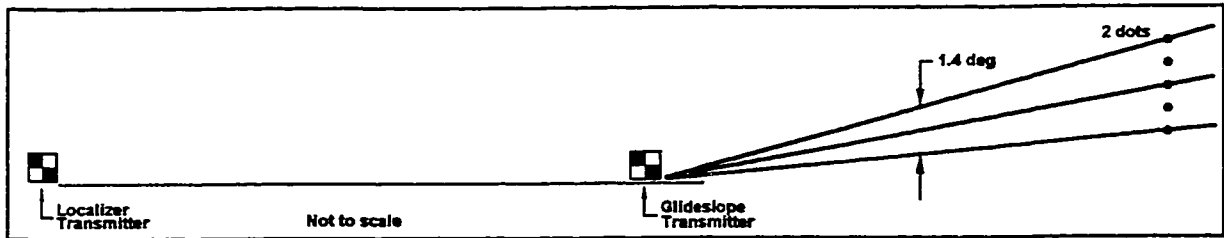


Figure 5.16 Glide slope deviation indication.

5.4.1 ILS Theory of Operation.

Both the glide slope and the localizer operate on a similar principle. Consider first the localizer. Two signals of 90 and 150 Hz are transmitted on a carrier frequency which the pilot tunes. The two signals are transmitted directionally so that their signal strength varies as shown in Figure 5.17. The ILS receiver detects both frequencies and then compares their relative strength and displays the difference between them as a deviation of the vertical bar on the CDI. If the aircraft is exactly on the localiser then both signals will have equal strength and the vertical bar will be centered. If the aircraft is to the left of the localiser beam then the 90 Hz signal will be stronger than the 150 Hz and the difference in strength will be indicated as a displacement to the right on the CDI. Similarly if the aircraft is to the right of the localiser beam then the 150 Hz signal would be stronger and the CDI would show an displacement of the vertical bar to the left. Figure 5.19 shows a block diagram representation of an ILS receiver. [5.2]

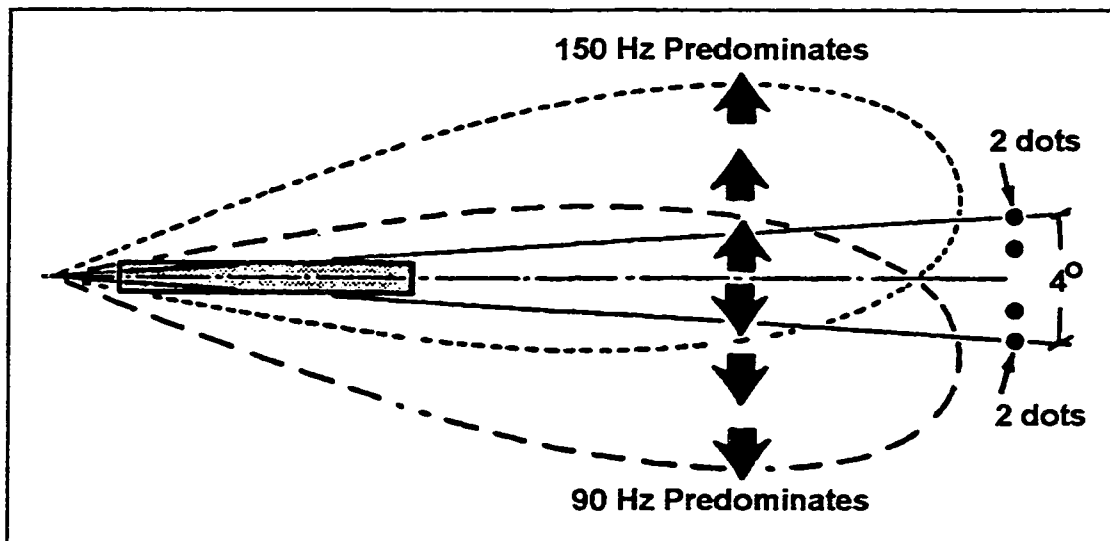


Figure 5.17 Localizer signal pattern. [5.2]

The glide slope operates in exactly the same manner as the localiser except that the 90 and 150 Hz signals are directionally transmitted as shown in Figure 5.18, on a carrier frequency several times higher than the carrier of the localizer. Appendix 11 lists the glide slope frequencies associated with each localiser frequency. Note that the pilot does not need to tune both the localiser and glide slope frequency. When a localiser frequency is selected, the ILS receiver will automatically select the corresponding glide slope frequency. If the aircraft is approaching the runway below the glide slope, then the 150 Hz signal would be stronger and the CDI would show an upward deviation of the horizontal bar. Similarly, if the aircraft drifts above the glide slope then the 90 Hz signal would be stronger and the CDI would show a downward deviation of the horizontal bar.

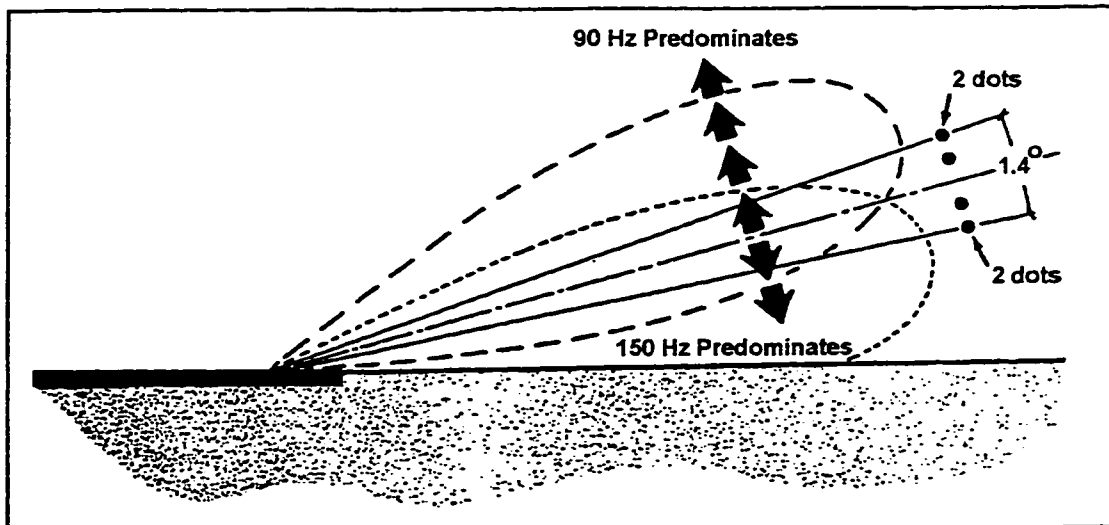


Figure 5.18 Glideslope signal pattern. [5.2]

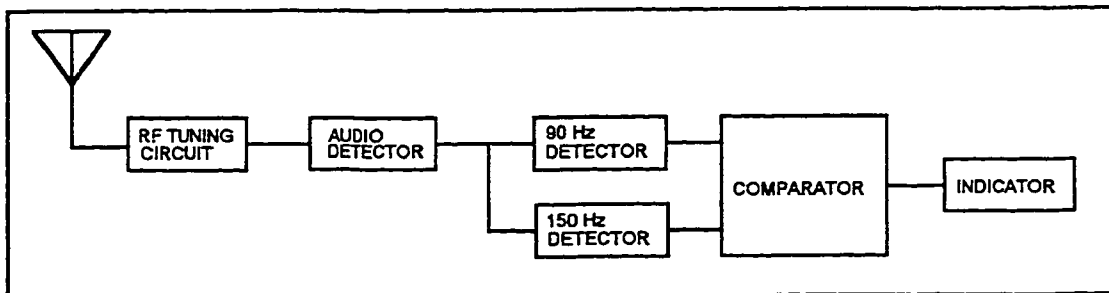


Figure 5.19 Typical ILS glide slope or localiser receiver. [5.2]

5.4.2 ILS Simulation

The localizer portion of the ILS system is simulated in the same manner as the course deviation from a VOR beacon except, that in the case of the ILS system, the pilot does need not select the course (or radial in the case of a VOR) but rather the runway heading is used. Equations 5.13 to 5.17 are used to calculate the deviation of an aircraft from the runway heading. As shown in Figure 5.15, this deviation is displayed on the CDI by the position of the vertical bar, where a full scale deflection would represent a deviation of plus or minus two degrees from the proper approach.

The deviation from the glide slope is calculated in a similar manner to the DME calculations. Since the ILS system is used over the relatively small area around an airport, it is assumed that the curvature of the earth is negligible which greatly simplifies the calculations. First, the straight line distance from the aircraft to the glide slope transmitter is calculated in exactly the same manner used to calculate the distance to a DME station. The approach angle is calculated by taking the arcsine of the quotient of altitude above the runway divided by the straight line distance, as given in equation 5.21.

$$\text{Approach Angle} = \arcsin\left(\frac{\text{Altitude Above Runway}}{\text{Straight Line Distance}}\right) \quad (5.21)$$

This approach angle is compared to the glide slope angle, typically 2.5 to 3 degrees, and the deviation is displayed on the CDI.

6.0 HARDWARE CONFIGURATION

The new software was developed to operate in a number of configurations ranging from a desktop PC with minimal input devices, to a PC with more advanced input devices such as mice, joysticks and rudder pedals, to a fully instrumented representation of a light twin engine aircraft: the Beech Dutchess 76.

6.1 Full Simulated Cockpit Environment

The Concordia Light Aircraft Simulation System consists of four major components: the computer and its interface, the cockpit section containing all the flight instruments and controls, the visual system which generates the forward view projected ahead of the windshield, and the instructor station to monitor and control the simulation. Figure 4.1 shows the actual cockpit section of the flight simulator.

One of the objectives of this project was to preserve the basic configuration while upgrading the computer and interface. The new configuration supports the development of advanced flight models using high level programming languages, and has the capability to read and drive many devices with its large complement of analog and digital interfaces.

6.1.1 The Original Simulator System

The original simulator, which was started in the early 1980s, was representative of the state of the art technology of the time. The flight model and all interfaces were driven by an Intel 8088 CPU at 4 MHz. The flight model was loaded each time from an 8088 personal computer via one of four RS-232 serial ports at 9600 baud. The personal computer acting as a dumb terminal performed the task

of the instructor station via an RS-232 link with very limited capabilities. Custom designed and built interface cards were mounted on an 8 bit data bus to drive all the cockpit instruments and controls.

A separate Silicon Graphics IRIS work station received the coordinates and orientation of the aircraft from the CPU via an RS-232 interface, and generated the forward cockpit view which is projected ahead of the cockpit.

6.1.2 Objectives of Upgraded System

The goal of this project was to upgrade the system to a modern platform using modern computers, and to interface it to the cockpit section. This will have numerous advantages.

The first and most obvious advantage of an upgraded system will be an increase in performance. On the original system, the flight model recalculated all parameters at an iteration rate of 9 Hz. Running a Pentium 150 MHz motherboard has yielded an increase in performance to an iteration rate in excess of 1000 Hz when running without the burden of interfacing to the cockpit section, and over 100 Hz when interfaced to the cockpit section. This performance rivals commercially available flight simulators which typically run at a maximum of 60 Hz, while simulating many more systems in greater detail. This translates directly into an increase in fidelity especially during rapid changes in flight conditions, and will permit the implementation of more complex flight models such as stalls and other unusual attitudes, with very minor losses in performance.

Using an 80X86 family computer will ensure the ability to upgrade the motherboard as this line of computers is advanced, without necessarily making major modifications to the flight model or interface cards. With its many analog and

digital interfaces, this system was built in a manner so that it may be connected to drive other simulators or act as a versatile laboratory data acquisition system.

6.1.3 Communication Systems

The original computer was limited to communicating with the instructor station and visual system via an RS-232 interface. The new system includes an ethernet network card, allowing for high speed communication with other hardware. Although not exploited in this work, the network card opens many possibilities. Communication with the visual system could be greatly accelerated, allowing it to concentrate on the computationally intensive task of generating the visual image, rather than communication. With an interface program rather than the complete flight model, the flight models on other computers could be interfaced and tested with the cockpit section. The possibility of networking multiple simulators to create a virtual airspace becomes possible.

6.1.4 Cockpit Signal Requirements

The cockpit section provides the pilot with a simulated cockpit environment, recreating the accurate response of all the instruments and controls. The new interface is able to read and send all the necessary analog and digital signals required to drive the cockpit section's flight instruments and controls in a realistic manner. This thesis focuses on the requirements to drive the flight instruments.

6.1.4.1 Analog Output Requirements

The majority of the flight instruments are of an analog type, which represent the required information by the position of a needle. Where possible, a common drive system was used in most instruments.

6.1.4.1.1 Brushless DC Motors

Since many instruments require that the needle be capable of making a full rotation, brushless dc motors were selected. The selected motor is a model 11-PTR-8RQX manufactured by Clifton Precision. The operation of this motor is shown in Figure 6.1. Two independent coils are energized by voltages V_1 and V_2 , up to a maximum of ± 3 volts. A permanent magnet is mounted on the output shaft which will align itself with the resultant magnetic field. The angle of the output shaft ϕ_s is given by equations 6.1 and 6.2. [6.1]

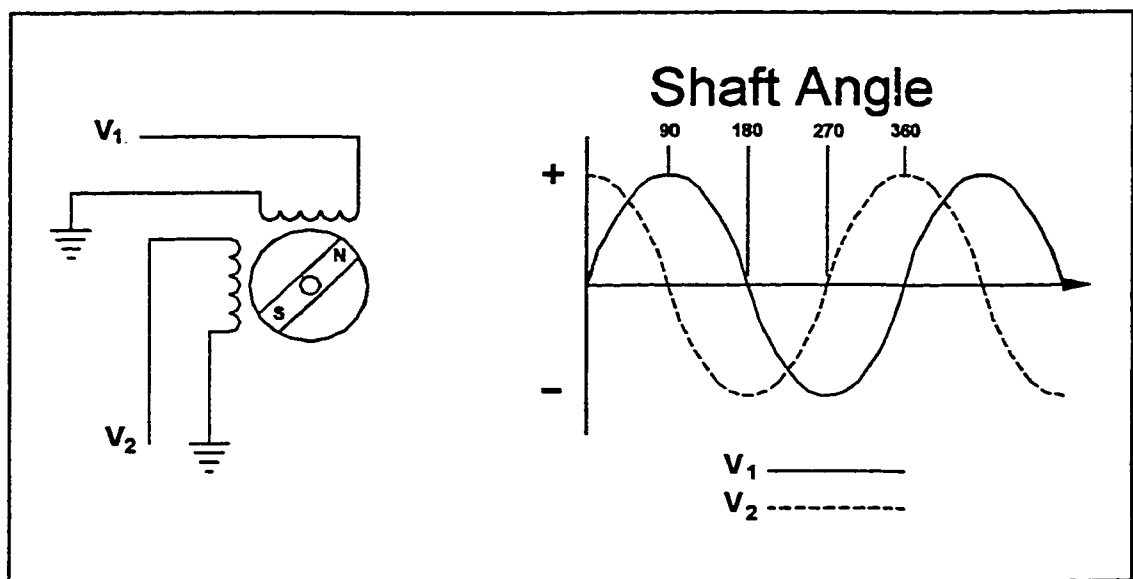


Figure 6.1 Brushless DC motor operation.

$$V_1 = 3\sin\phi_s \quad (6.1)$$

$$V_2 = 3\cos\phi_s \quad (6.2)$$

A major advantage of using these motors is that they do not require any feedback or initialization for position control. The drawback is that they require two signals to drive them. Table 6.1 lists all instruments driven by brushless dc motors and the number of motors required to drive each of them.

Instruments Driven By Brushless DC Motors

Air Speed Indicator	1
Altimeter	3
Manifold Pressure	2
Turn Coordinator	2
Vertical Speed Indicator	1
Tachometer	2
ADF	2
Instrument Air	1
Magnetic Compass	1
Thermometer	1
TOTAL	16

Table 6.1

The accuracy achieved by this system is further described in Section 6.1.5 with the description of the analog output board.

6.1.4.1.2 Galvometer Gauges

Other instruments function as simple galvometers, requiring only a variable voltage in the range from 0 to 10 volts dc to deflect their needles. Table 6.2 lists all the galvometer type instruments.

Galvometer Instruments

Left Fuel Gauge
Right Fuel Gauge
Left Fuel Pressure
Right Fuel Pressure
Left Oil Pressure
Right Oil Pressure
Left Oil Temperature
Right Oil Temperature
Left Cylinder Head Temperature
Right Cylinder Head Temperature
Left Alternator Load
Right Alternator Load
Left Exhaust Gas Temperature
Right Exhaust Gas Temperature
HSI Deviation Bar
HSI Glideslope Indicator
CDI Deviation Bar
CDI Glideslope Indicator
Flap Indicator

Table 6.2

6.1.4.1.3 Stepper Motor Driven Attitude Indicator

The compact brushless dc motor used to directly drive the needles of other instruments do not develop sufficient torque to drive the mechanically complex attitude indicator. For this reason two Airpax K82401-P1 stepper motors are used to drive this instrument. The operation of a stepper motor is shown in Figure 6.2. By sequentially energizing the motors coils, the shaft is incrementally rotated by a fixed angle. Through a gear reduction, the step angle is reduced. However, this is still too coarse to move the attitude indicator with a smooth and realistic motion. For this reason a microstepping technique is used whereby a smoothly varying sine and cosine signal are sent to the motors rather than a stepped signal, thus giving a smooth and variable motion between the steps. [6.2] Appendix 12 lists selected specifications from the manufacturer.

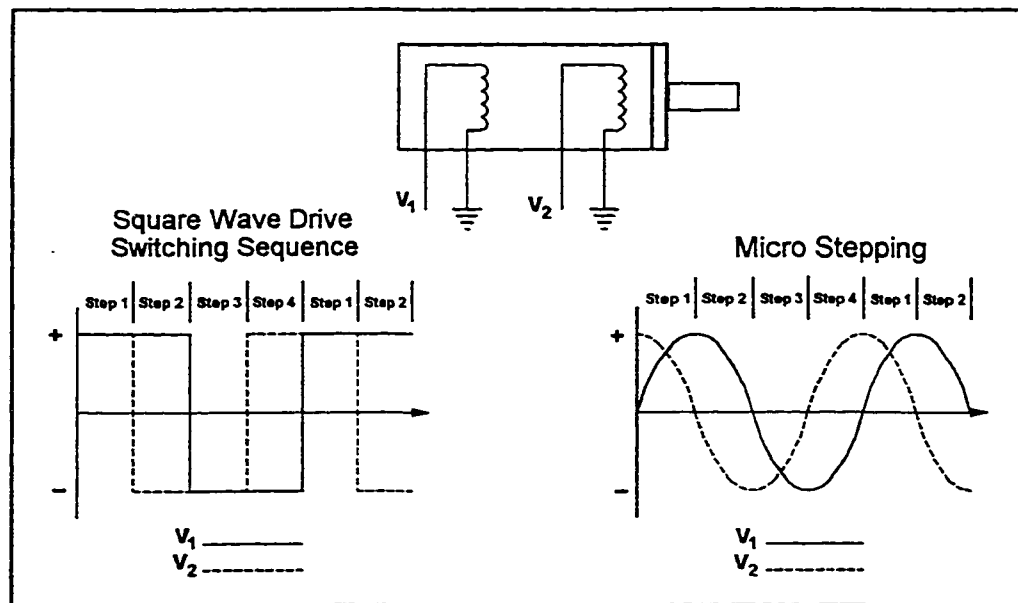


Figure 6.2 Stepper motor micro stepping

6.1.5 Analog Output Card

It was found that commercially available analog output cards lacked sufficient channels and power to drive the cockpit instruments. For this reason a custom designed analog output card was designed and built. Schematically shown in Figure 6.3, two eight bit 7228 digital to analog converter (DAC) chip analog outputs are simultaneously driven by a single sixteen bit data word. This increases the performance of the system by supplying both voltages required to position a brushless DC motor with a single sixteen bit command. The reduction in precision resulting in the use of eight bit DACs as opposed to twelve bit DACs is within an acceptable range. Shown in Figure 6.4, the step between two consecutive angular positions never exceeds 0.45 degrees. The output of the 32 DACs is amplified by a 759 operational amplifier capable of delivering up to 0.5 amps. The 7228 DACs are supplied with a variable volt reference which is set to 3 volts. The outputs directly drive the amplifier stage, providing an output which varies from minus to plus three volts as the DAC's output varies from zero to three volts. The amplifiers'

outputs are sufficient to drive both the brushless dc motors used in the flight instruments, as well the stepper motors driving the attitude indicator.

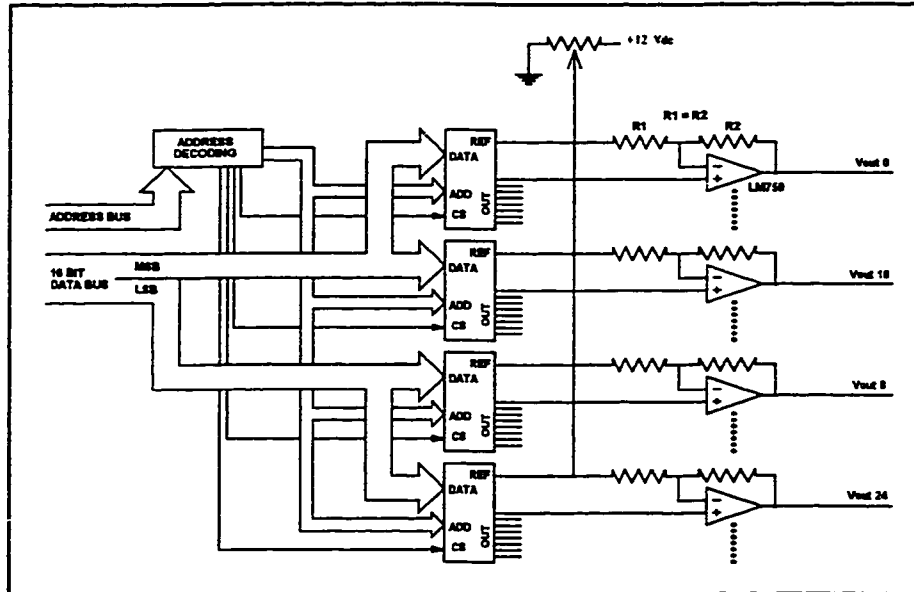


Figure 6.3 Analog output board schematic.

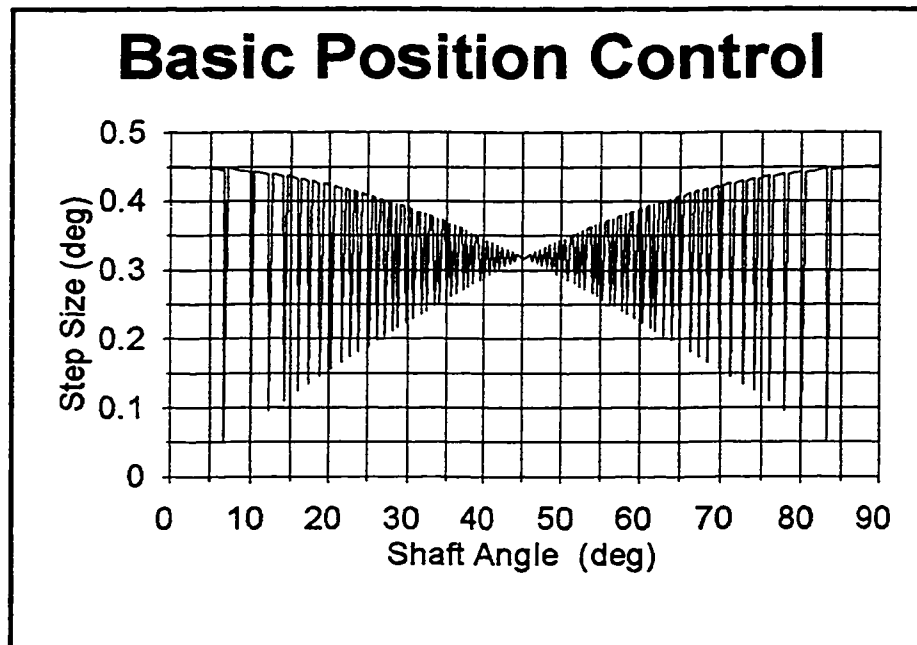


Figure 6.4 Brushless DC motor step size with 8 bit DAC resolution

Although not yet implemented, another concept is under consideration to achieve a smaller step size while keeping the same eight bit DACs. Consider Figure 6.5. The axis represents the respective voltages V_1 and V_2 driving the two coils in the motor. The circle represents the voltages used according to equations 6.1 and 6.2. The square indicates the maximum voltage of three volts for each coil. If only the points on the circle are considered, all the points between

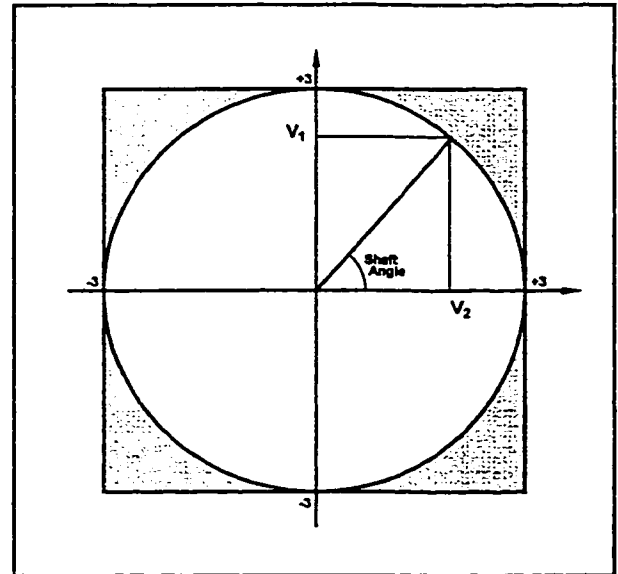


Figure 6.5 Brushless DC motor optimization.

the circle and the square are ignored. If one also considered these points, then the number of possible voltage combinations available to drive the motor would greatly increase, offering more shaft positions and significantly decreasing the step size depending on the position of the shaft as shown in Figure 6.6.

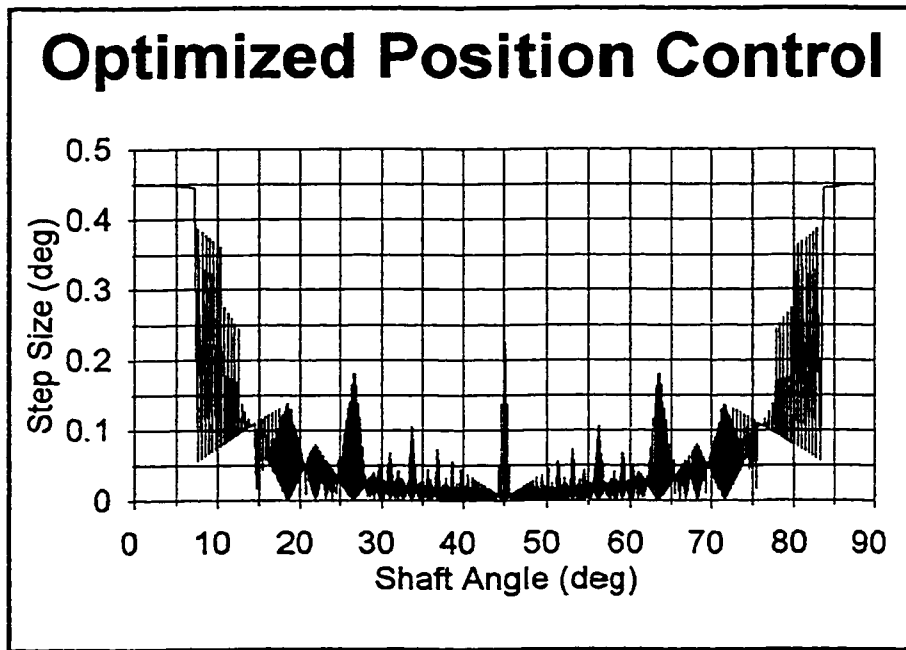


Figure 6.6 Optimized brushless DC motor step size with 8 bit DACs.

Future work will see the development of an analog output card with an onboard programmable logic device. This will read a single 16 bit word representing the desired angular displacement, and in turn drive two 12 bit DACs. This will permit a higher resolution output without sacrificing performance by requiring any additional commands from the computer.

6.1.6 Analog Inputs Requirements

All proportional flight control signals are created by a potentiometer, with the exception of the heading bug of the HSI and the heading selector on the CDI which are created by synchros. All potentiometers are driven with a voltage between 0 and 5 volts. However, due to mechanical limitations, not all are employed over their full range of travel. The result is that, although they are driven by 0 to 5 volts, some outputs may vary over a much smaller range. Table 6.3 lists all the

analog controls.

Potentiometer Analog Outputs

Altimeter Setting
Left Throttle
Right Throttle
Left Propeller Pitch Control
Right Propeller Pitch Control
Left Mixture Control
Right Mixture Control
Left Carburettor Heat
Right Carburettor Heat
Ailerons
Elevator
Rudder
Left Cowl Flaps
Right Cowl Flaps
Left Fuel Selector
Right Fuel Selector
Rudder Trim
Elevator Trim
Left Brake
Right Brake
Instrument Flood
Post Lights

Table 6.3

The analog signals are read by a National Instruments AT-MIO-64E-5 card which provides 64 12 bit analog input channels at a sampling rate of 100 kHz. This is a 16 bit full size ISA bus card mounted in the PC. Being a 12 bit analog to digital converter, it has a sufficient resolution to read those inputs which do not use the full voltage range without any noticeable loss in performance. Appendix 13 lists selected specifications for the AT-MIO-64E-5 card. [6.3]

6.1.7 Digital Outputs

Indicator lights, warning lights, and flags require a single digital signal to turn them on or off. The signal must be capable of delivering sufficient current to energize the system. Table 6.4 lists all the digital indicators.

Digital Indicators	
Clock	DC
Left Hobbs Meter	DC
Right Hobbs Meter	DC
HSI Heading Flag	DC
HSI Glideslope Flag	DC
CDI Heading Flag	DC
CDI Glideslope Flag	DC
Left Alternator Under Voltage Light	DC
Left Alternator Over Voltage Light	DC
Left Alternator Under Voltage Light	DC
Left Alternator Over Voltage Light	DC
Landing Gear Transit Light Test	DC
Nose Gear Down Light Test	DC
Left Gear Down Light Test	DC
Right Gear Down Light Test	DC
Landing Gear Transit Light	DC
Nose Gear Down Light	DC
Left Gear Down Light	DC
Right Gear Down Light	DC
Left Instrument Air Fail	DC
Right Instrument Air Fail	DC
Ventilation Fan	AC
Force Feel Master Air Supply Solenoid Valve	AC

Table 6.4

Both the digital inputs and outputs are handled by a 216 line Industrial Computer Source PCDIO 216 digital I/O board. This is a full size 8 bit ISA bus card mounted in the PC. The digital outputs driving DC devices are each optically isolated by an ODC5 module and the AC devices by OAC5 modules. The modules are mounted on two 16 position back planes. Appendix 14 lists selected specifications for the PCDIO 216 card. [6.4]

6.1.8 Digital Inputs

The cockpit electrical system has a number of switches and circuit breakers, each of which must be read on its own digital line. Table 6.5 lists these devices.

<u>On / Off Switches</u>
14 Rocker Switches
36 Circuit Breakers
Up Electric Trim
Down Electric Trim
Landing Gear
Flaps Up
Flaps Down
<u>Alternator Warning Lamp Test</u>
Table 6.5

These inputs may be read directly by the PCDIO 216 card since each line is equipped with a pull up resistor. Therefore, it is sufficient to connect each switch to an input line, such that closing it will ground the input.

6.1.9 Radio Stack Requirements

The radio stack simulation is not within the scope of this work. However, its requirements are presented, here as a reference for future work. Due to the complexity of the requirements to read all the switches and drive the displays on the radio stack, it has its own micro computer dedicated to this task. This computer communicates with the computer running the flight model via a serial RS-232 link at 9600 baud. This thesis is concerned with the flight model and not the simulation or modelling of the radio stack. It is discussed here only as an existing piece of equipment which the new computer system is equipped to communicate with.

6.2 Desktop Configuration

When running on a desktop PC, the program will detect and make use of the available devices. These include the joystick and rudder pedals connected to the PC's game port, and the mouse.

6.2.1 Interfacing with the PC Game Port

Since many home PCs serve as entertainment devices through interactive games, home computers are often sold with a four channel proportional input channel and 4 discrete input lines. Often only two of the proportional channels are used to read the displacement of a joystick, and two discrete lines are reserved for reading the status of two buttons on the joystick.

A standard joystick is used to represent the aircraft's control stick. A left or right

displacement of the joystick is read through one channel and acts as the pilot's roll command to the aircraft. A forward or backward motion of the joystick is read through the second channel and acts as the pilot's pitch command to the aircraft. The third channel is often used by some joysticks specifically designed to emulate aircraft control such as a throttle. In this thesis, the mouse is used for throttle control and will be discussed later. The fourth channel is used to read the rudder pedal displacement and acts as the pilot's yaw command to the aircraft. Figure 6.7 shows the pin configuration and interfacing for the standard fifteen pin game port.

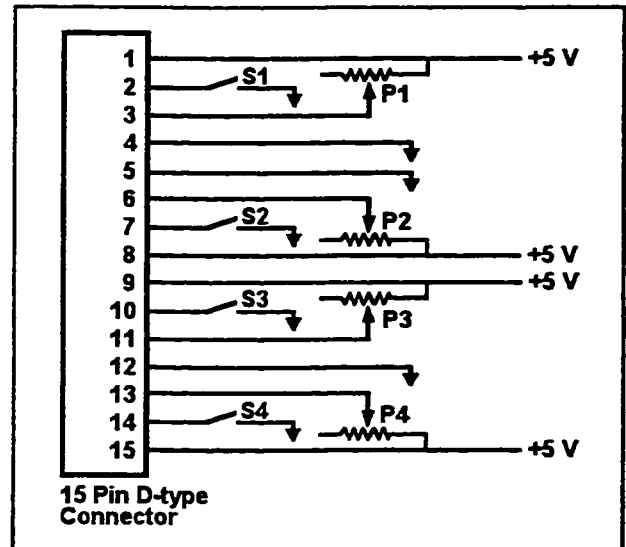


Figure 6.7 PC game port pinout.

It was noticed that the use of the game port requires a significant amount of computing time. This is due to the manner in which the joystick position is read by the computer. Rather than using a digital to analog converter, the variable resistance in the joystick serves as the resistive element in an RC circuit. The time constant is measured and the position of the joystick, or pedal is determined [6.5].

The joystick is read using the DOS BIOS functions. The following is an example of a Borland Turbo C code to read the joystick position.

```
/*-----*/
/*  READ JOYSTICK POSITION      */
/*-----*/
# include <dos.h>
int      X,Y,R;
union RAGS regs;
void READ_JOYSTICK()
{
    regs.h.ah = 0x84;
    regs.x.dx = 0x01;
    int86(0x15,&regs,&regs);
    X = regs.x.ax;          /* X - position      */
    Y = regs.x.bx;          /* Y - position      */
    R = regs.x.cx;          /* Rudder - position */
    return;
}
```

6.2.2 Interfacing With the PC Mouse

A mouse is virtually standard with any new PC. It can operate on a number of mechanical systems such as a rolling ball system or a non contact optical system. In most cases, interface to the PC is via an RS-232 serial port or less commonly through a dedicated interface on the PC's bus. In either case the mouse's position can be read using DOS BIOS functions. The following is an example of a code used to read the mouse position and button status written in Borland Turbo C.

```

/*-----*/
/*  READ MOUSE POSITION AND STATUS */
/*-----*/
# include <dos.h>
int      X,Y,B;
union REGS regs;
void READ_MOUSE()
{
    regs.x.ax = 0x03;
    int86(0x33,&regs,&regs);
    X = regs.x.cx;          /* X - position      */
    Y = regs.x.dx;          /* Y - position      */
    B = regs.x.bx;          /* Button status     */
    return;
}

```

If a joystick is present, then the mouse will be employed as throttles. Pressing the left or right mouse button and advancing or retracting the mouse will advance or retract the left or right throttle.

If the joystick is not present, then the mouse will be used as a control column, moving the mouse in the same direction as one would move the stick. This is a more difficult way to control the simulation because, unlike the joystick, it does not have a reference neutral position. For this reason, if the simulated EFIS display is selected, a graphical representation of the control surface deflection is displayed.

7.0 STABILITY AND CONTROL COEFFICIENTS CALCULATION

This chapter describes the implementation of a method to determine an aircraft's longitudinal stability and control coefficients from its geometry. This work is based on Dr. Roskam's Aircraft Design Part 6 [7.1], which outlines the US Air Force Data Compendium's (DATCOM) method [2.5]. This chapter is devoted to the application of the DATCOM and other methods to a basic geometry aircraft operating in the subsonic regime. Although DATCOM addresses both the longitudinal and lateral coefficients, this work is limited to the longitudinal coefficients.

The result of this work is a program which estimates the stability and control coefficients of an aircraft, whose dimensions can be modified to view the effects on the coefficients. The program also offers a step by step solution of all the equations, the values used in them and the final result, as well as any graphs used in the solution. This not only helps in demonstrating the process but also validates the solution, and the implementation of the algorithms to solve any graphical data. Appendix 15 contains selected samples of the step by step solution.

7.1 Aircraft Design Constraints

DATCOM is a comprehensive document, describing many complex aircraft geometries at different speeds. To implement the entire document would be beyond the scope of a Master's thesis. Since this program is intended as a teaching aid for introductory courses in flight stability and control systems, an in depth investigation would exceed the scope of such a course. Students studying and designing complex transonic and supersonic aircraft should use existing CFD and other design tools. This program is intended solely as an introductory teaching tool.

Since the method presented in DATCOM is based largely on experimental results, interpolating beyond the range of the data could yield results with significant errors. For this reason the limitations of each step are programmed as constraints. Equations have been fitted to the graphical data presented in DATCOM and Airplane Design which is used to determine the stability and control coefficients using "Jandel Scientific Table Curve 3D". The functions fitted to these curves are presented in Appendix 16.

Figure 7.1 illustrates a typical aircraft under consideration, it is operating in the subsonic region where effects due to the compressibility of air may be neglected. The aircraft is flying at zero angle of attack. This does not imply that the wings and horizontal stabilizer are at zero angle of attack since they may be mounted on the fuselage at some angle of incidence.

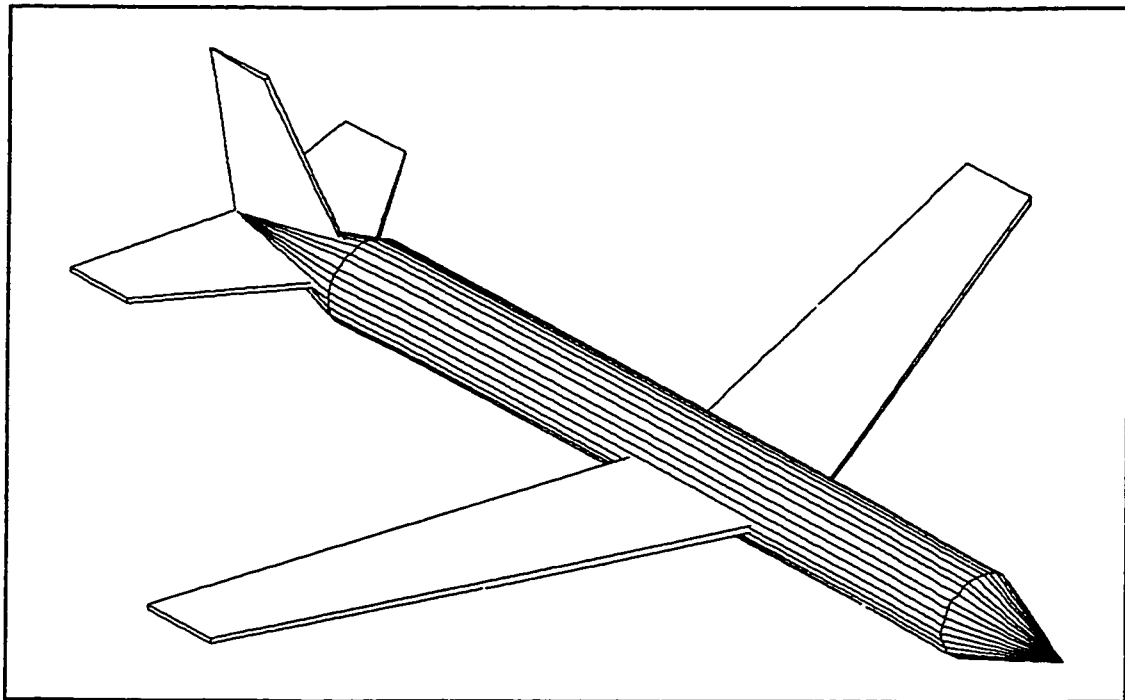


Figure 7.1 Typical aircraft under consideration.

7.1.1 Fuselage Geometry

The fuselage consists of a conical nose section followed by a cylindrical section and terminated by a conical tail section. All three fuselage sections are centred on the aircraft's centre line. The length of each section, as well as the diameter of the cylindrical section, may be entered as design parameters.

7.1.2 Wing Geometry

The wings are simple straight tapered wings located forward of the horizontal stabilizer. One may define the slope of the wing airfoil's lift curve as well as its zero lift angle of attack. The wing geometry is defined by the span, root chord, tip chord, and leading edge sweep angle as shown in Figure 7.2. From these, all other geometric parameters are calculated using Equations 7.1 to 7.12.

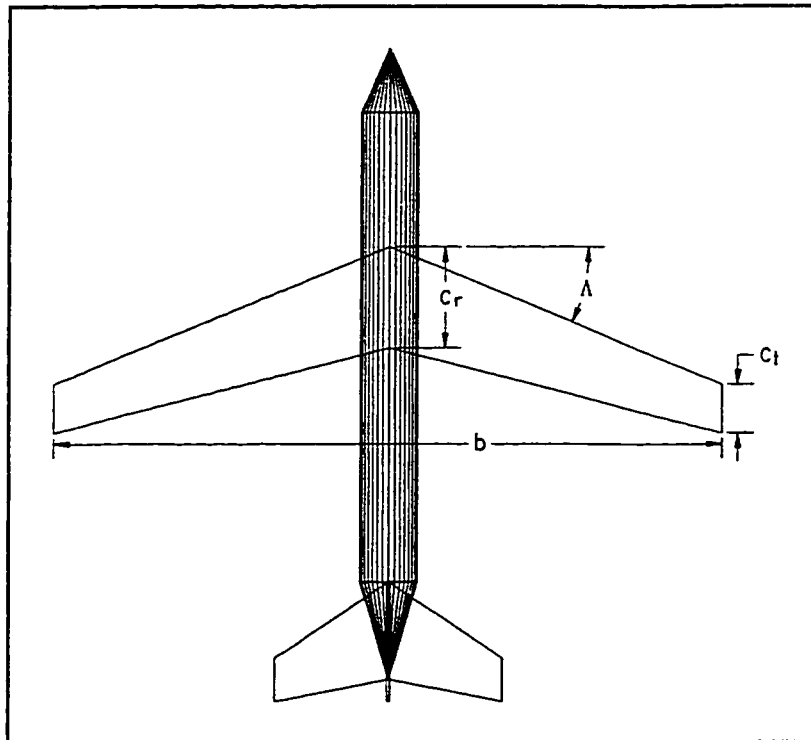


Figure 7.2 Wing geometry.

The taper ratio is the ratio of the tip chord to the root chord

$$\lambda = \frac{c_t}{c_r} \quad (7.1)$$

The projected wing area

$$S = \frac{b}{2} c_r (1 + \lambda) \quad (7.2)$$

Mean aerodynamic chord

$$\bar{c} = \frac{2(1 + \lambda + \lambda^2)}{3(1 + \lambda)} c_r \quad (7.3)$$

Aspect ratio describes the slenderness of a wing

$$A = \frac{b^2}{S} \quad (7.4)$$

Mid chord sweep angle

$$\Lambda_{mid} = \tan^{-1} \left(\tan \Lambda_{le} - \frac{c_r(c_r - 1)}{b(c_t + c_r - 1)} \right) \quad (7.5)$$

Quarter chord sweep angle

$$\Lambda_{1/4} = \tan^{-1} \left(\tan \Lambda_{le} - \frac{c_r(c_r - 1)}{2b(c_t + c_r - 1)} \right) \quad (7.6)$$

Mean aerodynamic chord span location

$$y_{mac} = \frac{b(1 + 2\lambda)}{6(1 + \lambda)} \quad (7.7)$$

The taper ratio of the exposed wing

$$\lambda_{\text{exp}} = \frac{c_t}{c_{fus}} \quad (7.8)$$

Mean aerodynamic chord of the exposed wing

$$\bar{c}_{\text{exp}} = \frac{2(1 + \lambda_{\text{exp}} + \lambda_{\text{exp}}^2)}{3(1 + \lambda_{\text{exp}})} c_f \quad (7.9)$$

The location of the mean aerodynamic chord of the exposed wing

$$y_{\text{mac}} = \frac{b(1 + 2\lambda_{\text{exp}})}{6(1 + \lambda_{\text{exp}})} \quad (7.10)$$

Chord at the fuselage

$$c_f = c_r - (c_r - c_t) \frac{d_f}{b} \quad (7.11)$$

The wetted wing area is defined as the total wing exposed surface area

$$S_{\text{wet}} = (b - d_f) \left(1 + \frac{c_t}{c_f} \right) c_f \quad (7.12)$$

7.1.3 Horizontal Stabilizer Geometry

The horizontal stabilizer is defined in the same manner as the wings. A subscript h is added to all applicable variables to indicate that they refer to the horizontal stabilizer.

7.1.4 Vertical Stabilizer Geometry

Figure 7.3 illustrates a typical vertical stabilizer. Its characteristics can be found using the same equations as the wing and horizontal stabilizer, using twice the span and then dividing the resulting projected and wetted areas by two. To avoid confusion a subscript v is added to all applicable variables to indicate that they refer to the vertical stabilizer.

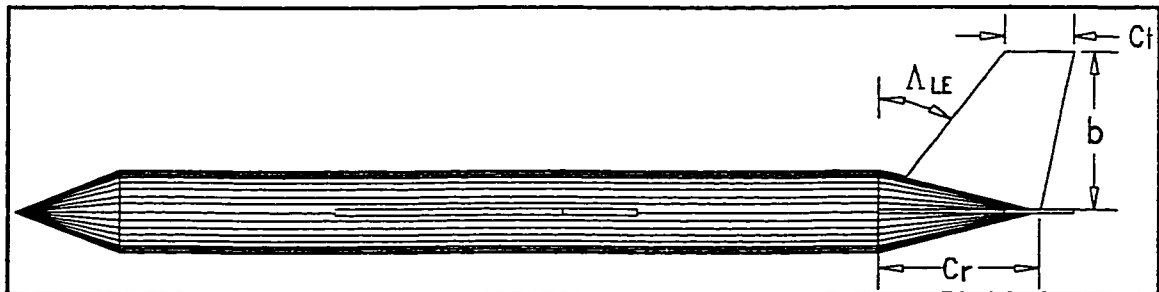


Figure 7.3 Vertical stabilizer geometry.

7.2 Aerodynamics

The stability and control coefficients are assumed to be the sum of the coefficients due to the individual components. The first step is to determine the stability coefficients for the wings, horizontal, and vertical stabilizer.

7.2.1 Lift

The wing's aerodynamics depend both on its geometry and the characteristics of its airfoil. Figure 7.4 depicts a typical airfoil's lift curve against the angle of attack. It consists of a linear portion followed by a nonlinear portion and finally a maximum lift angle of attack. Increasing the angle of attack beyond this point of maximum lift will result in the airflow detaching from the airfoil, causing it to

stall. For this work it is assumed that the airfoil is operating in the linear range, therefore the only quantities needed to describe the airfoil's characteristics are the zero lift angle of attack and the slope of the lift curve.

Theoretical studies have shown that the maximum theoretical slope of an airfoil's lift curve is 2π /radian.

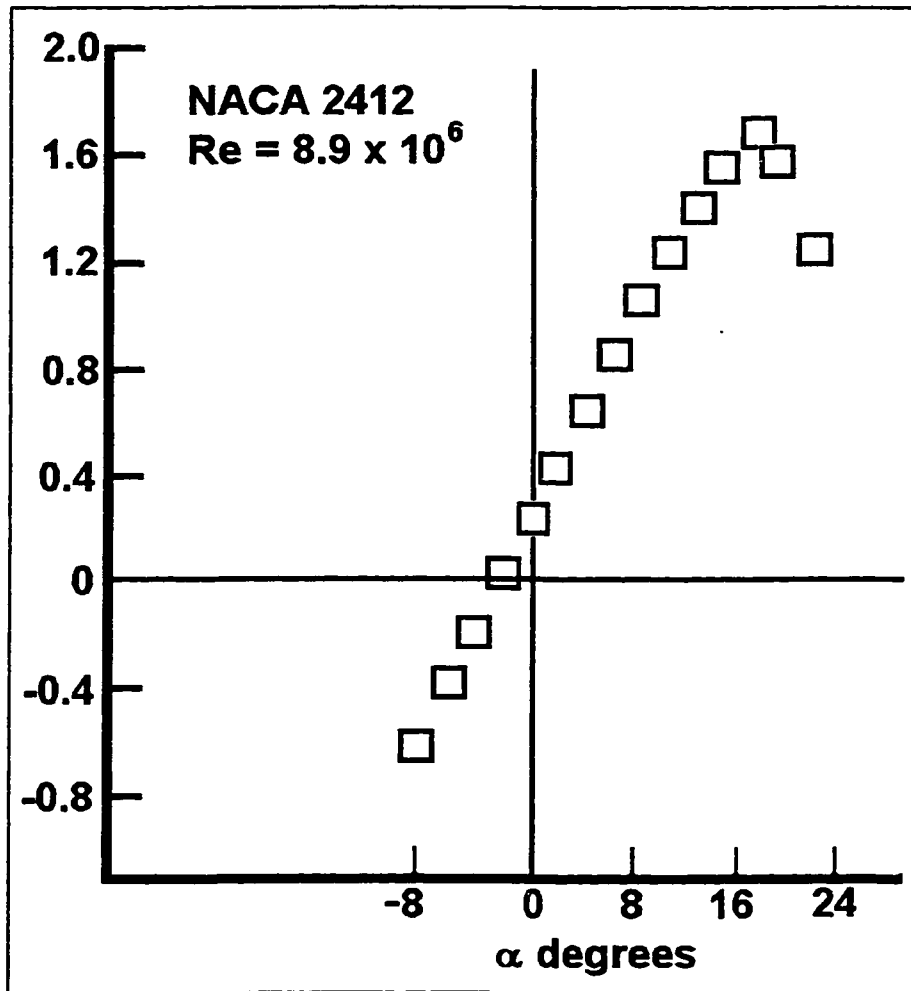


Figure 7.4 Typical airfoil lift curve [7.2].

7.2.1.1 Wing Lift Characteristics

Given the slope of the airfoil's lift curve, the slope of the wing's lift will depend on the aspect ratio and the sweep of the wing, according to Equation 7.13:

$$C_{L\alpha_w} = \frac{2\pi A}{2 + \sqrt{\left(\frac{A}{k}\right)^2 (1 + \tan^2 \Lambda_{mid}) + 4}} \quad (7.13)$$

where k is the ratio of the airfoil's lift curve slope to the ideal slope of 2π :

$$k = \frac{C_l}{2\pi} \quad (7.14)$$

From these equations, it is apparent that greater lifts can be achieved through a large aspect ratio with minimal sweep, as seen in gliders.

For a straight (untwisted) wing, the zero lift angle of attack is equal to that of its airfoil.

$$\alpha_{0L_w} = \alpha_{0l} \quad (7.15)$$

At zero angle of attack, the wing's lift is given by equation 7.16.

$$C_{L0_w} = -\alpha_{0L_w} C_{L\alpha_w} \quad (7.16)$$

7.2.1.2 Horizontal Stabilizer Lift Characteristics

The horizontal stabilizer's lift curve slope and the zero lift angle of attack are found in the same manner as the wing's. Where confusion might exist between the variables, the subscripts h or w is added to the variables.

7.2.1.3 Fuselage Lift Characteristics

The fuselage, being symmetrical in this case, generates no lift at zero angle of attack. It does however, interfere with the flow over the wings and will affect the performance of the aircraft. This will be taken into account by determining the interference due to the fuselage on the wing.

7.2.2 Drag

The drag which an aircraft experiences can be divided into two categories. The profile drag which is the drag due to the geometry of the aircraft passing through the air, and the induced drag which results from the generation of lift. A major source of induced drag results from the wing tip vortices.

Wing tip vortices occur from a lift-producing wing when the high pressure below the wing 'curls' over the wing tip to the low pressure above it. The result is a trailing vortex behind each wing tip as the aircraft flies.

Two common methods to reduce the wing tip vortices and thus the induced drag is to use a high aspect ratio wing or to install winglets at the tip of the wings as shown in Figure 7.5.



Figure 7.5 Boeing 747-400 with winglets.

7.2.2.1 Wing Profile Drag Characteristics

The wing's zero lift profile drag coefficient may be estimated from Equation 7.17.

$$C_{D_{o_w}} = R_{wf} R_{LS} C_{fw} (1 + L'(t/c) + 100(t/c)^4) \frac{S_{wet_w}}{S_w} \quad (7.17)$$

R_{wf} is a wing fuselage interference coefficients which depends on the Reynolds number of the fuselage, given by Equation 7.18, and the mach number.

$$R_{N_f} = \frac{\rho U l_f}{\mu} \quad (7.18)$$

Figure 7.6 gives the wing fuselage interference factor. Since the aircraft in question is flying at low mach numbers, the curve corresponding to $M=0.25$ is used.

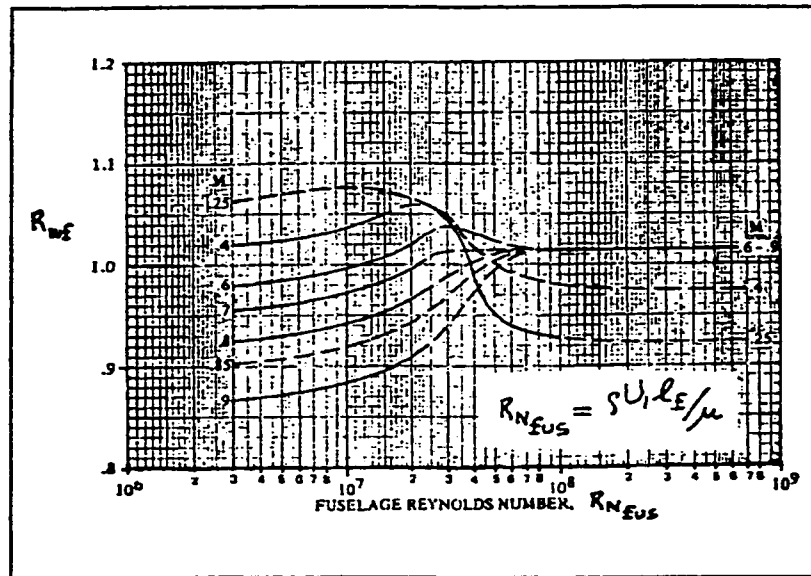


Figure 7.6 Wing fuselage interference factor. [2.5]

R_{LS} is a factor that takes into account the sweep of the wing. A swept wing will produce less lift than an unswept wing since the air hitting the leading edge of a swept wing will be partially deflected to the tip and off the wing as opposed to over it. Figure 7.7 gives the correction factor as a function of the cosine of the sweep at the point of maximum airfoil thickness, and the Mach number. Since the aircraft is flying at low Mach numbers, the curve corresponding to Mach=0.25 is used.

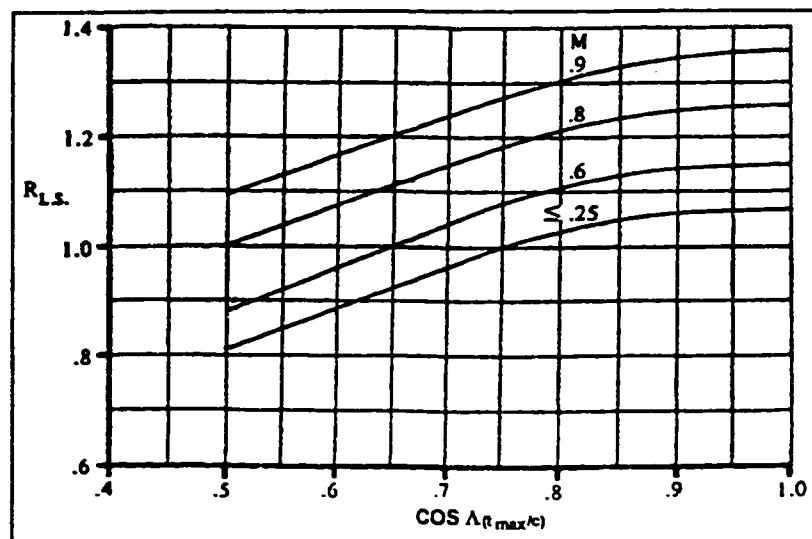


Figure 7.7 Lift correction factor for a swept wing. [2.5]

C_{fw} is the turbulent friction coefficient for a flow over a flat plate. Its value will depend on both the Reynolds number based on the mean aerodynamic chord of the exposed wing given by Equation 7.19, and the mach number using Figure 7.8.

$$R_{N_w} = \frac{\rho U \bar{c}_{we}}{\mu} \quad (7.19)$$

Since only low mach numbers are being considered, the curve corresponding to Mach=0 is used.

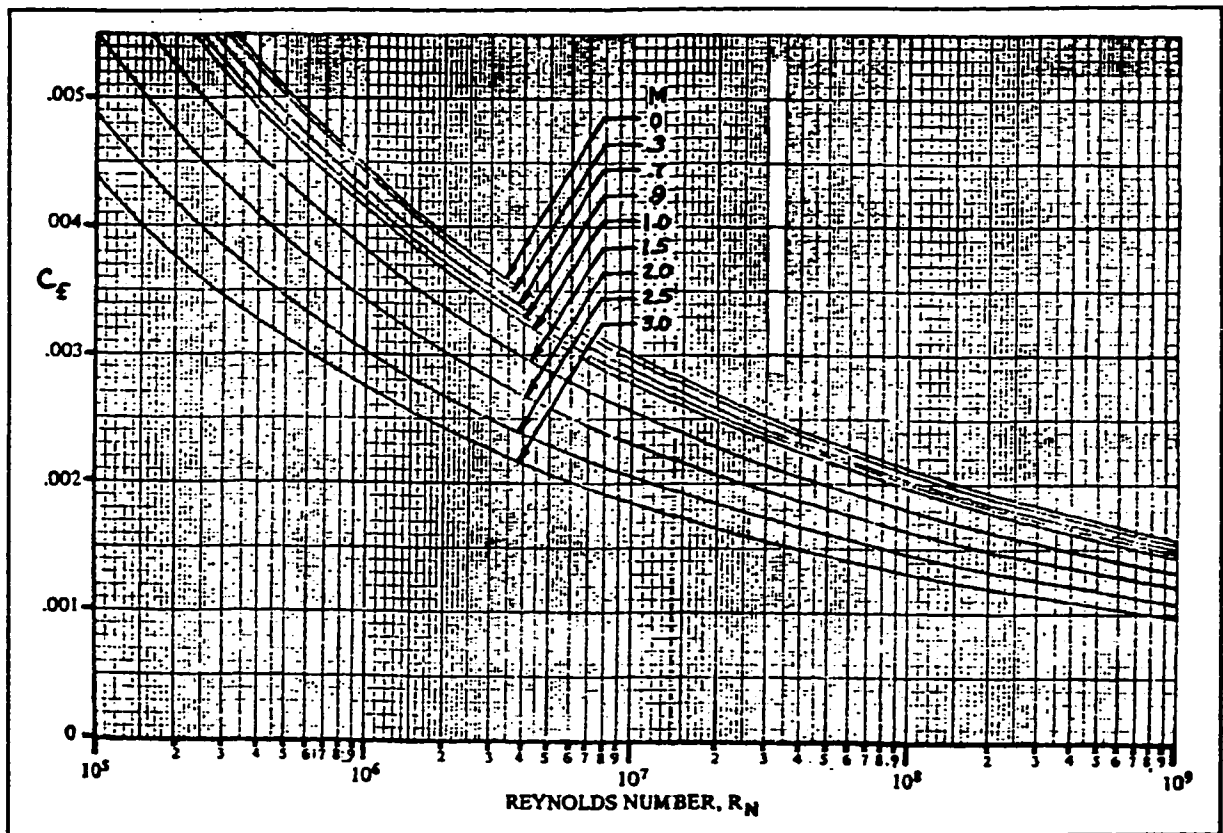


Figure 7.8 Skin friction coefficient. [2.5]

L' is a parameter based on the location of the airfoil's point of maximum thickness and is given in Figure 7.9. (t/c) is the ratio of the airfoil's maximum thickness to its chord, illustrated in Figure 7.9.

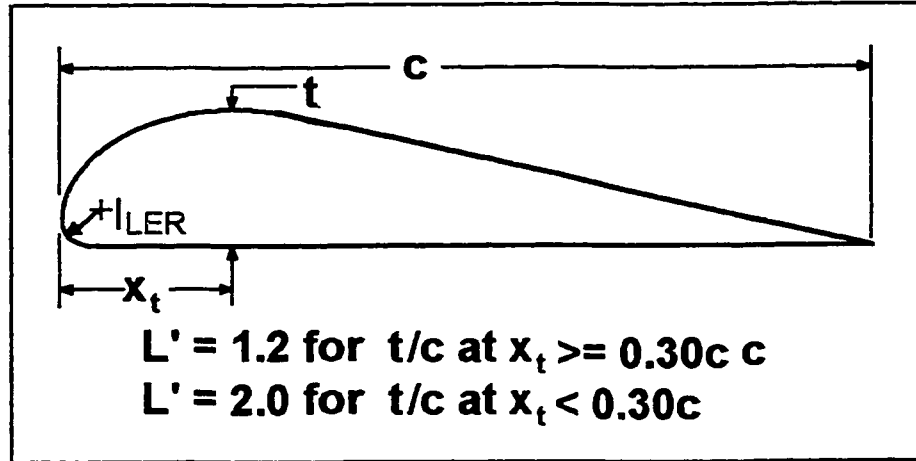


Figure 7.9 Airfoil thickness location parameter. [2.5]

7.2.2.2 Wing Induced Drag Characteristics

For an untwisted wing, the induced drag, or drag due to lift, is found from equation 7.20.

$$C_{D_{L_w}} = \frac{C_{L_w}^2}{\pi A e} \quad (7.20)$$

Since the aircraft in question is flying at zero angle of attack, the angle of attack of the wing is its angle of incidence on the aircraft. The lift coefficient of the wing at zero aircraft angle of attack is given by equation 7.21.

$$C_{L_w} = C_{L_{0_w}} + C_{L_{\alpha_w}} i_w \quad (7.21)$$

The wing span efficiency factor, e , is given by equation 7.22.

$$e = \frac{1.1 C_{L\alpha_w} / A}{R(C_{L\alpha_w} / A) + (1 - R) \pi} \quad (7.22)$$

where R is the leading edge suction parameter given by figure 7.10. The Reynolds number used in Figure 7.10 is based on the leading edge radius of the wing:

$$R_{l_{LER}} = \frac{\rho u l_{LER}}{\mu} \quad (7.23)$$

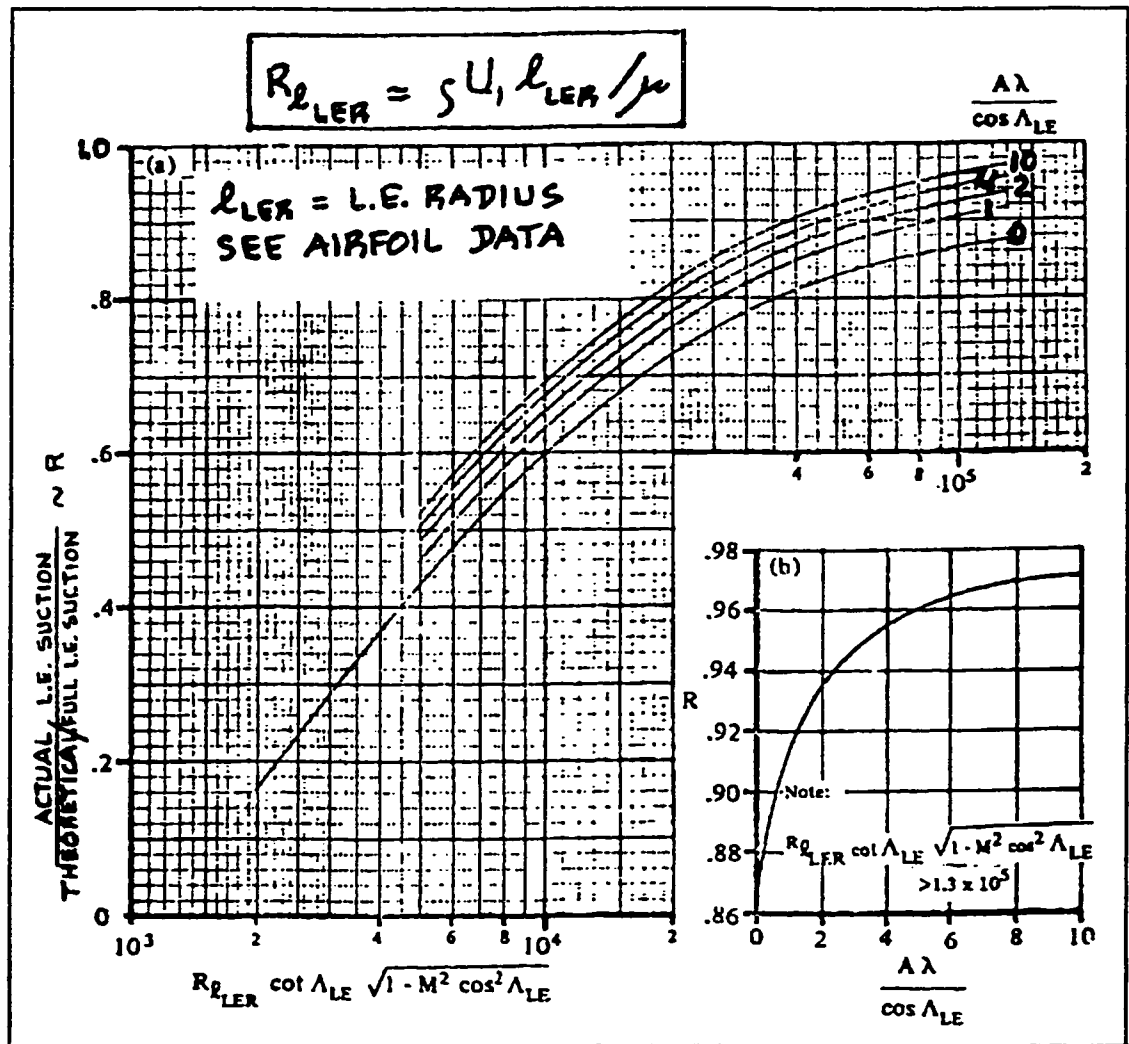


Figure 7.10 Leading edge suction parameter. [2.5]

The chain rule is used to obtain the change in drag due to angle of attack about a specific point. The derivative is taken with respect to the lift and multiplied by the slope of the lift curve:

$$C_{D_{\alpha_w}} = \frac{dC_{D_{L_w}}}{dC_{L_w}} \frac{dC_{L_w}}{d\alpha_w} = \frac{2C_{L_w}}{\pi A e} C_{L_{\alpha_w}} \quad (7.24)$$

7.2.2.3 Horizontal Stabilizer Drag Characteristics

The drag characteristics of the horizontal stabilizer are found in the same manner as for the wing. When the possibility of confusion exists between the variables, the subscripts h or w are added to the variables.

7.2.2.4 Vertical Stabilizer Drag Characteristics

The aircraft under consideration has a vertical stabilizer with a symmetric airfoil mounted with zero angle of incidence. During normal flight with no sideslip, the vertical stabilizer does not produce any lift (side force). For this reason, only the profile drag of the vertical stabilizer is considered. It is found in the same manner as for the wing. When the possibility of confusion exists between the variables, the subscripts v or w are added to the variables.

7.2.2.5 Fuselage Drag Characteristics

The aircraft in question, shown in Figure 7.11 has a circular cross section symmetrical about the X body axis, and coming to a point at both the nose and the tail. Given this, the fuselage drag at zero angle of attack is estimated from

Equation 7.25.

$$C_{Do_f} = R_{wf} C_{ff} \left(1 + 60 \left(\frac{d_f}{l_f} \right)^3 + 0.0025 \left(\frac{l_f}{d_f} \right) \right) \frac{S_{wet_f}}{S} \quad (7.25)$$

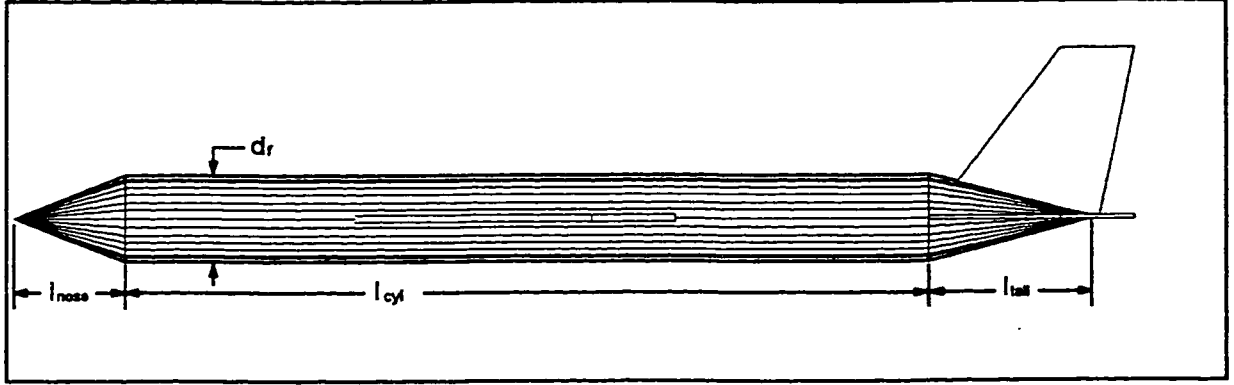


Figure 7.11 Fuselage dimensions.

The wing fuselage interference factor R_{wf} is found in the same way as for the wing, and using Figure 7.6 The skin friction coefficient C_{ff} is found using Figure 7.8 with the Reynolds number based on the fuselage length.

$$R_{N_{fus}} = \frac{\rho u l_f}{\mu} \quad (7.26)$$

The wetted area S_{wet_f} is the total exposed surface area of the fuselage. For the wing being considered, the wetted area is given by equation 7.27.

$$S_{wet_f} = \pi d_f \left(l_{cyl} - \frac{(l_{nose} + l_{tail})}{2} \right) \quad (7.27)$$

7.2.3 Pitch

In order to describe any moment, it must be clear about which point

it is acting. Two key points on an airfoil which must be understood are the centre of pressure and the aerodynamic centre.

The centre of pressure is the point on the airfoil at which the net force due to the pressure distribution about the airfoil acts. This point may move along the airfoil as the angle of attack and pressure distribution changes. The pitching moment about the centre of pressure will be zero.

The aerodynamic centre is a fixed point on the airfoil about which the pitching moment remains constant with changes in the angle of attack. This constant pitching moment will be nonzero except in the case of a symmetrical airfoil. The aerodynamic centre is usually given for an airfoil as a fraction of its chord. It usually lies at about the 25% chord position for most airfoils.

It follows that if the pitching moment remains constant about the aerodynamic centre, then as the net force on the airfoil increases, the centre of pressure moves towards the aerodynamic centre. As the net force decreases, the centre of pressure moves away from the aerodynamic centre. This is expressed by Equation 7.28.

$$c_l(X_{cp} - X_{ac}) = \text{constant} \quad (7.28)$$

For an arbitrary reference point on an airfoil, the pitching moment coefficient is given by Equation 7.29

$$c_m = c_{m_o} + c_l(X_{ref} - X_{ac}) / c \quad (7.29)$$

By differentiating this equation with respect to c_l , the variation in the pitching moment coefficient with the lift coefficient is found.

$$\frac{dc_m}{dc_l} = (X_{ref} - X_{ac}) / c = \bar{X}_{ref} - \bar{X}_{ac} \quad (7.30)$$

To simplify calculations, the reference point is taken at the aerodynamic centre, resulting in a constant pitching moment and a lifting force which does not change position regardless of the angle

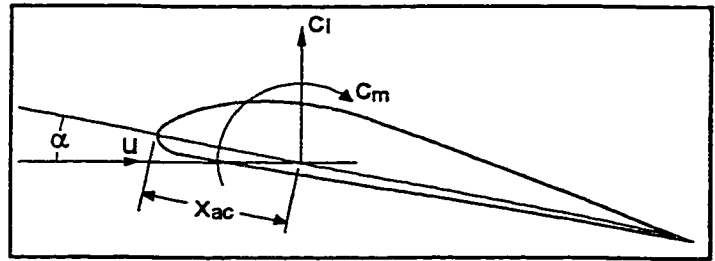


Figure 7.12 Lift and pitching coefficient about the aerodynamic center.

of attack. This is achieved as shown in Figure 7.12. The net force acting at the centre of pressure is replaced by the same force and a couple acting at the aerodynamic centre. Figure 7.13 shows a typical lift and pitching moment curve for a typical (NACA 2412) airfoil. Notice that in the linear range, the pitching moment coefficient about the aerodynamic centre remains almost constant. The aerodynamic centre is typically located very close to the 25% chord position and is often taken as such.

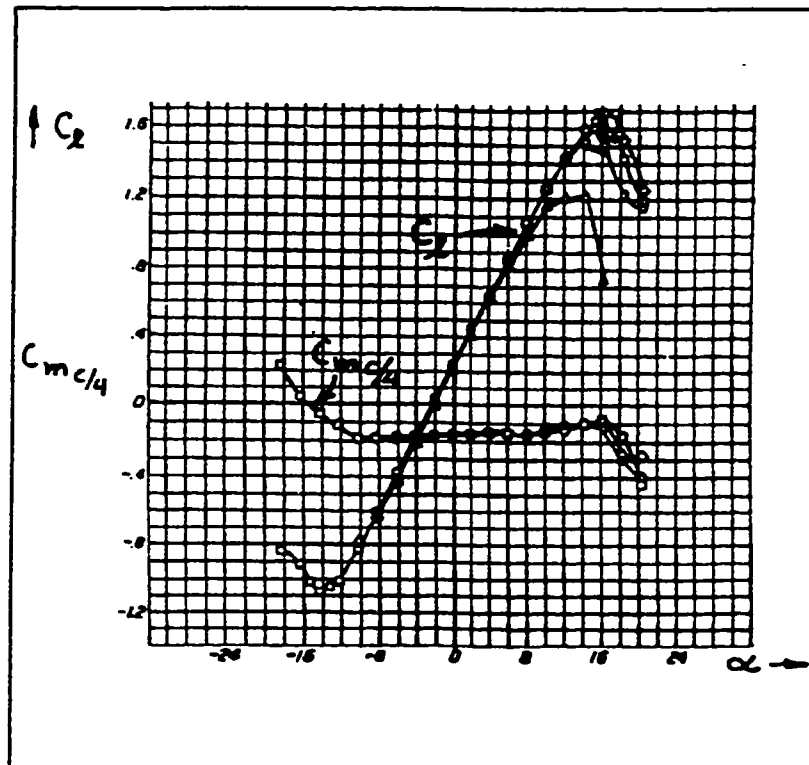


Figure 7.13 NACA 2412 Lift and pitching moment coefficients. [7.1]

7.3 Aircraft Aerodynamics

Having determined the aerodynamic characteristics of the major components of the aircraft, it is possible to calculate the stability and control coefficients for the entire aircraft. Table 7.1 list the coefficients which according to Roskam's Airplane Design [7.1], are sufficient to describe most aircraft's performance in a given flight condition. In this work, only the longitudinal coefficients : lift, drag, and pitch, will be considered.

<u>Lift</u>	<u>Drag</u>	<u>Pitch</u>	<u>Roll</u>	<u>Yaw</u>	<u>Side Force</u>
C_{L_o}	C_{D_o}	C_{m_o}	C_{l_o}	C_{n_o}	C_{Y_o}
C_{L_α}	C_{D_α}	C_{m_α}	C_{l_β}	C_{n_β}	C_{Y_β}
$C_{L_{\dot{\alpha}}}$		$C_{m_{\dot{\alpha}}}$	C_{l_p}	C_{n_p}	C_{Y_p}
C_{L_q}		C_{m_q}	C_{l_r}	C_{n_r}	C_{Y_r}
$C_{L_{\delta_e}}$		$C_{m_{\delta_e}}$	$C_{l_{\delta_a}}$	$C_{n_{\delta_a}}$	$C_{Y_{\delta_r}}$
			$C_{l_{\delta_r}}$		

Table 7.1 Stability and control coefficients.

7.3.1 Aircraft Lift Coefficient

The aircraft lift is due mainly to the wing and horizontal stabilizer. Since the fuselage is symmetrical about the X body axis and the aircraft is flying at zero angle of attack, the fuselage itself does not contribute to the lift but indirectly

affects it by altering the airflow over the wings and horizontal stabilizer.

7.3.1.1 Aircraft Lift Coefficient at Zero Angle of Attack

The lift acting on the aircraft is due to the wings and the horizontal stabilizer. When the air passes around the wing, it is deflected slightly downward as it leaves the wing. The result of this is that the air does not hit the horizontal stabilizer in the same direction as the free stream velocity but at a slight downward angle ϵ_h . This is illustrated in Figure 7.14. This angle through which the air deflects is called the downwash angle and has the effect of reducing the horizontal stabilizer's angle of attack. The magnitude of the downwash angle will depend on the angle of attack of the aircraft, the wing geometry and the proximity of the horizontal stabilizer to the wing. For preliminary design work, Roskam assumes that the downwash angle is negligible.

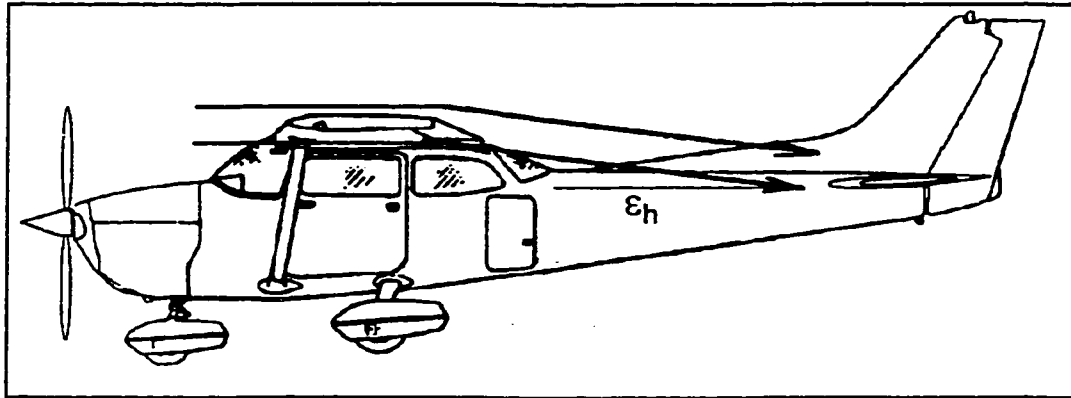


Figure 7.14 Horizontal stabilizer downwash angle (exaggerated).

If the horizontal stabilizer lies in the path of the air leaving the propeller, then the velocity of the air hitting the horizontal stabilizer would be higher than the free stream velocity, increasing the dynamic pressure on it. To take this into account, the ratio of the free stream dynamic pressure and the dynamic pressure at the horizontal stabilizer is given by Equation 7.31.

$$\eta_h = \frac{\bar{q}_h}{\bar{q}} \quad (7.31)$$

Taking these factors into account, the aircraft lift coefficient at zero angle of attack may be estimated with Equation 7.32.

$$C_{L_o} = (i_w - \alpha_{L o_w}) C_{L_{\alpha_w}} + (i_h - \varepsilon_{o_h} - \alpha_{L o_h}) \eta_h \left(\frac{S_h}{S} \right) C_{L_{\alpha_h}} \quad (7.32)$$

7.3.1.2 Aircraft Lift Curve Slope

The aircraft's lift curve slope is dominated by the lift curve slope of the wing and horizontal stabilizer. Two key points to consider are the effects of the downwash angle and the interference of the fuselage on the airflow. Considering these factors, the aircraft's lift curve slope may be estimated with Equation 7.33.

$$C_{l_\alpha} = K_{wf} C_{L_{\alpha_w}} + C_{L_{\alpha_h}} \eta_h \left(\frac{S_h}{S} \right) \left(1 - \frac{d\varepsilon_h}{d\alpha} \right) \quad (7.33)$$

Where K_{wf} is the wing fuselage interference factor given by Equation 7.34.

$$K_{wf} = 1 + 0.025 \left(\frac{d_f}{b} \right) - 0.25 \left(\frac{d_f}{b} \right)^2 \quad (7.34)$$

The derivative of the downwash angle with respect to the angle of attack is estimated from Equation 7.35.

$$\frac{d\varepsilon_h}{d\alpha} = 4.44 \left(K_A K_\lambda K_h \sqrt{\cos \Lambda_{c/4}} \right)^{1.19} \quad (7.35)$$

where K_A , K_λ and K_h depend on the wing geometry and the position of the horizontal stabilizer relative to the wing, and are given by Equations 7.36 to 7.38. l_h is the horizontal distance from the wing tip quarter chord point to the horizontal stabilizer

root quarter chord point. In this work, the vertical distance between the wing and the horizontal stabilizer h_h is zero.

$$K_A = \frac{1}{A} - \frac{1}{1+A^{1.7}} \quad (7.36)$$

$$K_\lambda = \frac{10-3\lambda}{7} \quad (7.37)$$

$$K_h = \frac{1 - \frac{h_h}{b}}{\sqrt[3]{2 \frac{l_h}{b}}} \quad (7.38)$$

7.3.1.3 Aircraft Zero Lift Angle of Attack

Knowing the aircraft lift curve slope and the lift at zero angle of attack, the aircraft zero lift angle of attack can be calculated using Equation 7.39.

$$\alpha_o = -\frac{C_{L_0}}{C_{l_\alpha}} \quad (7.39)$$

7.3.1.4 Aircraft Lift Coefficient Due to Rate of Change of Angle of Attack

As the aircraft's angle of attack increases, the downwash from the wing hitting the horizontal stabilizer will also increase, reducing its angle of attack. Since the angle of attack is changing as the aircraft moves forward, there will be a delay for the new downwash angle to take effect as it leaves the trailing edge on the wing and reaches the leading edge of the horizontal stabilizer. Therefore during the

transient angle of attack, the horizontal stabilizer will actually be producing more lift than would be associated with the aircraft's instantaneous angle of attack. This is estimated by Equation 7.40.

$$C_{L_{\alpha}} = 2C_{L_{\alpha_h}} \eta_h \bar{V}_h \left(\frac{d\varepsilon_h}{d\alpha} \right) \quad (7.40)$$

where V_h is the horizontal tail volume coefficient as shown in Figure 7.15, given by Equation 7.41.

$$\bar{V}_h = \frac{(X_{ac_h} - X_{cg})}{\bar{c}} \left(\frac{S_h}{S} \right) \quad (7.41)$$

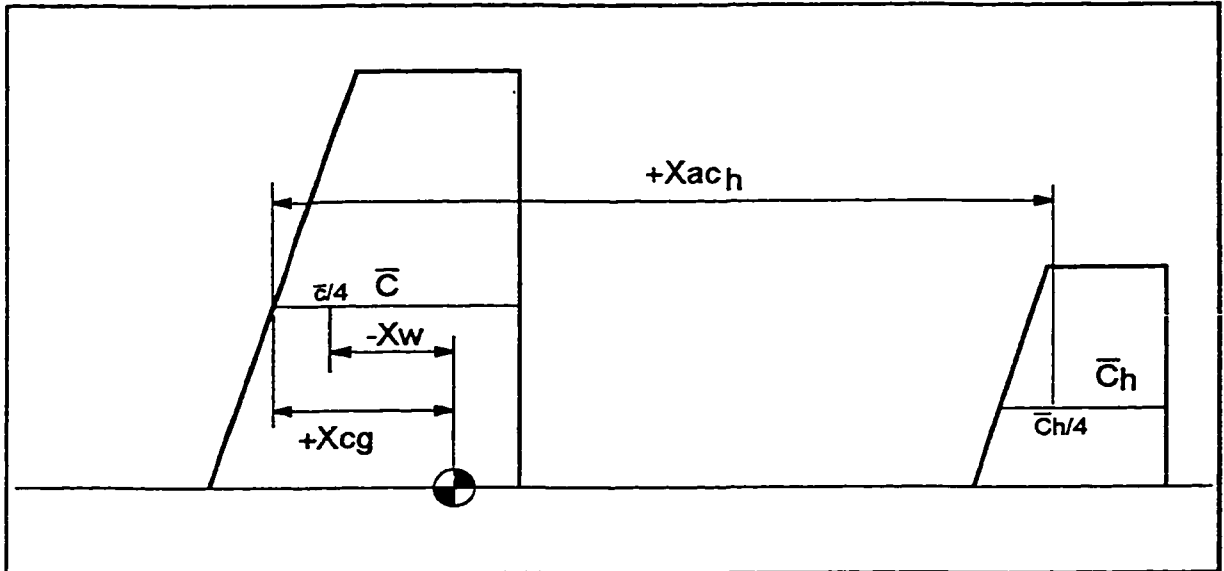


Figure 7.15 Horizontal stabilizer and center of gravity location.

7.3.1.5 Aircraft Lift Coefficient Due to Pitch Rate

The lift coefficient due to the pitch rate may be estimated as the sum of the coefficients due to the wing and the horizontal stabilizer.

$$C_{L_q} = C_{L_{\alpha_w}} + C_{L_{\alpha_h}} \quad (7.42)$$

The wing component is estimated from Equation 7.43,

$$C_{L_{\alpha_w}} = \left(0.5 + \frac{2X_w}{\bar{c}} \right) C_{l_{\alpha_w}} \quad (7.43)$$

where x_w is the distance from the centre of gravity to the quarter chord point on the mean geometric chord as shown in Figure 7.15.

The horizontal stabilizer component is estimated from Equation 7.44.

$$C_{L_{\alpha_h}} = 2C_{L_{\alpha_h}} \eta_h \bar{V}_h \quad (7.44)$$

7.3.1.6 Aircraft Lift Coefficient Due to Elevator Deflection

The effect of elevator deflection on the aircraft's lift coefficient is determined by calculating an equivalent change in angle attack of the horizontal stabilizer, which would give the same effect as the elevator deflection. This equivalent angle of attack is then multiplied by the slope of the horizontal stabilizer lift curve to determine the effect of the elevator. Mathematically this is described by Equation 7.45.

$$\frac{dC_L}{d\delta_e} = \frac{d\alpha_h}{d\delta_e} \frac{dC_L}{d\alpha_h} \quad (7.45)$$

or

$$C_{L_{\delta_e}} = \alpha_{h_{\delta_e}} C_{L_{\alpha_h}} \quad (7.46)$$

$\alpha_{L\delta e}$ is a function of the elevator geometry based on Equation 7.47.

$$\alpha_{\delta e} = K_b \left(\frac{c_{l_\delta}}{(c_{l_\delta})_{theory}} \right) (c_{l_\delta})_{theory} k' \frac{1}{c_{l_{\alpha h}}} \left(\frac{(\alpha_\delta)_{c_L}}{(\alpha_\delta)_{c_l}} \right) \quad (7.47)$$

K_b is a factor which takes into account the width of the elevator, its spanwise location, and the taper ratio of the horizontal stabilizer. It is found using figure 7.16. $(c_{l_\delta})_{theory}$ describes the theoretical change in lift of an airfoil as a function of its thickness and the size of the elevator. It is found using Figure 7.17. From this figure it is noted that the elevator is restricted to less than half the chord and the airfoil thickness is restricted to values up to 0.17. $C_{l_\delta}/(c_{l_\delta})_{theory}$ corrects $(c_{l_\delta})_{theory}$ based on the airfoil's lift curve slope and the size of the elevator, and is found using Figure 7.18. From this figure, it is noted that the elevator is restricted to less than half the chord. k' is a constant to account for nonlinearities during large elevator deflections. Since this work is only considering small deflections, this value is assumed to be one. $(\alpha_\delta)_{c_L}/(\alpha_\delta)_{c_l}$ takes into account the aspect ratio and the size of the elevator to determine the effect of the three dimensional wing from the airfoil's results. Figure 7.19 is used to find its value.

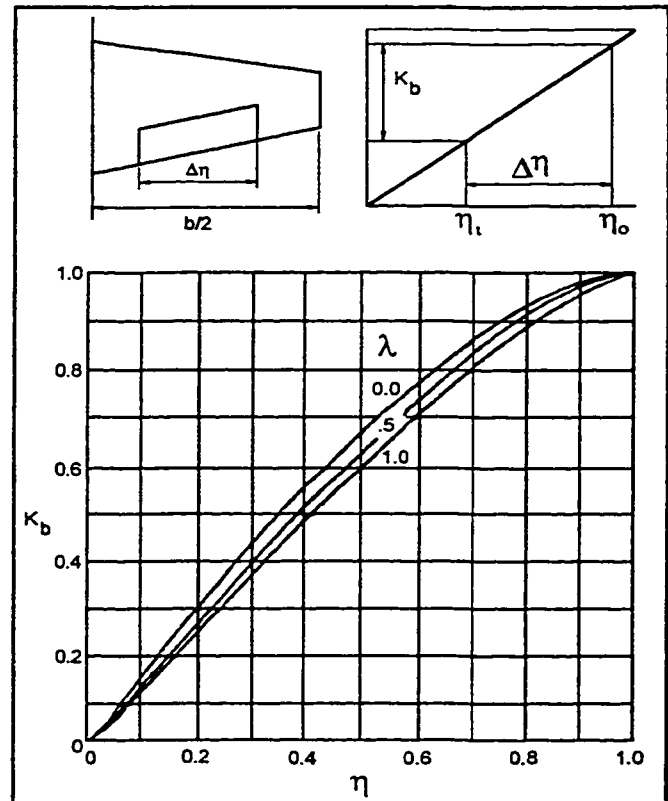


Figure 7.16 Effect of taper ratio and flap span [2.5]

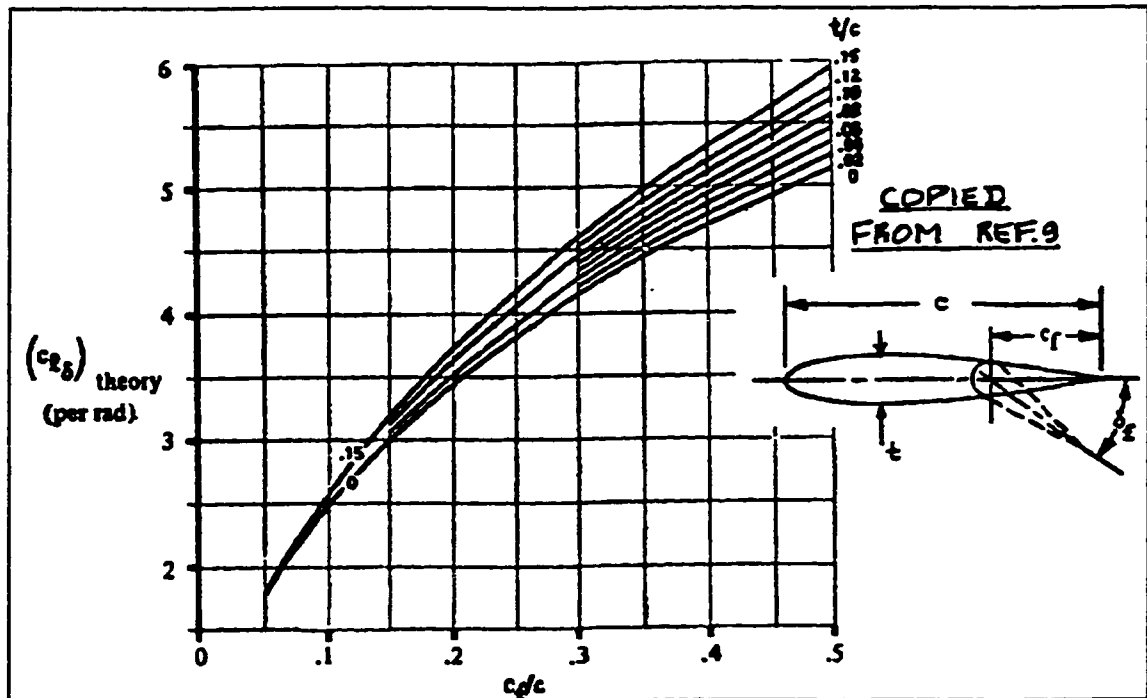


Figure 7.17 Plain flap lift effectiveness. [2.5]

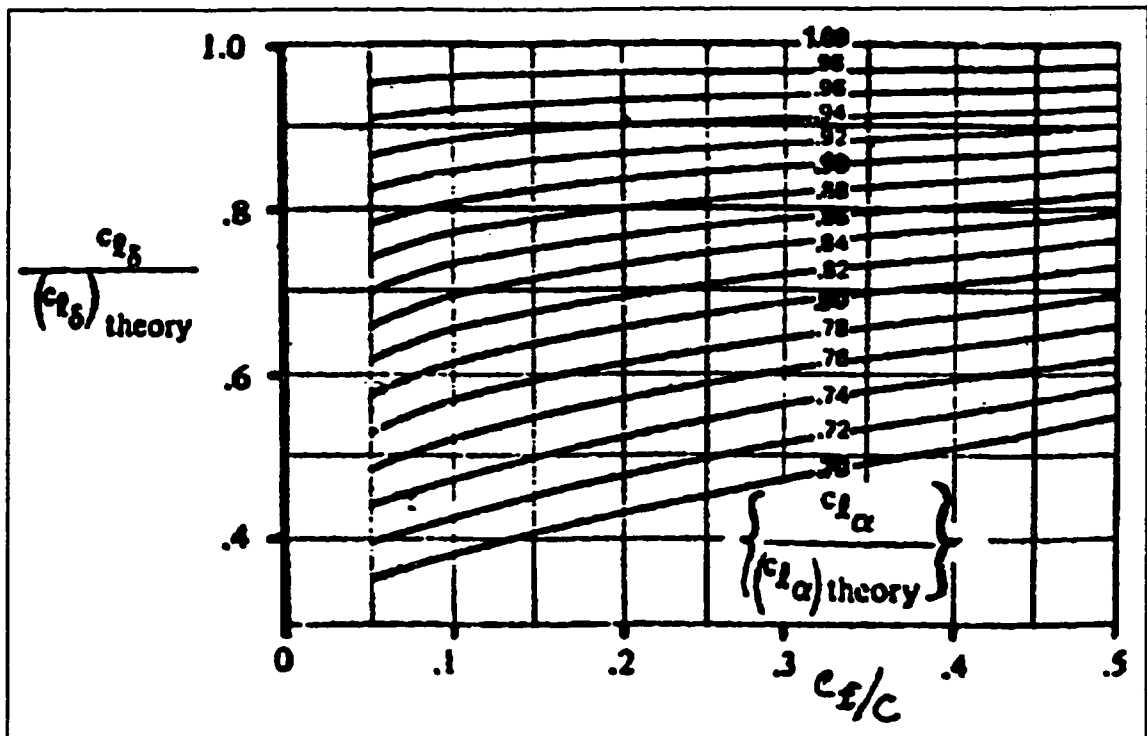


Figure 7.18 Plain flap lift correction factor. [2.5]

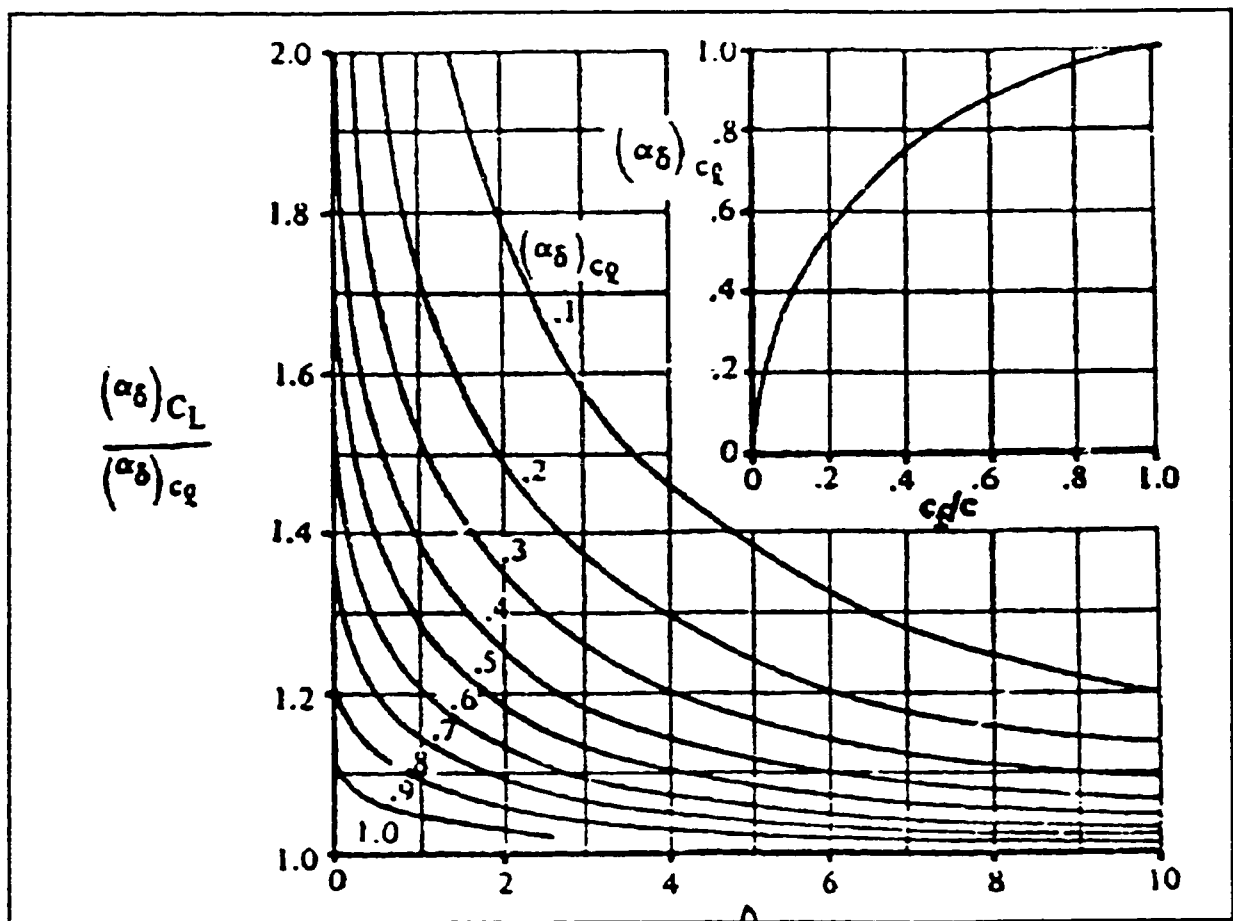


Figure 7.19 Effect of aspect ratio and flap chord ratio on three dimensional flap effectiveness. [2.5]

7.3.2 Aircraft Drag

The drag is due primarily due to the aircraft's profile drag and induced drag. Although rudder, elevator, and aileron deflections will create some additional drag, this drag is negligible compared to the overall drag and is neglected during initial calculations.

7.3.2.1 Aircraft Drag at Zero Angle of Attack

At zero angle of attack, the drag coefficient is estimated as the sum of the zero lift drag coefficients of the fuselage, wings, horizontal and vertical stabilizer, and the drag due to lift of the wings and vertical stabilizer. The wing and horizontal stabilizer drag must include the drag due to the angle of incidence at which they are mounted.

$$C_{D_o} = C_{D_{o_f}} + C_{D_{o_w}} + C_{D_{L_w}} + C_{D_{o_h}} + C_{D_{L_h}} + C_{D_{o_v}} \quad (7.48)$$

7.3.2.2 Aircraft Drag at Angle of Attack

At a given angle of attack, the slope of the drag curve is estimated as the sum of the slope of the wing's and horizontal stabilizer's drag curve.

$$C_{D_\alpha} = C_{D_{\alpha_w}} + C_{D_{\alpha_h}} \quad (7.49)$$

7.3.3 Aircraft Pitch Moment Coefficient

The aircraft pitching moment coefficient may be estimated as the sum of the individual moments due to the wing fuselage combination and the horizontal stabilizer. The reference point about which the moments are taken for the aircraft is its centre of gravity.

7.3.3.1 Aircraft Pitch Moment Coefficient at Zero Lift.

At low mach numbers, the zero lift pitching moment coefficient is the sum of the coefficients due to the fuselage, wing and horizontal stabilizer at zero aircraft lift as expressed by Equation 7.50.

$$C_{m_{0L}} = C_{m_{0L_f}} + C_{m_{0L_w}} + C_{m_{0L_h}} \quad (7.50)$$

For the aircraft in question, with a fuselage symmetric about its centre line, the fuselage pitching contribution is estimated by Equation 7.51 and Figure 7.20.

$$C_{m_{0L_f}} = \frac{(k_2 - k_1)}{36.5 S \bar{c}} \sum_{j=1}^{j=13} (w_{f_j}^2 (i_w - \alpha_{0L_w}) \Delta x_j) \quad (7.51)$$

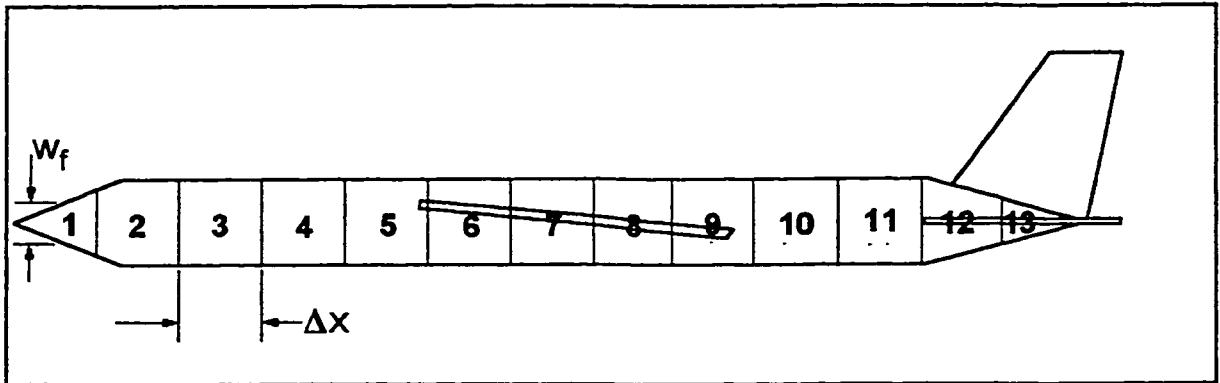


Figure 7.20 Fuselage segmentation. [7.1]

Where $(k_2 - k_1)$ is a factor from Figure 7.21 which depends on the slenderness of the fuselage.

The wing zero lift pitching moment coefficient was found in Section 7.2.3.1.

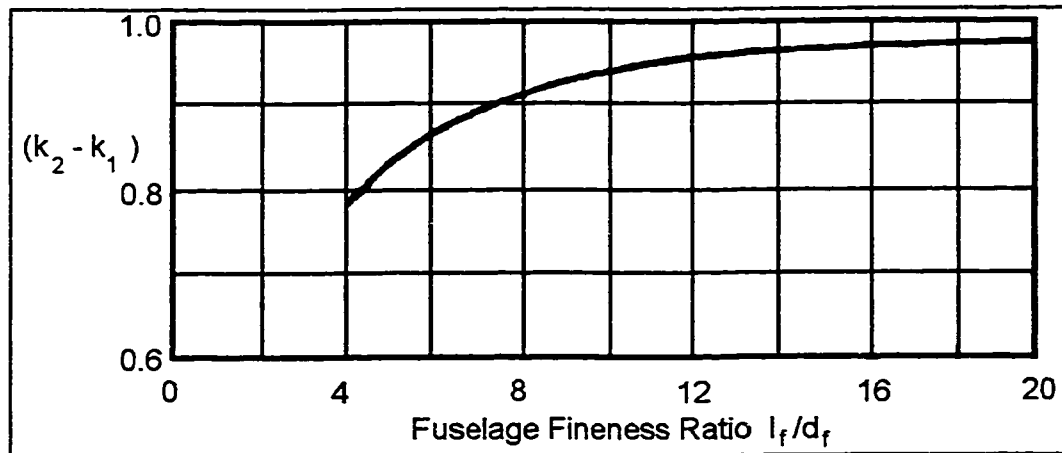


Figure 7.21 Effect of fuselage slenderness ratio. [2.5]

The horizontal stabilizer's contribution is found with Equation 7.52 and referring to Figure 7.15. The distances X are measured to the leading edge of the wing at its mean aerodynamic chord.

$$C_{m_{0L_h}} = -C_{L_{\alpha_h}} (\alpha_{0L} + i_h - \alpha_{0L_h}) \frac{X_{ac_h} - X_{cg}}{\bar{c}} \quad (7.52)$$

7.3.3.2 Aircraft Pitch Moment Coefficient Due to Angle of Attack

The pitching moment coefficient due to angle of attack is estimated as the sum of the contributions due to the wings, the fuselage and the horizontal stabilizer.

The wing contribution depends on the location of the wing's aerodynamic center relative to the aircraft's center of gravity.

$$C_{m_{\alpha w}} = -\frac{X_w}{\bar{c}} C_{L_{\alpha w}} \quad (7.53)$$

For simple fuselage geometries, an approximation by Gilruth [7.4] is used to estimate the fuselage contribution. Based on the position of the wing's quarter chord root position on the fuselage Figure 7.22 is used to determine a factor K_f used in Equation 7.54.

$$C_{m_{af}} = \frac{K_f d_f^2 l_f}{S \bar{c} C_{L_{avf}}} \quad (7.54)$$

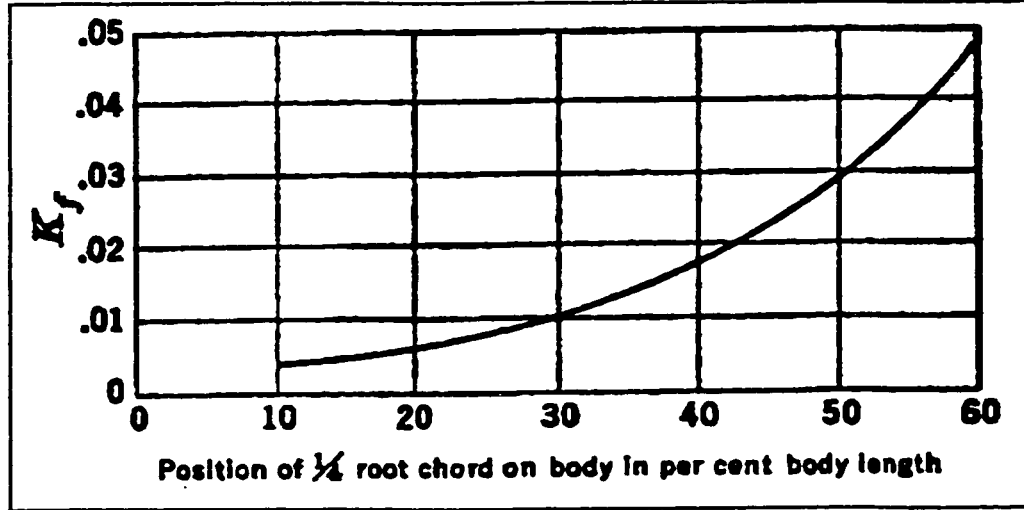


Figure 7.22 Fuselage pitch due to angle of attack factor. [7.4]

The horizontal stabilizer contribution depends on the tail volume.

$$C_{m_{ah}} = -C_{L_{ah}} \bar{V}_h \eta_h \left(1 - \frac{d\varepsilon}{d\alpha} \right) \quad (7.55)$$

7.3.3.3 Aircraft Pitch Moment Coefficient at Zero Angle of Attack.

Knowing the slope of the pitching moment coefficient, the zero lift pitching moment coefficient and the zero lift angle of attack, the zero angle of attack pitching moment coefficient is obtained using Equation 7.56.

$$C_{m_0} = C_{m_{0L}} + C_{m_\alpha} \alpha_{0L} \quad (7.56)$$

7.3.3.4 Aircraft Pitch Moment Coefficient Due to Rate of Change of Angle of Attack

It has been shown that as the angle of attack of an aircraft changes, there is a delay for the air affected by the wings to reach the tail. This is reflected in the rate coefficients. The pitching moment coefficient due to the rate of change of angle of attack is estimated by Equation 7.57

$$C_{m_\alpha} = -2C_{L_{\alpha_h}} \eta_h \bar{V}_h \frac{X_{ac_h} - X_{cg}}{\bar{c}} \left(\frac{d\varepsilon}{d\alpha} \right) \quad (7.57)$$

7.3.3.5 Aircraft Pitch Moment Coefficient Due to Pitch Rate

The pitch damping term is estimated by the sum of the effect due to the wings and the horizontal stabilizer according to Equation 7.58.

$$C_{m_q} = C_{m_{q_w}} + C_{m_{q_h}} \quad (7.58)$$

At low mach numbers, the effect due to the wings is estimated using Equation 7.59:

$$C_{m_{q_w}} = -K_w C_{L_{\alpha_w}} \cos \Lambda_{c/4} \left(\frac{A \left(2 \left(\frac{X_w}{\bar{c}} \right)^2 + 0.5 \left(\frac{X_w}{\bar{c}} \right) \right)}{A + 2 \cos \Lambda_{c/4}} + \frac{A^3 \tan^2 \Lambda_{c/4}}{24 (A + 6 \cos \Lambda_{c/4})} + \frac{1}{8} \right)$$

where K_w is given by Figure 7.23 based on the wing's aspect ratio.

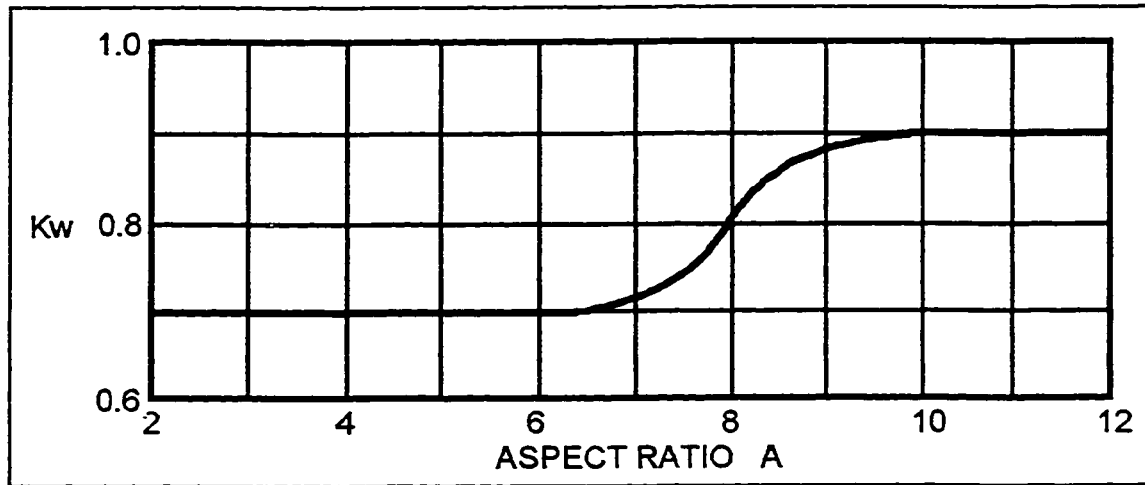


Figure 7.23 Wing correction factor for pitch damping. [2.5]

7.3.3.6 Aircraft Pitch Moment Coefficient Due to Elevator Deflection

The elevator is the control surface used to control the aircraft's pitch. Its effect is found in a manner similar to the lift due to elevator deflection. An equivalent angle of attack for the given elevator deflection is found using Equation 7.47 and then is used in Equation 7.60.

$$C_{m_{\delta_e}} = -\alpha_{\delta_e} C_{L_{\alpha_h}} \eta_h \bar{V}_h \quad (7.60)$$

8.0 CONCLUSION

8.1 General Review

The purpose of this project was to initiate the development of Concordia's flight simulator system into a versatile research and teaching simulation system. This objective was achieved by replacing the existing computer hardware with a modern computer and interfaces. This allowed for a flexible flight simulation software to be developed to exploit the academic potential of this system. The development of this software is documented, starting with the description of the development of the dynamic model and the implementation of this model into a working simulation program. A detailed description of the operation of the flight and navigational instruments was presented with a description of the method used to simulate them in the code. Finally to offer an insight into typical aircraft stability and control characteristics, a program was developed to implement a method to estimate the stability and control coefficients for a given aircraft configuration.

8.2 Overview and Discussion

8.2.1 The Dynamic Model

Chapter 2 presented the mathematical dynamic model used to simulate the motion of the aircraft. This included a full six degree of freedom flight model, the ground model, the forces and moments due to engine thrust and gravity, and the atmospheric model.

The flight model was developed primarily based on the work of Connelly [2.1] and completely describes the motion both translationally and

rotationally of an aircraft travelling through three dimensional space. The only restriction imposed on the model is that the aircraft should be symmetrical both geometrically and in its mass distribution about a vertical plane running along the longitudinal axis. Aircraft to which this assumption does not hold are very rare and are usually experimental designs. The three coordinate systems used; the body, stability, and fixed axis, were carefully described along with techniques to transform quantities from one system to another. Understanding of these systems is fundamental to understanding the flight dynamics since the forces acting on the aircraft are calculated in the stability axis, transformed to the body axis to apply the equations of motion, and then the motion is transformed to the fixed axis to calculate the aircraft motion and position in three dimensional space.

It was originally planned to implement a simplified ground dynamic model similar to those used on commercial flight simulators. It was soon discovered that the complexity of these models, even if simplified, made them impractical for a simulation which was to be easily reconfigurable. The understanding of the tire and road interaction required to reasonably tune such a model would be excessive for this application to the point where it would compromise the effectiveness of this program as an academic tool. For this reason a hybrid dynamic and kinematic ground model was developed. When the aircraft is on the ground the lateral equations, yaw, and side motion, would no longer be governed by the dynamic equations but would switch to a kinematic model depending only on the forward velocity and the geometric arrangement of the landing gear including the nose gear steering angle. The longitudinal and vertical equations of roll, pitch, vertical and longitudinal motion would still be governed by the dynamic equations allowing the aircraft to accelerate, takeoff, and land.

The application of engine thrust in this model was significantly simplified. While the effect of jet or prop wash over control surfaces can have a

significant effect on the performance of an aircraft, it was decided that due to the desired flexibility in defining the aircraft's stability and control characteristics the engines would be modelled as an application of two force vectors at a symmetrically definable point in the body axis. Only the resulting pitching and yawing moments and the forward thrust would be considered in the dynamic model.

The ambient atmospheric conditions are critical to an aircraft performance both from the point of the aerodynamic and the engine performance. To model the changes in atmospheric conditions the ICAO standard atmospheric model was implemented. Although the ICAO model describes the atmosphere to very high altitudes, it was implemented only to sixty five thousand feet. This is well above the maximum altitude of any commercial aircraft.

8.2.2 Computer Implementation of the Dynamic Model

Chapter three described the implementation of the dynamic model into a working flight simulation program. The equations are solved iteratively to yield a continuous real time solution. The overall flow of the program was described including core points of the simulation such as the solution of the dynamic equations to the position of the aircraft in the fixed coordinate system. Timing is critical in any real time simulation. Since different computers run at different speeds, slowing the simulation to accommodate the slowest machine would be unacceptable. Instead, the iteration rate of the simulation is constantly being self monitored and any variations are accounted for by varying the iteration period used in time dependent processes such as numerical integration to determine velocity and position. Unfortunately the available timing functions lacked the precision to monitor and update the iteration period every iteration. To overcome this, an average iteration rate is measured every two seconds and used to modify the iteration period as

needed. Updating the iteration timer every two seconds did not create any noticeable ill effects on the simulation. To avoid cumulative errors the time used to measure the iteration rate is always referenced to the time at the start of the simulation.

The user interface is essential in achieving the goals of this work. Depending on the type of computer being used, many options are available both to control and monitor the program. On the computer display the user may select to display a graphical representation of an EFIS display including navigational instruments. This display is very useful if it is desired to fly the simulation manually. Another display shows a map of the Montreal area including the local airports and navigational beacons with their frequencies. By default when the simulation starts, the aircraft is positioned on runway 07 at Cedars Airport south west of Montreal ready to take off and fly over Montreal on an heading of 70 degrees magnetic. Along the right side of this display is an isometric representation of the aircraft that will rotate with the simulation to indicate the aircraft's attitude. Below this are a list of constantly updated flight parameters. The position of the aircraft is represented by a yellow square on the map. While it is possible to fly the simulation from this screen, it is difficult without the flight instruments. This screen was intended to monitor the progress of a pilot flying from the full simulated cockpit or an autopilot in future work. Lastly a screen may be selected which can display up to any eight flight parameters in real time like a strip chart recorder with a definable time scale. This display proved useful especially in the class room where it was used to demonstrate the time response to an external disturbance.

To control the simulation a number of options are available. Firstly if connected, the controls may be read from the simulated cockpit. Failing this a number of standard PC inputs may be used such as a joystick, rudder pedals,

mouse, and keyboard. These standard input devices did have their limitations. The joystick and pedals while they may be physically representative of actual aircraft controls, the game port to which they connect was found to be slow and caused the iteration rate to decrease. The mouse is a much faster device than the joystick, but if used as a control column lacks the centre reference point and certainly lacks the feel of a control column. Although these drawbacks do effect the performance they are outweighed by the advantages of being able to run the simulation on many platforms in different locations. If the drawbacks are considered to be unacceptable then two other option were implemented. Firstly the simulation could be flown according to a set of predefined commands stored in a file. This would not only bypass the shortcomings of the input devices but would also permit exactly repeatable flight tests. Secondly if the simulation is slowed to the point that the fidelity is compromised then one may select to run the simulation at a fixed iteration period for the sake of numerical calculation but the simulation would still run at same iteration rate. The result of this would restore the fidelity but at the cost of no longer running in real time. This would not be a problem if the simulation is being controlled by a file and the results monitored or logged. In this way a slower computer could achieve the exact same results as the most powerful ones, only take a little longer to do so.

To be valuable as an academic tool, provisions must be included to store flight data to a file for later analysis. This was achieved by allowing the user to open a file and store flight data at a specified rate. It could be noted that this is possible only though the use of programming pointers supported by C. A simple data storage feature could not have been implemented in Fortran, one would have had to actually modify the source code.

8.2.3 Aircraft Instruments, Operation and Simulation

Part of a flight simulation code which is to be flyable from a pilot's standpoint must include a simulation of the aircraft flight and navigational instruments. A detailed description of the operation of the primary flight instruments was provided in chapter four. This included the errors inherent in the systems, the approach used to model them, and a description of the servo mechanisms used in the simulated instruments.

The navigation systems are more complicated. This thesis limited itself to systems commonly found on most instrument flight certified aircraft. They included the ADF, VOR, ILS, and DME. Two approaches were described to simulate the navigational systems. A cartesian approach which neglects the curvature of the earth and the polar approach which assumes a spherical earth. It was decided to implement the spherical approach since, although more complicated than the cartesian, it offered a complete solution which would remain valid in the more northern and southern hemispheres. The data used to define the location and frequency are stored in a simple editable file. As the navigation beacons are modified in this file, it would be reflected in the map display described in the previous section.

8.2.4 Simulation Hardware Selection

Without upgrading the computer hardware, this thesis would not have been possible. The choice to use a standard 80x86 personal computer was virtually self evident from the objectives. To be useful as an academic tool it must not be confined to a laboratory dependant on a large simulated cockpit. The PC family of computers is an inexpensive machine in widespread use in schools and homes.

Selecting this platform ensured that the use of this program could be portable and plentiful. The PC in its most common home configuration has sufficient input devices such as a mouse and joystick, to run this program. If not the purchase of the necessary hardware would certainly be less than a hundred dollars, making it a reasonable cost solution. With the motherboard chosen, the selection of the interface cards to drive the simulated cockpit involved an extensive search of the market. In the end, a commercial 64 channel high speed analog input board and a 216 line digital input output board were chosen. These cards by far exceeded the minimum requirements to drive the simulated cockpit but offered extensive possibilities for future applications with a modest premium over what the minimums would have cost. It was found that commercially available analog output cards lacked the number of channels and current capabilities to drive the simulated instruments. As a result a custom analog output board was developed with 32 channels. The board was built specifically to the simulator's needs. Sixteen pairs of eight bit digital to analog converters are mounted on an sixteen bit data bus. This allowed a simulated instrument to be driven by a single command, providing the two voltages required to drive it. It was found that the sacrifice in resolution due to the eight rather than twelve bit DACs did not noticeably affect the performance of the instruments.

8.2.5 Aircraft Stability and Control Estimation

Since the simulation program offers the ability to modify the stability and control coefficients, an additional package was developed to estimate the longitudinal stability and control coefficients of an aircraft. Based largely on the US Air Force DATCOM [2.5] as presented by Roskam [7.1], the program implements the Roskam method, simplified for a subsonic aircraft with limited geometry. The program allows the student to modify the aircraft configuration within the limitations

of the method. The result is a step by step presentation of all the calculations and graphs required to estimate the coefficients with the actual parameters entered by the student. By varying the geometry of the aircraft the student can gain an insight in to the parameters which affect the aircraft stability.

8.3 Future Work

This thesis has started the next level of development of Concordia's flight simulator. The simulation program has been rewritten to include a reconfigurable flight model and feature to enhance its use as a academic tool. There still remains a number of aspects to be addressed as future work. Much of this could be the topic of student course projects.

8.3.1 Sound Simulation

The audio cues to a pilot are a key feedback, engine, radio, and other sounds enhance situation awareness. In anticipation of this, a commercially available "Sound Blaster" sound card has been installed in the computer. Through the many commercially available programs on the market which make extensive use of this card, there is no question that it is capable of generating realistic sounds for the simulation. Extensive use will be made of free programming assistance available from the manufacturer's , Creative Labs, internet web site.

8.3.2 Navigational Instruments

While the code has been written to simulate the navigational systems,

the hardware and software required to read and drive the actual cockpit navigational instruments is still to be implemented. These instruments are for the most part actual aircraft instruments. Custom interface boards will be built to read and drive the three phase synchros and resolvers used in those systems. The necessary electronic hardware exists from the original system and will be interfaced to the same external card rack as the analog output board.

8.3.3 Propulsion Modelling

At present the engines are simulated as a simple variable thrust vector. Future work will see the development of realistic engine models including propeller performance, and the governor system used to control the blade pitch angle to maintain a constant engine speed at different power settings. A lower priority will be the development of a turbine engine model.

8.3.4 Control Loading

Another important feedback to the pilot is the force which he feels in the controls. During the majority of a flight pilots generally fly according to the feel in the controls rather than their actual position. A pneumatic system is already in place which can provide forces on the controls by varying the pressure in a pneumatic cylinder with a dc servo regulator valve. The forces which the pilot feels would be a function of the position of the controls, the aerodynamic forces acting on the control surfaces to which they are linked and the trim setting. Simulation software will be needed to model these effects and drive the regulator to provide the pilot with a realistic feel.

8.4 Concluding Remarks

Through this thesis the evolution of Concordia's flight simulator to a powerful academic tool has begun. While there still remains work to be done to complete this task, it has already had some academic applications applied to it in a teaching environment. The insight and understanding which it has given students is undeniable and hopefully just a hint of what it will finally be capable of, once complete.

REFERENCES

- [1.1] F.J. Hooven, "The Wright brother's flight-control system", Scientific American, 239, 1978.
- [1.2] J.M. Rolfe and K.J. Staples, Flight Simulation, Cambridge University Press, New York, 1986.
- [1.3] US Air Force Museum, Wright-Patterson Air Force Base, Dayton,
<http://www.wpafb.af.mil/museum/index.htm>
- [1.4] United Airlines, United Airlines Flight Training Centre.
<http://www.ualfttctr.com>
- [1.5] "The Unforgettable Winston Churchill", Time-Life News Book, New York, 1965.
- [1.6] C. Orlebar, The Concord Story, Temple Press, London, 1986.
- [1.7] MAXVIEW visual scene, CAE Electronics, Montreal 1998.
- [1.8] Photograph C-76429-16, Canadair, Montreal, 1996.
- [1.9] J. Svoboda, S. Lasnier, G.M. McKinnon, "Microprocessor-Based Light Aircraft Simulator", Canadian Industrial Computer Society, Ottawa, May 1986.
- [2.1] Mark E. Connelly, "Simulation of Aircraft", M.I.T. Report, Massachusetts Institute of Technology, 1958.

- [2.2] R. C. Nelson, Flight Stability and Automatic Control, McGraw Hill Inc., New York, 1989.
- [2.3] J. Roskam, Airplane Flight Dynamics and Automatic Flight Controls, Roskam Aviation and Engineering Corporation, Ottawa, 1994.
- [2.4] Sport Aviation, Experimental Aircraft Association, October 1996.
- [2.5] R.D. Finck, D.E. Ellison, L.V. Malthan, USAF Stability and Control Datcom, Flight Control Division, Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, 1978.
- [2.6] Boeing, Aircraft on Ground Performance, Boeing Report D6-44062.
- [2.7] J. Roskam, Airplane Aerodynamics and Performance, Roskam Aviation and Engineering Corporation, Ottawa, 1988, pp.
- [3.1] Borland, Borland C++ Version 3.1 Library Reference, Borland International, Scotts Valley, 1992.
- [3.2] Microsoft, Microsoft MS-DOS User's Guide and Reference, Microsoft Corporation, Redmond, 1991.
- [3.3] T. Dettmann, DOS Programmer's Reference, 4th ed., QUE Corporation, Carmel, 1993.
- [3.4] L.C. Eggbrecht, Interfacing to the IBM Personal Computer, 2nd ed., Macmillan Computer Publishing, Carmel, 1990.
- [3.5] Honeywell, <http://www.cas.honeywell.com/ats/products/aims/imapics.html>

- [4.1] From the Ground Up, 25th ed., Aviation Publishers, Ottawa, 1987

- [4.2] Transport Canada, Canadian Air Regulations. Section 605.16,
<http://www.tc.gc.ca/aviation/regserv/carac/CARS/toccarse.htm#302>

- [4.3] E. Pallett, Aircraft Instruments, 2nd ed., Pitman Publishing, Marshfield, 1982.

- [4.4] Unclassified document, Standard Magnetic Variation, Appendix 6 to STANAG 4294 1st ed., North Atlantic Treaty Organization, 1993.

- [4.5] C.E. Barton, Revision of International Reference Field Released,
http://www.agu.org/eos_elec/95206e.html. ©1996 American Geophysical Union.

- [4.6] International Geomagnetic Reference Field, NASA / Goddard Space Flight Center, <http://fdd.gsfc.nasa.gov/IGRF.html>

- [4.7] Pilot's Operating Handbook and FAA Approved Airplane Flight Manual, for the Beechcraft Duchess 76, Beech Aircraft Corporation, Wichita, 1978.

- [4.8] Allied Signal Commercial Avionics Systems,
http://www.aliedsignal.com/aerospace/product/avionics_cat/index.html

- [4.9] G.E. Irving, Aircraft Instruments, 1st ed., McGraw Hill, New York, 1941.

- [4.10] P. Garrison, Autopilots, Flight Directors and Flight Control Systems, 1st ed., Tab, Blue Ridge Summit, 1985.

- [5.1] M.J. Culhane, Commercial Pilot Ground School Course, Accelerated Aviation Training, Vancouver, 1997
- [5.2] Avionics Fundamentals, International Standard Book Number 0-89100-293-6.
- [5.3] Airborne ADF System Mark3, Arinc Characteristic 570, Aeronautical Radio Inc., 1968.
- [5.4] Transport Canada, Flight Training Manual, 3rd ed. Gage Educational Publishing Company, Toronto, 1989.
- [5.5] Canada Flight Supplement, Natural Resources Canada, July 17, 1997
- [5.6] Transport Canada, Flight Training Manual, 4th ed. Gage Educational Publishing Company, Toronto, 1994.
- [5.7] Airborne VOR Receiver, Arinc Characteristic 579-2, Aeronautical Radion Inc., 1989.
- [5.8] Precision Airborne Distance Measuring Equipment, Arinc Characteristic 709A, Aeronautical Radio Inc., 1994.
- [5.9] Airborne ILS Receiver, Arinc Characteristic 578-4, Aeronautical Radio Inc., 1983.
- [6.1] DC Motors Speed Controls Servo Systems, Robbins & Myers / Electro Craft, Eden Praire.

- [6.2] E.K. Pritchard, "Analog Operation of Stepping Motors", Proceedings of 6th Annual Incremental Motion Control Systems Devices Symposium, University of Illinois, Urbana-Champaign, 1997, pp 283-294.
- [6.3] 1994 Cataloge, National Instruments, AT-MIO-64E-5 Specifications.
- [6.4] Model PCPIO Series Product Manual, Industrial Computer Source, San Diego, 1995.
- [6.5] L.C. Eggbrecht, Interfacing to the IBM Personal Computer, 2nd ed., Macmillan Computer Publishing, Carmel, 1990.
- [7.1] J. Roskam, Aircraft Design Part VI, Roskam Aviation and Engineering Corporation, Ottawa, 1994.
- [7.2] J.D. Anderson, Fundamentals of Aerodynamics, 2nd ed., McGraw Hill, New York, 1991
- [7.3] Perkins and Hage, Airplane Performance. Stability and Control, Wiley, York New, 1949.
- [7.4] R.R. Gilruth, "Analysis and Prediction of Longitudinal Stability of Airplanes", NACA TR 711.

Background References

B.W. McCormick, Aerodynamics, Aeronautics and Flight Mechanics, 2nd ed., John Wiley and Sons, New York, 1995.

R.S. Shevell, Fundamentals of Flight, 2nd ed., Prentice Hall, Englewood Cliffs, 1989.

M. Kayton and W.R. Fried, Avionics Navigation Systems, John Wiley and Sons, New York, 1969.

B. Etkin and L.D. Ried, Dynamics of Flight Stability and Control, 3rd ed., John Wiley and Sons, New York, 1996.

APPENDIX 1

SIMULATION GLOBAL VARIABLES

```

/*****
/*
/*      Configurable Flight Simulator
/*      Global Variables
/*
/*      Written by Peter Lawn      Fall 1995
/*      Fluid Power and Simulation Laboratory
/*      Department of Mechanical Engineering
/*      Concordia University
/*      1455 DeMaisonneuve boul West
/*      Montreal, Quebec
/*      CANADA
/*      tel (514)-848-3131
/*      fax (514)-848-3175
/*
/*      This file GLOB_VAR.C
/*
/*      This file contains all global variables.
/*      Unless otherwise stated, the units for all variables are :
/*      meter
/*      radian
/*      kilogram
/*      Newton
/*      second
/*
*****/

```

```

/*=====
/*
/*      AIRCRAFT CONFIGURATION
/*
/*=====

```

```

/*-----*/
/*      REFERENCE SPEED
/*-----*/

```

```

extern
float Ref_Vel;          /* Reference velocity          (m/s) */

```

```

/*-----*/
/*      LIFT
/*-----*/

```

```

extern
float Lift ,           /* Net lift in body axis system      (N) */
      C_L_Tot ,        /* Total aerodynamic lift coefficient */
      C_L ,            /* Base coefficient of lift
      C_L_alpha ,      /* Coef lift func(angle of attack)
      C_L_alphaDOT ,   /* Coef lift func(rate of ang of attack)
      C_L_q ,          /* Coef lift func(pitch rate)
      C_L_delta_e ;    /* Coef lift func(elevator angle)

```

```

/*-----*/
/*      DRAG
/*-----*/

```

```

extern
float Drag ,           /* Net drag in body axis system      (N) */
      C_D_Tot ,        /* Total aerodynamic drag coefficient */
      C_D ,            /* Base coefficient of drag
      C_D_alpha ,      /* Coef drag func(angle of attack)
      C_D_delta_e ;    /* Coef drag func(elevator angle)

```

```

/*-----*/
/*      SIDE FORCE
/*-----*/

```

```

extern
float C_y_Tot ,        /* Total aerodynamic side force coef

```



```

C_y      ,      /* Base coefficient of side force      */
C_y_beta ,      /* Coef SF func(side slip angle)      */
C_y_p    ,      /* Coef SF func(roll rate)            */
C_y_r    ,      /* Coef SF func(yaw rate)             */
C_y_delta_a , /* Coef SF func(aileron angle)       */
C_y_delta_r ;   /* Coef SF func(rudder angle)        */

/*-----*/
/*      PITCHING MOMENT      */
/*-----*/

extern
float C_m_Tot ,      /* Total aerodynamic pitch moment coef */
      C_m      ,      /* Base pitching moment coefficient    */
      C_m_alpha ,      /* Coef PM func(angle of attack)       */
      C_m_alphaDOT, /* Coef PM func(rate of ang of attack) */
      C_m_q     ,      /* Coef PM func(pitch rate)           */
      C_m_delta_e ;   /* Coef PM func(elevator angle)       */

/*-----*/
/*      ROLLING MOMENT      */
/*-----*/

extern
float C_l_Tot ,      /* Total aerodynamic roll moment coef */
      C_l      ,      /* Base rolling moment coefficient     */
      C_l_beta ,      /* Coef RM func(side slip angle)       */
      C_l_p    ,      /* Coef RM func(roll rate)            */
      C_l_r    ,      /* Coef RM func(yaw rate)             */
      C_l_delta_a, /* Coef RM func(aileron angle)       */
      C_l_delta_r;   /* Coef RM func(rudder angle)        */

/*-----*/
/*      YAWING MOMENT      */
/*-----*/

extern
float C_n_Tot ,      /* Total aerodynamic yawing moment coef */
      C_n      ,      /* Base yawing mement coefficient      */
      C_n_beta ,      /* Coef YM func(sideslip angle)       */
      C_n_p    ,      /* Coef YM func(roll rate)            */
      C_n_r    ,      /* Coef YM func(yaw rate)             */
      C_n_delta_a, /* Coef YM func(aileron angle)       */
      C_n_delta_r;   /* Coef YM func(rudder angle)        */

/*-----*/
/*      MASS      */
/*-----*/

extern
float AC_Mass ,      /* Aircraft weight (kg) */
      AC_Ix   ,      /* Rolling moment of inertia (kg m^2) */
      AC_Iy   ,      /* Pitching moment of inertia (kg m^2) */
      AC_Iz   ,      /* Yawing moment of inertia (kg m^2) */
      AC_Ixz  ;      /* Product of inertia about xz (kg m^2) */

/*-----*/
/*      DIMENTIONS      */
/*-----*/

extern
float AC_S ,      /* Wing platform area (m^2) */
      AC_b ,      /* Wing span (m) */
      AC_c ;      /* Mean chord (m) */

/*-----*/
/*      CONTROL SURFACE DEFLECTIONS      */
/*      Set the value in deg, rad is      */
/*      calculated from deg      */
/*-----*/

extern
float Delta_a_deg ,      /* Maximum airleron deflection (+/-deg) */

```

```

Delta_e_deg ,      /* Maximum elevator deflection (+/-deg) */
Delta_r_deg ,      /* Maximum rudder deflection (+/-deg) */
Delta_f_deg ,      /* Maximum flap deflection (+/-deg) */
Delta_w_deg ,      /* Maximum gear steering angle (+/-deg) */

Delta_a_rad,       /* Aileron deflection (+/-rad) */
Delta_e_rad,       /* Elevator deflection (+/-rad) */
Delta_r_rad,       /* Rudder deflection (+/-rad) */
Delta_f_rad,       /* Flap deflection (+/-rad) */
LG_Steer_Ang;      /* Nose gear steering angle (+/-rad) */

/*-----*/
/* ENGINE CONFIGURATION */
/*-----*/

extern
float Eng_T_X ,      /* Engine thrust X point of application */
      Eng_T_Y ,      /* Engine thrust Y point of application */
      Eng_T_Z ;      /* Engine thrust Z point of application */

/*-----*/
/* FORCES AND MOMENTS AND ACCELERATIONS */
/*-----*/

/*-----*/
/* MOMENTS IN BODY AXIS SYSTEM */
/*-----*/

extern
float M_roll_BA ,    /* Rolling moment (Nm) */
      M_pitch_BA ,   /* Pitching moment (Nm) */
      M_yaw_BA ;     /* Yawing moment (Nm) */

/*-----*/
/* FORCES IN BODY AXIS SYSTEM */
/*-----*/

extern
float Fx_BA ,        /* Longitudinal force (N) */
      Fy_BA ,        /* Lateral force (N) */
      Fz_BA ;        /* Vertical force (N) */

/*-----*/
/* ACCEL IN BODY AXIS SYSTEM */
/*-----*/

extern
float Ax_BA,         /* X-acceleration in body axis (m/s^2) */
      Ay_BA,         /* Y-acceleration in body axis (m/s^2) */
      Az_BA;         /* Z-acceleration in body axis (m/s^2) */

/*-----*/
/* ACCEL IN BODY FIXED SYSTEM */
/*-----*/

extern
float Ax_FA,         /* X-acceleration in body axis (m/s^2) */
      Ay_FA,         /* Y-acceleration in body axis (m/s^2) */
      Az_FA;         /* Z-acceleration in body axis (m/s^2) */

/*-----*/
/* ACCEL IN MAGNETIC COORDINATES */
/*-----*/

extern
float Mag_E_Accel,    /* Accel in mag east direction (m/s^2) */
      Mag_N_Accel;    /* Accel in mag nrth direction (m/s^2) */

/*-----*/

```

```

/*
/*      VELOCITIES, POSITION, AND ORIENTATION
/*
/*-----*/

/*-----*/
/*      VELOCITY IN BODY AXIS SYSTEM
/*-----*/

extern
float Vx_BA ,      /* Forward velocity          (m/s) */
      Vy_BA ,      /* Lateral velocity          (m/s) */
      Vz_BA ,      /* Vertical velocity         (m/s) */
      V_roll_BA ,  /* Roll rate                 (rad/s) */
      V_pitch_BA , /* Pitch rate                (rad/s) */
      V_yaw_BA ,   /* Yaw rate                  (rad/s) */
      AlphaDOT ;   /* Rate of change of angle of attack */

/*-----*/
/*      VELOCITY IN FIXED AXIS SYSTEM
/*-----*/

extern
float Vx_FA ,      /* Forward velocity (N/S)      (m/s) */
      Vy_FA ,      /* Lateral velocity (E/W)      (m/s) */
      Vz_FA ,      /* Vertical velocity          (m/s) */
      V_roll_FA ,  /* Roll rate                  (rad/s) */
      V_pitch_FA , /* Pitch rate                 (rad/s) */
      V_yaw_FA ;   /* Yaw rate                   (rad/s) */

/*-----*/
/*      VELOCITY IN STAB AXIS SYSTEM
/*-----*/

extern
float Vx_SA ;      /* Forward velocity          (m/s) */

/*-----*/
/*      POSITION IN FIXED AXIS SYSTEM
/*-----*/

extern
float Pos_x_FA ,   /* X position (N/S)          (m) */
      Pos_y_FA ,   /* Y position (E/W)          (m) */
      Pos_z_FA ;   /* Z position (true altitude) (m) */

extern
double Pos_Lat_Dec , /* Latitude                (deg) */
      Pos_Lng_Dec ; /* Longitude                (deg) */

extern
int Pos_Lat_Deg ,  /* Latitude seconds component */
    Pos_Lat_Min , /* Latitude minutes component */
    Pos_Lng_Deg , /* Longitude seconds component */
    Pos_Lng_Min , /* Longitude minutes component */
    Pos_Lat_Sec , /* Latitude seconds component */
    Pos_Lng_Sec ; /* Longitude seconds component */

/*-----*/
/*      ORIENTATION IN WIND AXIS SYSTEM
/*-----*/

extern
float Alpha ,      /* Angle of attack          (rad) */
      Beta ;      /* Side slip angle          (rad) */

/*-----*/
/*      ORIENTATION IN FIXED AXIS SYSTEM
/*-----*/

extern
float Pitch_FA ,   /* Pitch angle              (rad) */

```

```

Roll_FA      ,      /* Roll angle                      (rad) */
Yaw_FA       ,      /* Yaw                      (rad) */
Head_Mag     ;      /* Magnetic Headding       (deg) */

/*=====*/
/*
/*      EARTH's MAGNETIC FIELD
/*
/*=====*/

extern
float B_Lat   ,      /* Lateral field potential      */
      B_Lon   ,      /* Longitudinal field potential */
      B_Rad   ,      /* Radial                      */
      Mag_Var ,      /* Local variation             (deg) */
      Mag_Dip ;      /* Local dip                   (deg) */

/*=====*/
/*
/*      AMBIENT AIR CONDITIONS
/*
/*=====*/

extern
float Amb_Air_D ,      /* Ambient air density          (kg/m^3) */
      Amb_Air_T ,      /* Ambient air temperature      (deg C) */
      Amb_Air_P ,      /* Ambiet ait press            (kPa) */
      gamma     ,      /*
      Gas_Const_R,      /* Gas constant                 (kJ/kg K) */
      Fixed_Altitude, /* Altitude for constant atmo   (m) */
      Alt_Setting, /* Altimter Setting             (in Hg) */
      To_z0     ,      /* Temperature at 0 altitude    (K) */
      Po_z0     ,      /* Pressure at 0 altitude       (kPa) */
      ro_z0     ;      /* Density at 0 altitude        (kg/m^3) */

extern
short int Fixed_Alt;    /* Flag to use constant atmosphere */

/*=====*/
/*
/*      PROPELLER
/*
/*=====*/

extern
float Prop_Dia   ,      /* Propeller diameter          (m) */
      Prop_Pitch ;      /* Propeller pitch β           (deg) */

extern
int Prop_No_Bld ;      /* Number of blades/propeller */

/*=====*/
/*
/*      ENGINE PERFORMANCE
/*
/*=====*/

extern
float Eng_Thrust[2];    /* Thrust                      (N) */

extern
int Eng_Type   ;      /* Engine Type 0=Pure thrust    */
                        /*      1=Piston Prop          */
                        /*      2=Turbo Prop           */
                        /*      3=Jet                  */
/*-----*/
/* Pure Thrust
/*-----*/
extern
float PTEng_Max ;      /* Maximum pure thrust at full thro (N) */
/*-----*/

```

```

/* Piston Prop */
/*-----*/
extern
float PPEng_Throt_D , /* Throttle Diameter (m) */
Eng_RPM[2] , /* Engine RPM */
PPEng_MAP[2] , /* Engine Manifold Press (kPa) */
PPEng_Displ , /* Engine displacement (l) */
PPEng_AF , /* Air fuel ratio */
PPEng_CR , /* Compression ratio */
PPEng_Fuel_HV , /* Fuel energy content (kJ/kg) */
PPEng_Inertia ; /* Engine inertia (kg m^2) */

/*=====*/
/*
/* CONTROL INPUTS
/*
/*
/*=====*/

extern
float Ctrl_e , /* Elevator control input (-1 to +1) */
Ctrl_a , /* Aileron control input (-1 to +1) */
Ctrl_r , /* Rudder control input (-1 to +1) */
Ctrl_f , /* Flap control input (0 to +1) */
Throttle[2] , /* Throttle positions (0 to +1) */
ConstSpd[2] , /* Prop Speed (0 to +1) */
Mixture[2] , /* A/F mixture (0 to +1) */
Ctrl_alt ; /* Altimeter setting (0 to +1) */

/*=====*/
/*
/* CIRCUIT BREAKERS 1 = Closed / 0 = Open
/*
/*
/*=====*/

extern
short int CB[4][9]; /* Circuit breakers */

/*=====*/
/*
/* SWITCH INPUTS (0=Off 1=On)
/*
/*
/*=====*/

extern
short int SW_Battery , /* Master battery switch */
SW_L_Alt , /* Left alternator */
SW_R_Alt , /* Right alternator */
SW_L_Prime , /* Left engine prime */
SW_L_LMag , /* Left engine left magneto */
SW_L_RMag , /* Left engine right magneto */
SW_L_Start , /* Left engine start */
SW_R_Prime , /* Right engine prime */
SW_R_LMag , /* Right engine left magneto */
SW_R_RMag , /* Right engine right magneto */
SW_R_Start , /* Right engine start */
SW_Land_L , /* Landing lights */
SW_Taxi_L , /* Taxi lights */
SW_Nav_L , /* Navigation lights */
SW_Strobe_L , /* Strobe lights */
SW_Cabin_Blow , /* Cabine air blower */
SW_Pilot_Heat , /* Pilot heater */
SW_Auto_Pilot , /* Auto pilot On/Off switch */
SW_Elec_Trim , /* Electric trim */
SW_Heater , /* Heater */
SW_L_Aux_Pump , /* Left auxiliary fuel pump */
SW_R_Aux_Pump , /* Right auxiliary fuel pump */
SW_Gear_Down , /* Landing gear down (1=up) */
SW_Prk_Brake , /* Parking brake */
SW_Flaps_Up , /* Raise flaps */
SW_Flaps_Down ; /* Lower flaps */

/*=====*/

```

```

/*                                                    */
/*      INSTRUMENT DISPLAY VALUES                    */
/*                                                    */
/*=====*/

extern
float Inst_Mag_Comp,          /* MAGnetic Compass          (deg) */
    Ang_ASI,                 /* Air speed indicator angle  (deg) */
    Ang_ALT1,                /* Altimeter 100 angle        (deg) */
    Ang_ALT10,               /* Altimeter 100 angle        (deg) */
    Ang_ALT100,              /* Altimeter 100 angle        (deg) */
    Ang_TC,                  /* Turn coordinator angle     (deg) */
    Ang_SI,                  /* Slip indicator angle        (deg) */
    Ang_VSI,                 /* Vert speed ind angle        (deg) */
    Ang_Roll,                /* ATT IND roll angle          (deg) */
    Ang_Ptch,                /* ATT IND ptch angle          (deg) */
    Inst_Roll_tgt,           /* Roll target angle           (deg) */
    Inst_Ptch_tgt,           /* Pitch target angle          (deg) */
    Inst_Roll_vel,           /* Roll step size              (deg) */
    Inst_Ptch_vel;           /* Pitch step size             (deg) */

/*=====*/
/*                                                    */
/*      ELECTRICAL SYSTEMS                          */
/*                                                    */
/*=====*/

extern
float AI_Volts[30],          /* Measured analog input      */
    AI_Min[30],              /* Lower voltage limit         */
    AI_Max[30],              /* Upper voltage limit         */
    AI_Value[30],            /* Scale analog input          */
    AI_LowLim[30];           /* Scale lower limit (usually 0.0) */

/*=====*/
/*                                                    */
/*      ELECTRICAL SYSTEMS                          */
/*                                                    */
/*=====*/

extern
short int Electric_ON;       /* Set to 1 if power available */

/*=====*/
/*                                                    */
/*      AUTO PILOT                                  */
/*                                                    */
/*=====*/

extern
float AP_Heading ,           /* Auto pilot heading          (deg) */
    AP_Altitude ;           /* Auto pilot altitude         (m) */

/*=====*/
/*                                                    */
/*      LANDING GEAR                                */
/*                                                    */
/*=====*/

extern
float LG_Nose_Pos ,          /* Landing gear position 0 = retracted */
    LG_Left_Pos ,           /*                               1 = extended */
    LG_Right_Pos,           /*                               0.? = transit */

    LG_Nose_Time ,          /* Time for nose gear to extend (sec) */
    LG_Left_Time ,          /* Time for left gear to extend (sec) */
    LG_Right_Time,          /* Time for right gear to extend (sec) */

    LG_Length ,             /* Rear Z-length from gnd to CG (m) */
    LG_X_Nose_BA,            /* Nose gear X position from CG (m) */
    LG_X_Main_BA,            /* Main gear X position from CG (m) */
    LG_Y_Main_BA,            /* Main gear ± Y position from CG (+m) */
    LG_Ks_Nose ,            /* Nose gear spring constant (N/m) */

```

```

        LG_Kd_Nose ,          /* Nose gear damping constant      (Ns/m) */
        LG_Ks_Main ,          /* Main gear spring constant       (N/m) */
        LG_Kd_Main ,          /* Main gear damping constant      (Ns/m) */
        LG_Steer_Ang,         /* Nose gear steering angle        (rad) */

        LG_F_Z_Nose,          /* Nose gear Z force                (N) */
        LG_F_Z_Left,          /* Left gear Z force                (N) */
        LG_F_Z_Right,         /* Right gear Z force               (N) */
        LG_F_X_Nose,          /* Nose gear X force in body axis   (N) */
        LG_F_Y_Nose,          /* Nose gear Y force in body axis   (N) */
        LG_F_X_Left,          /* Left gear X force in body axis   (N) */
        LG_F_Y_Left,          /* Left gear Y force in body axis   (N) */
        LG_F_X_Right,         /* Right gear X force in body axis   (N) */
        LG_F_Y_Right,         /* Right gear Y force in body axis   (N) */
        LG_Fx_BA,              /* Net X langing gear force in BA    (N) */
        LG_Fy_BA,              /* Net Y langing gear force in BA    (N) */
        LG_Fz_BA,              /* Net Z langing gear force in BA    (N) */
        LG_M_roll_BA,          /* Net landing gear rolling moment   (Nm) */
        LG_M_pitch_BA,         /* Net landing gear pitch moment     (Nm) */
        LG_M_yaw_BA,           /* Net landing gear yawing moment    (Nm) */
        LG_Nose_Max_Force,     /* Maximum crash nose force          (N) */
        LG_Main_Max_Force;     /* Maximum crash nose force          (N) */

extern
short int LG_AOG;              /* Aircraft on ground flag           */

/*=====*/
/*
/*      CRASH ROUTINE
/*
/*=====*/

extern
short int Crash_On;            /* Crash check on/off                */

/*=====*/
/*
/*      FLAPS
/*
/*=====*/

extern
float Flap_Time ;              /* Time for full flap extention      (sec) */

/*=====*/
/*
/*      MAP DISPLAY VARIABLES
/*
/*=====*/

extern
float  MAP_Lat ,               /* Lattitude of map upper right corner */
      MAP_Lng ,               /* Longitude of map upper right corner */
      MAP_Range ,              /* Range of map                        (Nm) */
      MAP_Lat_Sc ,             /* Lattitude graphic scale (pixel/deg) */
      MAP_Lng_Sc ;             /* Longitude graphic scale (pixel/deg) */

/*=====*/
/*
/*      VOR
/*
/*=====*/

extern
int  VOR_Count,                /* VOR database index                */
    VOR_Freq[25],              /* VOR Frequency                     (MHzx100) */
    VOR_Frequency,             /* Seleced VOR frequency              (MHzx100) */
    VOR_OBS,                   /* Selected bearing                    (deg) */
    VOR_ON ,
    VOR_Flag;

extern
char VOR_ID[5][5],             /* VOR 3 letter id                   */

```

```

        VOR_Name[5][20];          /* VOR Name                                */

extern
float VOR_Lat[25],                /* VOR latitude                      (dec deg) */
      VOR_Lng[25],                /* VOR longitude                     (dec deg) */
      VOR_Var[25];                /* VOR variation                     (dec deg) */

/*=====*/
/*
/*      VORTAC                                */
/*
/*=====*/

extern
int    VORTAC_Count,              /* VORTAC database index                                */
      VORTAC_Freq[25],           /* VORTAC Frequency                      (MHzx100) */
      VORTAC_Chان[25];           /* VORTAC Channel                          */

extern
char   VORTAC_ID[25][5],         /* VORTAC 3 letter id                                */
      VORTAC_Name[25][20];       /* VORTAC Name                                    */

extern
float  VORTAC_Lat[25],           /* VORTAC latitude                      (dec deg) */
      VORTAC_Lng[25],           /* VORTAC longitude                     (dec deg) */
      VORTAC_Var[25];           /* VORTAC variation                     (dec deg) */

/*=====*/
/*
/*      DME                                */
/*
/*=====*/

extern
int    DME_Count,                /* DME database index                                */
      DME_Freq[25],              /* DME Channel                      */
      DME_Frequency,             /* Selected DME frequency          (MHzx100) */
      DME_ON;

extern
float  DME_Lat[25],              /* DME latitude                      (dec deg) */
      DME_Lng[25];              /* DME longitude                     (dec deg) */

/*=====*/
/*
/*      NDB                                */
/*
/*=====*/

extern
int    NDB_Count,                /* NDB database index                                */
      NDB_Freq[25],              /* NDB Frequency                      (KHz) */
      NDB_Frequency,            /* Selected NDB frequency          (KHz) */
      NDB_ON,
      NDB_Flag;

extern
char   NDB_ID[25][5],           /* NDB 3 letter id                                */
      NDB_Name[25][20];         /* NDB Name                                    */

extern
float  NDB_Lat[25],             /* NDB latitude                      (dec deg) */
      NDB_Lng[25];             /* NDB longitude                     (dec deg) */

/*=====*/
/*
/*      AIRCRAFT ON MAP                                */
/*
/*=====*/

extern
void *Shadow;

extern

```



```

    unsigned int Shadow_Size;

/*=====*/
/*
/*      STRIP RECORDER VARIABLES
/*
/*=====*/

extern
char    SR_Title[8][50];      /* Title of curve          */

extern
float    SR_Max[8],           /* Maximum value on chart  */
         SR_Min[8],           /* Minimum value on chart  */
         SR_Factor[8],        /* Scaling factor (eg m -> ft) */
         SR_Time_Span;        /* Chart time span         */

extern
double   SR_Time_E,           /* Elapsed time            */
         SR_Time_S;           /* Start time              */

extern
short int SR_Chart[8],        /* Chart identifier        */
         SR_ID[8];           /* Graph identifier        */

extern
float    *SR_Line[8];         /* Pointer to line variable */

/*=====*/
/*
/*      LOG FILE VARIABLES
/*
/*=====*/

extern
float    LF_Log_Period,        /* Log sampling period rate (s) */
         LF_Factor[17],       /* A scaling factor            */
         LF_Time,              /* Log file running time      (s) */
         LF_Next_Time;         /* Next time to record data (s) */

extern
int      LF_ID[17],           /* Column ID                  */
         LF_Tot_Col;          /* Number of columns-1        */

extern
float    *LF_Column[17];      /* Pointer to column variable */

extern
short int LF_Open;            /* True if the log file is open */

extern
char     LF_FileName[128];     /* File Name                  */

/*=====*/
/*
/*      FILE CONTROL VARIABLES
/*
/*=====*/

extern
double   FC_Time,             /* Elapsed time for file control sec */
         FC_Comm_Time;        /* Time to execute command          */

extern
int      FC_Comm_Count;       /* Command counter                  */

extern
char     FC_FileName[128];     /* File name                        */

/*=====*/
/*
/*      PROGRAM VARIABLES
/*
/*=====*/

```

```

extern
short int Prog_Quit ,          /* Flag to quit program          */
Visual_On ,                   /* 1=Visual connected 0=No visual */
Ctrl_Dev ,                    /* 1=Cockpit                      */
                                /* 2=Desktop controller          */
                                /* 3=Mouse, JoyStk, & Keyboard    */
                                /* 4=Mouse, JoyStk, Keyboard, & Pedals */
Disp_Dev ,                    /* 1 = Flight instruments        */
                                /* 2 = Graphs                    */
                                /* 3 = Instructor station        */
                                /* 4 = Area map                  */
                                /* 5 = None                      */
Use_Box ,                     /* Basic 6 instruments on parallel port */

Cycle_Quit,                   /* Flag to end simulation cycle    */
First_Cyc ,                   /* Set true on the first prog itt  */
Cycle_Pause,                  /* Set true to pause program       */
Real_Time ,                   /* Set to 1 for real time simulation */
Eng_Units ;                   /* Set to 1 for Eng 0 for metric   */

extern
float Cycle_Time ,            /* Cycle time                      (s) */
Fixed_Time ;                 /* Fixed time increment            (ms) */

/*=====*/
/*
/*      PILOT INPUT CONTROLS
/*
/*=====*/

extern
int      JoyStk_CX ,          /* Joy stick center X coordinate    */
JoyStk_CY ,                  /* Joy stick center Y coordinate    */
Rudder_C ;                   /* Rudder pedals center            */

/*=====*/
/*
/*      FAILURES
/*
/*=====*/

extern
short int Fail_Box_Color[5][7],
Prob_L_Fuel ,                /* Left fuel system failure        */
Prob_R_Fuel ;                /* Right fuel system failure        */

```

APPENDIX 2
SIMULATION TIMING FUNCTION

```

/*****
/*
/*      Configurable Flight Simulator
/*      Cycle Timer
/*
/*      Written by Peter Lawn      Fall 1995
/*      Fluid Power and Simulation Laboratory
/*      Department of Mechanical Engineering
/*      Concordia University
/*      1455 DeMaisonneuve boul West
/*      Montreal, Quebec
/*      CANADA
/*      tel (514)-848-3131
/*      fax (514)-848-3175
/*
/*      This file TIMER.C
/*
/*      This function calculates and regulates the simulation cycle time*/
/*
/*
*****/

void CYCLE_TIMER()
{
#include <time.h>
#include <sys\timeb.h>

#include "glob_inc.c"

    static unsigned long Start_Sec;      /* Starting time in seconds */
    unsigned long        Curr_Sec;      /* Current time in seconds */
    static int           Start_Thou,     /* Starting thousands of a sec */
                    Cycle_Count,        /* Cycle counter */
                    Two_Sec_Limit;      /* Cycles for 2 seconds */

    int                  Curr_Thou;      /* Current thousands of a sec */

    struct timeb t;

    char dummy[30];
    int Key;

    /*-----*/
    /*      PAUSE SIMULATION
    /*-----*/
    /*
    /*      If a simulation pause is requested then the program freezes here until
    /*      <F11> is pressed to resume the simulation or <F12> to end the simulation.
    /*
    if(Cycle_Pause)
    {
        Key = 0;
        while(Key!=11)
        {
            while(kbhit()) getch();
            while(!kbhit());
            while(kbhit()) Key = getch()-58;
            if(Key>70) Key = Key - 64;
            if(Key==26) Key=11;
            if(Key==27) Key=12;
            if(Key==12)
            {
                Cycle_Quit=1;
                Key=11;
            }
        }
        Cycle_Pause = 0;
        First_Cyc   = 2;
    }
}

```

```

/*-----*/
/*  READ AND STORE THE SYSTEM TIME  */
/*-----*/

    ftime(&t);                /* Read the system time          */
    Curr_Sec = t.time;        /* Store current seconds        */
    Curr_Thou= t.millitm;     /* Store current thousands     */

/*-----*/
/*  STORE TIME ON FIRST PASS  */
/*-----*/

    if(First_Cyc)             /* If this is the first itteration */
    {
        Start_Sec = Curr_Sec;
        Start_Thou = Curr_Thou;
        Cycle_Count= 0;
        Two_Sec_Limit = 100;
    }
/*-----*/
/*  NORMAL OPERATION  */
/*-----*/
/*
    Every 2 seconds the itteration rate is checked and updated if needed
*/
    else
    {
        Cycle_Count++;
        if(Cycle_Count==Two_Sec_Limit)
        {
            Cycle_Time = ((Curr_Sec-Start_Sec)
                          +0.001*(Curr_Thou-Start_Thou))/Two_Sec_Limit;
            Start_Sec = Curr_Sec;
            Start_Thou = Curr_Thou;
            Cycle_Count= 0;
            Two_Sec_Limit = 2.0/Cycle_Time;          /* Cycles/ 2sec */
        }
    }

/*-----*/
/*  NORMAL OPERATION  */
/*-----*/
    if(!Real_Time)
        Cycle_Time=Fixed_Time/1000.0;

    return;
}

```

APPENDIX 3
ICAO STANDARD ATMOSPHERE SIMULATION FUNCTION

```

/*****
/*
/*      Configurable Flight Simulator
/*      Atmosphere
/*
/*      Written by Peter Lawn      Fall 1995
/*      Fluid Power and Simulation Laboratory
/*      Department of Mechanical Engineering
/*      Concordia University
/*      1455 DeMaisonneuve boul West
/*      Montreal, Quebec
/*      CANADA
/*      tel (514)-848-3131
/*      fax (514)-848-3175
/*
/*      This file ATMOSPHR.C
/*
/*      This routine calculates the properties of the atmosphere
/*
/*
*****/

void ATMOSPHERE()
{
#include <math.h>
#include "GLOB_INC.C"

    float Altitude,          /* Altitude (m) */
          theta,             /* Temperature ratio */
          delta,             /* Pressure ratio */
          alpha,             /* Density ratio */
          Lapse_Rate = 6.4960629; /* Lapse rate (deg C/m) */

    /*-----*/
    /*      Select fixed or true altitude
    /*-----*/

    if(Fixed_Alt)
        Altitude = Fixed_Altitude;
    else
        Altitude = -Pos_z_FA;

    /*-----*/
    /*      Calculate properties below the isothermal layer ( < 36089 feet) */
    /*-----*/
    if(Altitude<10999.92)
    {
        /*-----*/
        /*      Temperature
        /*-----*/

        Amb_Air_T = To_z0 - (Altitude/1000.0)*Lapse_Rate;
        theta      = Amb_Air_T/To_z0;
        Amb_Air_T = Amb_Air_T-273.15; /* Convert deg K to deg C */

        /*-----*/
        /*      Pressure
        /*-----*/

        delta      = pow(theta,5.2561);
        Amb_Air_P = delta*Po_z0;

        /*-----*/
        /*      Density
        /*-----*/

        alpha      = pow(theta,4.2561);
        Amb_Air_D = alpha*ro_z0;
    }
    else
    /*-----*/
    /*      Calculate properties in the isothermal layer (above 36089 feet) */
    /*-----*/
}

```

```

{
    /*-----*/
    /* Temperature */
    /*-----*/

    Amb_Air_T = -56.5;

    /*-----*/
    /* Pressure */
    /*-----*/

    delta      = 0.2234*exp(-(Altitude-10999.92)/6341.882);
    Amb_Air_P = delta*Po_z0;

    /*-----*/
    /* Density */
    /*-----*/

    alpha      = 0.2971*exp(-(Altitude-10999.92)/6341.882);
    Amb_Air_D = alpha*ro_z0;
}
return;
}

```


APPENDIX 4
CALCULATION OF FORCES IN THE BODY AXIS

```

/*****
/*
/*      Configurable Flight Simulator
/*      Forces in Body Axis
/*
/*      Written by Peter Lawn      Fall 1995
/*      Fluid Power and Simulation Laboratory
/*      Department of Mechanical Engineering
/*      Concordia University
/*      1455 DeMaisonneuve boul West
/*      Montreal, Quebec
/*      CANADA
/*      tel (514)-848-3131
/*      fax (514)-848-3175
/*
/*      This file FORCES.C
/*
/*      This function calculates the forces in the body axis system
/*
*****/

void FORCES_BA()
{
# include <math.h>
# include "glob_inc.c"

    float Q,                /* Dynamic pressure          (N/m^2) */
          Temp_Lift,        /* Temp value of lift during rotation */
                               /* of the lift and drag vectors */

          V_roll_SA,        /* Roll rate in stability axis */
          V_pitch_SA,       /* Pitch rate in stability axis */
          V_yaw_SA;         /* Yaw rate in stability axis */

/*=====
/*
/*      CALCULATE ROLL AND YAW RATE IN STABILITY AXIS
/*
/*=====

    V_roll_SA = cos(Alpha)*V_roll_BA + sin(Alpha)*V_yaw_BA;

    V_pitch_SA = V_pitch_BA;

    V_yaw_SA  = -sin(Alpha)*V_roll_BA + cos(Alpha)*V_yaw_BA;

/*=====
/*
/*      CALCULATE THE DYNAMIC PRESSURE
/*
/*=====

    Q = 0.5 * Amb_Air_D * (Vx_BA*Vx_BA+Vz_BA*Vz_BA);

/*=====
/*
/*      CALCULATE THE LIFT
/*
/*=====

/*-----
/*      Coefficient of Lift
/*-----

    C_L_Tot = C_L
              + C_L_alpha * Alpha
              + C_L_alphaDOT * AlphaDOT*AC_c/(2.0*Ref_Vel)
              + C_L_q * V_pitch_SA*AC_c/(2.0*Ref_Vel)
              + C_L_delta_e * Delta_e_rad;

/*-----
/*      Lift
*/

```

```

/*-----*/
    Lift = Q * AC_S * C_L_Tot;

/*=====*/
/*
/*      CALCULATE THE DRAG
/*
/*=====*/

/*-----*/
/*      Coefficient of Drag
/*-----*/

    C_D_Tot = C_D
              + C_D_alpha * Alpha
              + C_D_delta_e * Delta_e_rad;

/*-----*/
/*      Drag
/*-----*/

    Drag = Q * AC_S * C_D_Tot;

/*=====*/
/*
/*      CALCULATE THE SIDE FORCE
/*
/*=====*/

/*-----*/
/*      Coefficient of Side Force
/*-----*/

    C_y_Tot = C_y
              + C_y_beta * Beta
              + C_y_p * V_roll_SA*AC_b/(2.0*Ref_Vel)
              + C_y_r * V_yaw_SA*AC_b/(2.0*Ref_Vel)
              + C_y_delta_a * Delta_a_rad
              + C_y_delta_r * Delta_r_rad;

/*-----*/
/*      Side Force
/*-----*/

    Fy_BA = AC_Mass*9.806*cos(Pitch_FA)*sin(Roll_FA)
           + Q * AC_S * C_y_Tot
           + LG_Fy_BA;

/*=====*/
/*
/*      ROTATE LIFT AND DRAG FROM WIND TO BODY AXIS SYSTEM
/*
/*=====*/

    Temp_Lift = Lift*cos(Alpha) + Drag*sin(Alpha);

    Drag      = -Lift*sin(Alpha) + Drag*cos(Alpha);
    Lift      = Temp_Lift;

/*-----*/
/*      Vertical Force
/*-----*/

    Fz_BA = AC_Mass*9.806*cos(Pitch_FA)*cos(Roll_FA)
           - Lift
           + LG_Fz_BA;

/*-----*/
/*      Lognitudinal Force
/*-----*/

```

```
Fx_BA = -AC_Mass*9.806*sin(Pitch_FA)
        - Drag
        + Eng_Thrust[0]
        + Eng_Thrust[1]
        + LG_Fx_BA;
return;
}
```

APPENDIX 5

CALCULATION OF MOMENTS IN THE BODY AXIS

```

/*****
/*
/*      Configurable Flight Simulator
/*      Moments in Body Axis
/*
/*      Written by Peter Lawn      Fall 1995
/*      Fluid Power and Simulation Laboratory
/*      Department of Mechanical Engineering
/*      Concordia University
/*      1455 DeMaisonneuve boul West
/*      Montreal, Quebec
/*      CANADA
/*      tel (514)-848-3131
/*      fax (514)-848-3175
/*
/*      This file MOMENTS.C
/*
/*      This function calculates the moments in the body axis system
/*
*****/

void MOMENTS_BA()
{
#include "glob_inc.c"
#include <math.h>

    float Q          ,      /* Dynamic pressure      (N/m^2) */
          M_roll_SA ,      /* Roll moment in stability axis (Nm) */
          M_pitch_SA,      /* Pitch moment in stability axis (Nm) */
          M_yaw_SA  ,      /* Yaw moment in stability axis (Nm) */

          V_roll_SA,      /* Roll rate in stability axis */
          V_pitch_SA,     /* Pitch rate in stability axis */
          V_yaw_SA;       /* Yaw rate in stability axis */

/*****
/*
/*      CALCULATE ROLL AND YA RATE IN STABILITY AXIS
/*
*****/

    V_roll_SA = cos(Alpha)*V_roll_BA + sin(Alpha)*V_yaw_BA;

    V_pitch_SA = V_pitch_BA;

    V_yaw_SA  = -sin(Alpha)*V_roll_BA + cos(Alpha)*V_yaw_BA;

/*****
/*
/*      CALCULATE THE DYNAMIC PRESSURE
/*
*****/

    Q = 0.5 * Amb_Air_D * (Vx_BA*Vx_BA+Vz_BA*Vz_BA);

/*****
/*
/*      CALCULATE THE COEFFICIENTS
/*
*****/

/*****
/*      Pitching Moment Coefficient
/*
*****/

    C_m_Tot = C_m
              + C_m_alpha * Alpha
              + C_m_alphaDOT * AlphaDOT * AC_c/(2.0*Ref_Vel)
              + C_m_q * V_pitch_SA * AC_c/(2.0*Ref_Vel)
              + C_m_delta_e * Delta_e_rad;

/*****

```

```

/*      Rolling Moment Coefficient      */
/*-----*/

C_l_Tot = C_l
      + C_l_beta * Beta
      + C_l_p * V_roll_SA * AC_b/(2.0*Ref_Vel)
      + C_l_r * V_yaw_SA * AC_b/(2.0*Ref_Vel)
      + C_l_delta_a * Delta_a_rad
      + C_l_delta_r * Delta_r_rad;

/*-----*/
/*      Yawing Moment Coefficient      */
/*-----*/

C_n_Tot = C_n
      + C_n_beta * Beta
      + C_n_p * V_roll_SA * AC_b/(2.0*Ref_Vel)
      + C_n_r * V_yaw_SA * AC_b/(2.0*Ref_Vel)
      + C_n_delta_a * Delta_a_rad
      + C_n_delta_r * Delta_r_rad;

/*=====*/
/*      CALCULATE THE MOMENTS IN THE STABILITY AXIS      */
/*=====*/

/*-----*/
/*      Rolling Moment      */
/*-----*/

M_roll_SA = Q * AC_S * AC_b * C_l_Tot;

/*-----*/
/*      Pitching Moment      */
/*-----*/

M_pitch_SA = Q * AC_S * AC_c * C_m_Tot;

/*-----*/
/*      Yawing Moment      */
/*-----*/

M_yaw_SA = Q * AC_S * AC_b * C_n_Tot;

/*=====*/
/*      ROTATE MOMENTS TO BODY AXIS      */
/*=====*/

M_yaw_BA = M_yaw_SA*cos(Alpha) + M_roll_SA*sin(Alpha);
M_roll_BA = -M_yaw_SA*sin(Alpha) + M_roll_SA*cos(Alpha);
M_pitch_BA = M_pitch_SA;

/*=====*/
/*      INCLUDE MOMENTS DUE TO THRUST AND LANDING GEAR      */
/*=====*/

/*-----*/
/*      Rolling Moment      */
/*-----*/

M_roll_BA = M_roll_BA
      + LG_M_roll_BA;

/*-----*/
/*      Pitching Moment      */
/*-----*/

```

```

    M_pitch_BA = M_pitch_BA
                +(Eng_Thrust[0]+Eng_Thrust[1])*Eng_T_Z
                +LG_M_pitch_BA;

/*-----*/
/*      Yawing Moment      */
/*-----*/

    M_yaw_BA = M_yaw_BA
               + (Eng_Thrust[0]-Eng_Thrust[1])*Eng_T_Y
               + LG_M_yaw_BA;

    return;
}

```


APPENDIX 6
CALCULATION OF MOTION IN THE BODY AXIS

```

/*****
/*
/*      Configurable Flight Simulator
/*      Motion in Body Axis
/*
/*      Written by Peter Lawn      Fall 1995
/*      Fluid Power and Simulation Laboratory
/*      Department of Mechanical Engineering
/*      Concordia University
/*      1455 DeMaisonneuve boul West
/*      Montreal, Quebec
/*      CANADA
/*      tel (514)-848-3131
/*      fax (514)-848-3175
/*
/*      This file MOTIONBA.C
/*
/*      This function calculates the motion in the body axis system.
/*
*****/

void MOTION_BA()
{
#include <math.h>
#include "glob_inc.c"

    float    A_roll_BA,          /* Roll-acceleration in body axis */
             A_pitch_BA,         /* Pitch-acceleration in body axis */
             A_yaw_BA,           /* Roll-acceleration in body axis */

             New_Alpha,

             A,                  /* Inetrmediace variable */
             B,                  /* Inetrmediace variable */
             C,                  /* Inetrmediace variable */

             V;                  /* Net velocity */

/*-----*/
/*      LINEAR ACCELERATIONS IN BODY AXIS
/*-----*/

    Ax_BA = Fx_BA/AC_Mass - V_pitch_BA*Vz_BA + V_yaw_BA*Vy_BA;
    Ay_BA = Fy_BA/AC_Mass - V_yaw_BA*Vx_BA + V_roll_BA*Vz_BA;
    Az_BA = Fz_BA/AC_Mass - V_roll_BA*Vy_BA + V_pitch_BA*Vx_BA;

/*-----*/
/*      ROTATIONAL ACCELERATIONS IN BODY AXIS
/*-----*/

    A = (-V_pitch_BA*V_yaw_BA*(AC_Iz-AC_Iy)
        +AC_Ixz*V_roll_BA*V_pitch_BA)/AC_Ix;

    B = (-V_yaw_BA*V_roll_BA*(AC_Ix-AC_Iz)
        -AC_Ixz*(V_roll_BA*V_roll_BA-V_yaw_BA*V_yaw_BA))/AC_Iy;

    C = (-V_roll_BA*V_pitch_BA*(AC_Iy-AC_Ix)
        -AC_Ixz*V_pitch_BA*V_yaw_BA)/AC_Iz;

    A_roll_BA = ((M_roll_BA+AC_Ixz*(M_yaw_BA/AC_Iz+C))/AC_Ix+A)
        /(1.0-AC_Ixz*AC_Ixz/(AC_Ix*AC_Iz));

    A_pitch_BA = M_pitch_BA/AC_Iy+B;

    A_yaw_BA = M_yaw_BA/AC_Iz + AC_Ixz/AC_Iz*A_roll_BA+C;
/*-----*/

```

```

/*  INTEGRATE LINEAR ACCELERATIONS TO CALCULATE VELOCITIES  */
/*-----*/

Vx_BA = Vx_BA + Ax_BA * Cycle_Time;
Vy_BA = Vy_BA + Ay_BA * Cycle_Time;
Vz_BA = Vz_BA + Az_BA * Cycle_Time;

/*-----*/
/*  INTEGRATE ROTATIONAL ACCELERATIONS TO CALCULATE VELOCITIES  */
/*-----*/

V_roll_BA = V_roll_BA + A_roll_BA*Cycle_Time;
V_pitch_BA = V_pitch_BA + A_pitch_BA*Cycle_Time;
V_yaw_BA   = V_yaw_BA   + A_yaw_BA*Cycle_Time;
Vx_SA =sqrt(Vx_BA*Vx_BA+Vz_BA*Vz_BA);

/*-----*/
/*  CALCULATE SIDESLIP AND ATTACK ANGLES  */
/*-----*/

V = sqrt(Vx_BA*Vx_BA + Vy_BA*Vy_BA + Vz_BA*Vz_BA);
if(Vx_BA!=0.0)      /* Don't calc if Vx_BA=0  */
    New_Alpha = atan2(Vz_BA,Vx_BA); /* Angle of attack  */
else
    New_Alpha = 0;

if(V!=0.0)      /* Don't calc if V=0  */
    Beta = asin(Vy_BA/V); /* Sideslip angle  */
else
    Beta = 0;

AlphaDOT = (New_Alpha-Alpha)/Cycle_Time;
Alpha = New_Alpha;

return;
}

```

APPENDIX 7

CALCULATION OF MOTION IN THE FIXED AXIS

```

/*****
/*
/*      Configurable Flight Simulator
/*      Motion in Fixed Axis
/*
/*      Written by Peter Lawn      Fall 1995
/*      Fluid Power and Simulation Laboratory
/*      Department of Mechanical Engineering
/*      Concordia University
/*      1455 DeMaisonneuve boul West
/*      Montreal, Quebec
/*      CANADA
/*      tel (514)-848-3131
/*      fax (514)-848-3175
/*
/*      This file MOTIONFA.C
/*
/*      This function calculates the motion in the fixed axis system.
/*
*****/

#define PI 3.141592653

void MOTION_FA()
{
#include <math.h>
#include "glob_inc.c"

float Sp,                /* Sin pitch
Cp,                    /* Cos pitch
Tp,                    /* Tan pitch

Sr,                    /* Sin roll
Cr,                    /* Cos roll

Sy,                    /* Sin yaw
Cy,                    /* Cos yaw

Vcg_BA_KM,            /* cgvelocity in Body axis for kin model*/
R_Main,                /* Turning radius at main gear
R_CG,                  /* Turning radius at center of gravity
Tau,                  /* Vcg Angle from X body axis
Temp;

Sp = sin(Pitch_FA);
Cp = cos(Pitch_FA);
Tp = Sp/Cp;

Sr = sin(Roll_FA);
Cr = cos(Roll_FA);

Sy = sin(Yaw_FA);
Cy = cos(Yaw_FA);

/-----*/
/*
/*      CALCULATE VELOCITIES IN THE FIXES AXIS SYSTEM
/*
*****/

Vx_FA = (Cp*Cy)*Vx_BA
        + (Cr*Cp*Cy-Cr*Sy)*Vy_BA
        + (Cr*Sp*Cy+Sr*Sy)*Vz_BA;

Vy_FA = (Cp*Sy)*Vx_BA
        + (Sr*Sp*Sy+Cr*Cy)*Vy_BA
        + (Cr*Sp*Sy-Sr*Cy)*Vz_BA;

Vz_FA = (-Sp)*Vx_BA
        + (Sr*Cp)*Vy_BA
        + (Cr*Cp)*Vz_BA;

```

```

/*-----*/
/*
/*      CALCULATE ACCELERATIONS IN FIXED AXIS SYSTEM
/*      (used only for magnetic compass errors)
/*
/*-----*/

Ax_FA = (Cp*Cy)*Ax_BA
      + (Cr*Cp*Cy-Cr*Sy)*Ay_BA
      + (Cr*Sp*Cy+Sr*Sy)*Az_BA;

Ay_FA = (Cp*Sy)*Ax_BA
      + (Sr*Sp*Sy+Cr*Cy)*Ay_BA
      + (Cr*Sp*Sy-Sr*Cy)*Az_BA;

Az_FA = (-Sp)*Ax_BA
      + (Sr*Cp)*Ay_BA
      + (Cr*Cp)*Az_BA;

/*-----*/
/*
/*      CALCULATE ACCELERATIONS IN MAGNETIC AXIS SYSTEM
/*      (used only for magnetic compass errors)
/*
/*-----*/

Mag_E_Accel = Ay_FA*cos(Mag_Var*PI/180.0)-Ax_FA*sin(Mag_Var*PI/180.0);
Mag_N_Accel = Ay_FA*sin(Mag_Var*PI/180.0)+Ax_FA*cos(Mag_Var*PI/180.0);

/*-----*/
/*
/*      CALCULATE ROTATIONAL VELOCITIES IN THE FIXES AXIS SYSTEM
/*
/*-----*/

V_roll_FA = V_roll_BA
          + (Sr*Tp)*V_pitch_BA
          + (Cr*Tp)*V_yaw_BA;

V_pitch_FA = (Cr)*V_pitch_BA
            + (-Sr)*V_yaw_BA;

V_yaw_FA = (Sr/Cp)*V_pitch_BA
          + (Cr/Cp)*V_yaw_BA;

/*-----*/
/*      KINEMATIC GROUND MODEL
/*
/*      IF ALL 3 WHEELS ARE ON THE GROUND THEN SWITCH TO KINEMATIC
/*      MODEL FOR Vx, Vy AND THE YAW RATE
/*      NOT FOR PITCH, ROLL, AND VERTICAL SPEEDS
/*
/*-----*/

Temp = LG_F_Z_Nose * LG_F_Z_Left * LG_F_Z_Right;

LG_AOG = 0;
if(Temp!=0.0 && Crash_On)
{
  LG_AOG = 1;
  if(LG_Steer_Ang!=0.0)
  {
    R_Main = (LG_X_Nose_BA-LG_X_Main_BA)/tan(LG_Steer_Ang);

    Tau    = atan(-LG_X_Main_BA/R_Main);

    R_CG   = R_Main/cos(Tau);

    Vcg_BA_KM= sqrt(Vx_FA*Vx_FA + Vy_FA*Vy_FA);

    V_yaw_FA = Vcg_BA_KM/R_CG;
  }
}

```

```

        Vx_FA = Vcg_BA_KM*cos(Yaw_FA + Tau);
        Vy_FA = Vcg_BA_KM*sin(Yaw_FA + Tau);
    }
    else
    {
        V_yaw_FA = 0.0;
        Vx_FA = Vx_BA*cos(Yaw_FA);
        Vy_FA = Vx_BA*sin(Yaw_FA);
    }
}
return;
}

```

APPENDIX 8

CALCULATION OF POSITION IN THE FIXED AXIS


```

/*****
/*
/*      Configurable Flight Simulator
/*      Motion in Fixed Axis
/*
/*      Written by Peter Lawn      Fall 1995
/*      Fluid Power and Simulation Laboratory
/*      Department of Mechanical Engineering
/*      Concordia University
/*      1455 DeMaisonneuve boul West
/*      Montreal, Quebec
/*      CANADA
/*      tel (514)-848-3131
/*      fax (514)-848-3175
/*
/*      This file MOTIONFA.C
/*
/*      This function calculates the motion in the fixed axis system.
/*
*****/

void POSITION_FA()
{
# include <math.h>
# include "glob_inc.c"

#define PI 3.141592653
#define ER 3437.746771          /* Earth's radius in Nm */

/*-----*/
/*
/*      CALCULATE POSITION IN THE FIXES AXIS SYSTEM
/*      CARTESIAN COORDINATES
/*
/*-----*/

    Pos_x_FA = Pos_x_FA + Vx_FA*Cycle_Time;

    Pos_y_FA = Pos_y_FA + Vy_FA*Cycle_Time;

    Pos_z_FA = Pos_z_FA + Vz_FA*Cycle_Time;

/*-----*/
/*
/*      CALCULATE POSITION IN THE FIXES AXIS SYSTEM
/*      POLAR COORDINATES
/*
/*-----*/

    Pos_Lat_Dec = Pos_Lat_Dec + (Vx_FA*1.9438/60.0)*Cycle_Time/3600.0;

    Pos_Lng_Dec = Pos_Lng_Dec
        + (Vy_FA*1.9438/60.0)/cos(Pos_Lat_Dec*PI/180.0)*Cycle_Time/3600.0;

    Pos_Lat_Deg = Pos_Lat_Dec;
    Pos_Lat_Min = 60.0*(Pos_Lat_Dec - Pos_Lat_Deg);
    Pos_Lat_Sec = 3600.0*(Pos_Lat_Dec - Pos_Lat_Deg - Pos_Lat_Min/60.0)+0.5;

    Pos_Lng_Deg = Pos_Lng_Dec;
    Pos_Lng_Min = 60.0*(Pos_Lng_Dec - Pos_Lng_Deg);
    Pos_Lng_Sec = 3600.0*(Pos_Lng_Dec - Pos_Lng_Deg - Pos_Lng_Min/60.0)+0.5;

/*-----*/
/*
/*      CALCULATE ORIENTATION IN THE FIXES AXIS SYSTEM
/*
/*-----*/

    Roll_FA = Roll_FA + V_roll_FA*Cycle_Time;

```

```

Pitch_FA = Pitch_FA + V_pitch_FA*Cycle_Time;
Yaw_FA = Yaw_FA + V_yaw_FA*Cycle_Time;
Head_Mag = Yaw_FA*180.0/PI - Mag_Var;
if(Roll_FA> PI) Roll_FA=Roll_FA-2.0*PI;
if(Roll_FA<-PI) Roll_FA=Roll_FA+2.0*PI;

if(Pitch_FA> PI) Pitch_FA=Pitch_FA-2.0*PI;
if(Pitch_FA<-PI) Pitch_FA=Pitch_FA+2.0*PI;

if(Yaw_FA>(2.0*PI)) Yaw_FA = Yaw_FA-2.0*PI;
if(Yaw_FA<0) Yaw_FA = Yaw_FA+2.0*PI;

return;
}

```

APPENDIX 9
MAGNETIC VARIATION AND DIP MODEL

```

/*****
/*
/*      Configurable Flight Simulator
/*      Atmosphere
/*
/*      Written by Peter Lawn      Fall 1995
/*      Fluid Power and Simulation Laboratory
/*      Department of Mechanical Engineering
/*      Concordia University
/*      1455 DeMaisonneuve boul West
/*      Montreal, Quebec
/*      CANADA
/*      tel (514)-848-3131
/*      fax (514)-848-3175
/*
/*      This file MAG_VAR.C
/*
/*      This routine calculates the earth's magnetic variation
/*
/*      Because this routine involves a significant amount of
/*      calculations and does not require a rapid update rate it
/*      is subanded and solved over a total of 316 itterations.
/*
/*      The equation numbers relate to Peter Lawn's Master's thesis
/*
*****/

#include <math.h>
#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include <dos.h>
#include "glob_inc.c"
#define PI 3.141592653

void MAGNETIC_VARIATION()
{
    static
    float g[11][11],
          h[11][11],
          K[11][11],
          P[11][11],
          S[11][11],
          DP[11][11],
          N,E,R,
          Lat,
          Lon;

    static int n,m;
    static int Band = 0;
    static int Prev_Band = 9;
    static int First = 0;
    static int calc_count = 0;

    float Tg,          /* Tempoary value of g while loading */
          Th,          /* Tempoary value of h while loading */
          J;
    char dummy[256];

    FILE *in;

/*****
/*      FIRST PASS
*****/
    if(First==0)
    {
        First = 1;

        Lat = Pos_Lat_Dec * PI/180.0;
        Lon = Pos_Lng_Dec * PI/180.0;
    }
}

```

```

/*****
/* INPUT DATA
/*
/* Normalized Spherical
/* Harmonic Coefficienty
/*
/* Set on first pass only
/*
/*****
in = fopen("DGRF90.DAT","rt");

if(in==0)
{
/*****
/* If file not found, use 1990 data
/*****

g[1][0] = -29775.0 ; h[1][0] = 0.0 ;
g[1][1] = -1848.0 ; h[1][1] = 5406.0 ;
g[2][0] = -2131.0 ; h[2][0] = 0.0 ;
g[2][1] = 3059.0 ; h[2][1] = -2279.0 ;
g[2][2] = 1686.0 ; h[2][2] = -373.0 ;
g[3][0] = 1314.0 ; h[3][0] = 0.0 ;
g[3][1] = -2239.0 ; h[3][1] = -284.0 ;
g[3][2] = 1248.0 ; h[3][2] = 293.0 ;
g[3][3] = 802.0 ; h[3][3] = -352.0 ;
g[4][0] = 939.0 ; h[4][0] = 0.0 ;
g[4][1] = 780.0 ; h[4][1] = 247.0 ;
g[4][2] = 325.0 ; h[4][2] = -240.0 ;
g[4][3] = -423.0 ; h[4][3] = 84.0 ;
g[4][4] = 141.0 ; h[4][4] = -299.0 ;
g[5][0] = -214.0 ; h[5][0] = 0.0 ;
g[5][1] = 353.0 ; h[5][1] = 46.0 ;
g[5][2] = 245.0 ; h[5][2] = 154.0 ;
g[5][3] = -109.0 ; h[5][3] = -153.0 ;
g[5][4] = -165.0 ; h[5][4] = -69.0 ;
g[5][5] = -36.0 ; h[5][5] = 97.0 ;
g[6][0] = 61.0 ; h[6][0] = 0.0 ;
g[6][1] = 65.0 ; h[6][1] = -16.0 ;
g[6][2] = 59.0 ; h[6][2] = 82.0 ;
g[6][3] = -178.0 ; h[6][3] = 69.0 ;
g[6][4] = 3.0 ; h[6][4] = -52.0 ;
g[6][5] = 18.0 ; h[6][5] = 1.0 ;
g[6][6] = -96.0 ; h[6][6] = 24.0 ;
g[7][0] = 77.0 ; h[7][0] = 0.0 ;
g[7][1] = -64.0 ; h[7][1] = -80.0 ;
g[7][2] = 2.0 ; h[7][2] = -26.0 ;
g[7][3] = 26.0 ; h[7][3] = 0.0 ;
g[7][4] = -1.0 ; h[7][4] = 21.0 ;
g[7][5] = 5.0 ; h[7][5] = 17.0 ;
g[7][6] = 9.0 ; h[7][6] = -23.0 ;
g[7][7] = 0.0 ; h[7][7] = -4.0 ;
g[8][0] = 23.0 ; h[8][0] = 0.0 ;
g[8][1] = 5.0 ; h[8][1] = 10.0 ;
g[8][2] = -1.0 ; h[8][2] = -19.0 ;
g[8][3] = -10.0 ; h[8][3] = 6.0 ;
g[8][4] = -12.0 ; h[8][4] = -22.0 ;
g[8][5] = 3.0 ; h[8][5] = 12.0 ;
g[8][6] = 4.0 ; h[8][6] = 12.0 ;
g[8][7] = 2.0 ; h[8][7] = -16.0 ;
g[8][8] = -6.0 ; h[8][8] = -10.0 ;
g[9][0] = 4.0 ; h[9][0] = 0.0 ;
g[9][1] = 9.0 ; h[9][1] = -20.0 ;
g[9][2] = 1.0 ; h[9][2] = 15.0 ;
g[9][3] = -12.0 ; h[9][3] = 11.0 ;
g[9][4] = 9.0 ; h[9][4] = -7.0 ;
g[9][5] = -4.0 ; h[9][5] = -7.0 ;
g[9][6] = -2.0 ; h[9][6] = 9.0 ;
g[9][7] = 7.0 ; h[9][7] = 8.0 ;
g[9][8] = 1.0 ; h[9][8] = -7.0 ;
g[9][9] = -6.0 ; h[9][9] = 2.0 ;
g[10][0] = -3.0 ; h[10][0] = 0.0 ;
g[10][1] = -4.0 ; h[10][1] = 2.0 ;

```

```

        g[10][2]=      2.0 ; h[10][2]=      1.0 ;
        g[10][3]=     -5.0 ; h[10][3]=      3.0 ;
        g[10][4]=     -2.0 ; h[10][4]=      6.0 ;
        g[10][5]=      4.0 ; h[10][5]=     -4.0 ;
        g[10][6]=      3.0 ; h[10][6]=      0.0 ;
        g[10][7]=      1.0 ; h[10][7]=     -2.0 ;
        g[10][8]=      3.0 ; h[10][8]=      3.0 ;
        g[10][9]=      3.0 ; h[10][9]=     -1.0 ;
        g[10][10]=     0.0 ; h[10][10]=    -6.0 ;

    }
    else
    {
        fgets(dummy,255,in);
        fgets(dummy,255,in);
        fscanf(in,"%d %d %f %f",&n,&m,&Tg,&Th);
        fgets(dummy,255,in);
        while(!feof(in))
        {
            g[n][m] = Tg;
            h[n][m] = Th;
            fscanf(in,"%d %d %f %f",&n,&m,&Tg,&Th);
            fgets(dummy,255,in);
        }
    }
    fclose(in);

/* -----*/
/* Equation 4.13 */
/* -----*/
S[0][0] = 1.0;
for(n=1;n<=10;n++)
{
    for(m=0;m<=n;m++)
    {
        if(m==1)
            J = 2.0;
        else
            J = 1.0;

        if(m==0)
/* -----*/
/* Equation 4.14 */
/* -----*/
            S[n][m] = (2.0*n-1.0)/n*S[n-1][0];
        else
/* -----*/
/* Equation 4.15 & 4.16 */
/* -----*/
            S[n][m] = sqrt((n-m+1.0)*J/(n+m))*S[n][m-1];

/* -----*/
/* Equation 4.17 */
/* -----*/
            K[n][m] = ((n-1.0)*(n-1.0)-m*m)/((2.0*n-1.0)*(2.0*n-3.0));
    }
}

Band = 1;
}

switch(Band)
{
/******
/* FIRST BAND */
/******
case(1):
{
    if(Prev_Band!=Band)
    {
        n = -1;

```

```

        m = 0;
        Prev_Band = Band;
        calc_count = 1;
    }
    n++;
    if(n>10)
    {
        m++;
        n = 0;
    }

    P[n][m] = 0.0;
    DP[n][m] = 0.0;

    if(m==10 && n==10)
    {
        /*-----*/
        /*      Equation 4.8 */
        /*-----*/
        P[0][0] = 1.0;

        /*-----*/
        /*      Equation 4.11 */
        /*-----*/
        DP[0][0] = 0.0;
        Band = 2;
    }
    break;
}

/*****
/*      SECOND BAND
*****/
case(2):
{
    if(Prev_Band!=Band)
    {
        n = 1;
        m = -1;
        Prev_Band = Band;
    }

    m++;
    if(m>n)
    {
        n++;
        m=0;
    }

    P[n][m] = 0.0 ;
    DP[n][m] = 0.0 ;

    if(n==m)
    {
        /*      -----*/
        /*      Equation 4.9 */
        /*      -----*/
        P[n][m] = P[n-1][m-1]*cos(Lat);

        /*      -----*/
        /*      Equation 4.11 */
        /*      -----*/
        DP[n][m]= DP[n-1][m-1]*cos(Lat)+P[n-1][m-1]*sin(Lat);
    }
    else
    {
        if(n>=2)
        {
            /*      -----*/
            /*      Equation 4.10 */
            /*      -----*/
            P[n][m] = P[n-1][m]*sin(Lat)

```

```

-K[n][m]*P[n-2][m];

/*      -----*/
/*      Equation 4.12 */
/*      -----*/
DP[n][m] = DP[n-1][m]*sin(Lat)
          -P[n-1][m]*cos(Lat)
          -K[n][m]*DP[n-2][m];
    }
    else
    {
/*      -----*/
/*      Equation 4.10 */
/*      -----*/
P[n][m] = P[n-1][m]*sin(Lat);

/*      -----*/
/*      Equation 4.12 */
/*      -----*/
DP[n][m] = DP[n-1][m]*sin(Lat)
          -P[n-1][m]*cos(Lat);
    }
}

if(n==10 && m==10)
{
    Band = 3;
}
break;
}

/*****
/*      THIRD BAND
*****/
case(3):
{
    if(Prev_Band!=Band)
    {
        n = 1;
        m = -1;
        Prev_Band = Band;
    }
    m++;
    if(m>n)
    {
        n++;
        m=0;
    }

/*      -----*/
/*      Equation 4.5 */
/*      -----*/
P[n][m] = S[n][m] * P[n][m];

/*      -----*/
/*      Equation 4.6 */
/*      -----*/
DP[n][m] = S[n][m] * DP[n][m];

    if(n==10 && m==10)
    {
        Band = 4;
    }
    break;
}

/*****
/*      FOURTH BAND
*****/
case(4):
{
    if(Prev_Band!=Band)
    {
        n = 1;
        m = 0;

```



```

        E = 0.0;
        N = 0.0;
        R = 0.0;
        Prev_Band = Band;
    }

    /* N - S field potential */
    N = N - (g[n][m]*cos(m*Lon) + h[n][m]*sin(m*Lon))*DP[n][m];

    /* E - W field potential */
    E = E + (1.0*m*(g[n][m]*sin(m*Lon) - h[n][m]*cos(m*Lon)))*P[n][m];

    /* Radial field potential */
    R = R + (1.0*(n+1.0)*m*(g[n][m]*cos(m*Lon)
        + h[n][m]*sin(m*Lon)))*P[n][m];

    m++;
    if(m>n)
    {
        n++;
        m=0;
    }
    if(n>10)
    {
        E = E/sin(Lat);
        Mag_Var = -atan(E/N)*180.0/PI;
        Mag_Dip = atan(R/sqrt(N*N+E*E))*180.0/PI;
        B_Lat = N;
        B_Lon = E;
        B_Rad = R;
        n = 1;
        m = 0;
        E = 0.0;
        N = 0.0;
        R = 0.0;
        Lat = Pos_Lat_Dec * PI/180.0;
        Lon = Pos_Lng_Dec * PI/180.0;
        Band = 1;
        calc_count = 0;
    }
    break;
}
}
calc_count++;
return;
}

void INITIAL_VARIATION()
{
    Mag_Var = 900.0;
    while(Mag_Var==900.0)
        MAGNETIC_VARIATION();
    return;
}

```

APPENDIX 10
RANGE AND BEARING FUNCTION

```

/*****
/*
/*      Configurable Flight Simulator
/*      Cycle Timer
/*
/*      Written by Peter Lawn      Fall 1995
/*      Fluid Power and Simulation Laboratory
/*      Department of Mechanical Engineering
/*      Concordia University
/*      1455 DeMaisonneuve boul West
/*      Montreal, Quebec
/*      CANADA
/*      tel (514)-848-3131
/*      fax (514)-848-3175
/*
/*      This file AVIONICS.C
/*
/*      This file preforms the avionics routines
/*
/*
*****/

#define deg2rad 3.141592653/180.0      /* Deg to Rad conversion factor */
#define rad2deg 180.0/3.141592653    /* Rad to Deg conversion factor */
#define Rearth 3437.746771           /* Radius of the earth in nm */
#define m2nm 1.0/1852.0              /* m to nm conversion */
#define ft2nm 1.0/6076.1;            /* ft to nm conversion */

#include <math.h>
#include "glob_inc.c"

void DISP_DME(int DD_Hours, int DD_Mins, float DD_Vel, float DD_Dist);
void DISP_ADF(float Angle);
void DISP_VOR();

/*=====*/
/*
/* Calculate the distance in nm between two points
/*
/*=====*/
float AV_RANGE(double Lat1, double Lon1, double Alt1,
               double Lat2, double Lon2, double Alt2)
{
    double X1,Y1,Z1,
           X2,Y2,Z2;

    float Range;

/* -----*/
/* Convert coordinated from degrees to radians */
/* -----*/

    Lat1 = Lat1*deg2rad;
    Lon1 = Lon1*deg2rad;

    Lat2 = Lat2*deg2rad;
    Lon2 = Lon2*deg2rad;

/* -----*/
/* Convert altitude from m ASL to mn ASL */
/* -----*/

    Alt1 = Alt1*m2nm + Rearth;
    Alt2 = Alt2*m2nm + Rearth;

/* -----*/
/* Convert coordinated to from polar to cartesian */
/* -----*/

    X1 = Alt1*cos(Lat1)*cos(Lon1);
    Y1 = Alt1*cos(Lat1)*sin(Lon1);
    Z1 = Alt1*sin(Lat1);

```

```

        X2 = Alt2*cos(Lat2)*cos(Lon2);
        Y2 = Alt2*cos(Lat2)*sin(Lon2);
        Z2 = Alt2*sin(Lat2);

/* -----*/
/* Calculate range in nm */
/* -----*/

        Range = sqrt( (X1-X2)*(X1-X2)
                      +(Y1-Y2)*(Y1-Y2)
                      +(Z1-Z2)*(Z1-Z2));

        return Range;
}

/*=====*/
/* Calculate the bearing from one point to another */
/*=====*/
float AV_BEARING(double LatF, double LonF, double AltF,
                 double LatT, double LonT, double AltT)
{
    double XF,YF,ZF,
           XT,YT,ZT;
    static float Bearing;

/* -----*/
/* Convert coordinated from degrees to radians */
/* -----*/

    LatF = LatF*deg2rad;
    LonF = LonF*deg2rad;

    LatT = LatT*deg2rad;
    LonT = LonT*deg2rad;

/* -----*/
/* Convert altitude from feet ASL to mn ASL */
/* -----*/

    AltF = AltF*m2nm + Rearth;
    AltT = AltT*m2nm + Rearth;

/* -----*/
/* Rotate both points so that FROM is at 0 Lon */
/* -----*/

    LonT = LonT - LonF;

/* -----*/
/* Convert coordinated to from polar to cartesian */
/* -----*/

    XT = AltT*cos(LatT)*cos(LonT);
    YT = AltT*cos(LatT)*sin(LonT);
    ZT = AltT*sin(LatT);

/* -----*/
/* Rotate both points so that FROM is at N Pole */
/* -----*/

    XT = XT*sin(LatF) - ZT*cos(LatF);

/* -----*/
/* Calculate bearing from FROM to TO */
/* -----*/

    if(XT == 0.0)

```

```

    {
        if(YT>=0.0)
            Bearing = 90.0;
        else
            Bearing = -90.0;
    }
    else
    {
        Bearing = atan2(YT,-XT)*rad2deg;
    }
    return Bearing;
}

```

APPENDIX 11
VOR, LOCALIZER AND GLIDESLOPE FREQUENCIES

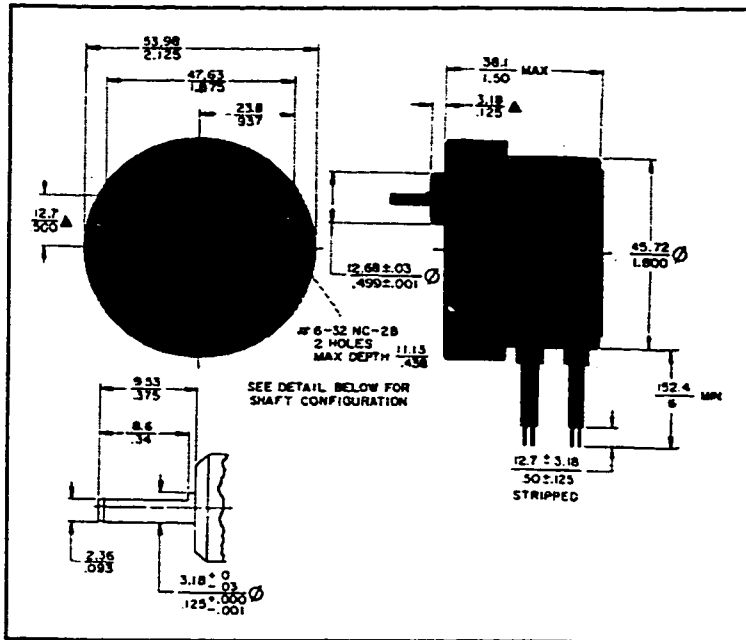
Freq (MHz)	VOR/LOC	G/S (MHz)	110.15	ILS	334.25
108.00	VOR		110.20	VOR	
108.05	VOR		110.25	VOR	
108.10	ILS	334.70	110.30	ILS	335.00
108.15	ILS	334.55	110.35	ILS	334.85
108.20	VOR		110.40	VOR	
108.25	VOR		110.45	VOR	
108.30	ILS	334.10	110.50	ILS	329.60
108.35	ILS	333.95	110.55	ILS	329.45
108.40	VOR		110.60	VOR	
108.45	VOR		110.65	VOR	
108.50	ILS	329.90	110.70	ILS	330.20
108.55	ILS	329.75	110.75	ILS	330.05
108.60	VOR		110.80	VOR	
108.65	VOR		110.85	VOR	
108.70	ILS	330.50	110.90	ILS	330.80
108.75	ILS	330.35	110.95	ILS	330.65
108.80	VOR		111.00	VOR	
108.85	VOR		111.05	VOR	
108.90	ILS	329.30	111.10	ILS	331.70
108.95	ILS	329.15	111.15	ILS	331.55
109.00	VOR		111.20	VOR	
109.05	VOR		111.25	VOR	
109.10	ILS	331.40	111.30	ILS	332.30
109.15	ILS	331.25	111.35	ILS	332.15
109.20	VOR		111.40	VOR	
109.25	VOR		111.45	VOR	
109.30	ILS	332.00	111.50	ILS	332.90
109.35	ILS	331.85	111.55	ILS	332.75
109.40	VOR		111.60	VOR	
109.45	VOR		111.65	VOR	
109.50	ILS	332.60	111.70	ILS	333.50
109.55	ILS	332.45	111.75	ILS	333.35
109.60	VOR		111.80	VOR	
109.65	VOR		111.85	VOR	
109.70	ILS	333.20	111.90	ILS	331.10
109.75	ILS	333.05	111.95	ILS	330.95
109.80	VOR		112.00	VOR	
109.85	VOR		112.05	VOR	
109.90	ILS	333.80	112.10	VOR	
109.95	ILS	333.65	112.15	VOR	
110.00	VOR		112.20	VOR	
110.05	VOR		112.25	VOR	
110.10	ILS	334.40	112.30	VOR	
Freq (MHz)	VOR/LOC	G/S (MHz)	112.35	VOR	

112.40	VOR	114.65	VOR
112.45	VOR	114.70	VOR
112.50	VOR	114.75	VOR
112.55	VOR	114.80	VOR
112.60	VOR	114.85	VOR
112.65	VOR	114.90	VOR
112.70	VOR	114.95	VOR
112.75	VOR	115.00	VOR
112.80	VOR	115.05	VOR
112.85	VOR	115.10	VOR
112.90	VOR	115.15	VOR
112.95	VOR	115.20	VOR
113.00	VOR	115.25	VOR
113.05	VOR	115.30	VOR
113.10	VOR	115.35	VOR
113.15	VOR	115.40	VOR
113.20	VOR	115.45	VOR
113.25	VOR	115.50	VOR
113.30	VOR	115.55	VOR
113.35	VOR	115.60	VOR
113.40	VOR	115.65	VOR
113.45	VOR	115.70	VOR
113.50	VOR	115.75	VOR
113.55	VOR	115.80	VOR
113.60	VOR	115.85	VOR
113.65	VOR	115.90	VOR
113.70	VOR	115.95	VOR
113.75	VOR	116.00	VOR
113.80	VOR	116.05	VOR
113.85	VOR	116.10	VOR
113.90	VOR	116.15	VOR
113.95	VOR	116.20	VOR
114.00	VOR	116.25	VOR
114.05	VOR	116.30	VOR
114.10	VOR	116.35	VOR
114.15	VOR	116.40	VOR
114.20	VOR	116.45	VOR
114.25	VOR	116.50	VOR
114.30	VOR	116.55	VOR
114.35	VOR	116.60	VOR
114.40	VOR	116.65	VOR
114.45	VOR	116.70	VOR
114.50	VOR	116.75	VOR
114.55	VOR	116.80	VOR
114.60	VOR	116.85	VOR

116.90	VOR
116.95	VOR
117.00	VOR
117.05	VOR
117.10	VOR
117.15	VOR
117.20	VOR
117.25	VOR
117.30	VOR
117.35	VOR
117.40	VOR
117.45	VOR
117.50	VOR
117.55	VOR
117.60	VOR
117.65	VOR
117.70	VOR
117.75	VOR
117.80	VOR
117.85	VOR
117.90	VOR
117.95	VOR

APPENDIX 12
SELECTED STEPPER MOTOR SPECIFICATIONS

AIRPAC K82401-P1



Specifications

DC Operating Voltage	5
Res. Per Winding Ω	26
Ind. Per Winding mH	9
Holding Torque oz-in	1.3
Step Angle	18°
Step Angle Tolerance	1.2°
Steps per Rev.	20
Rot Mom of Inertia gm ²	2x10 ⁻⁴
Max Operating Temp	100
Amb Operating Temp	-20 to 70 °C
Amb Storage Temp	-40 to 85 °C
Insulation Res @ 500Vdc	100m Ω
Bronze Sleeve Bearings	
Weight oz	3
Lead Wire AWG	26

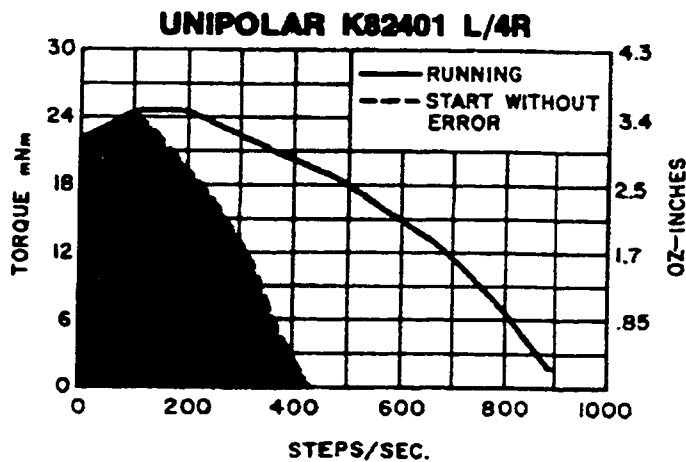
Gear Reduction

K82-19

Gear Ratio	7.5 : 1
Output Step Angle	1.00°

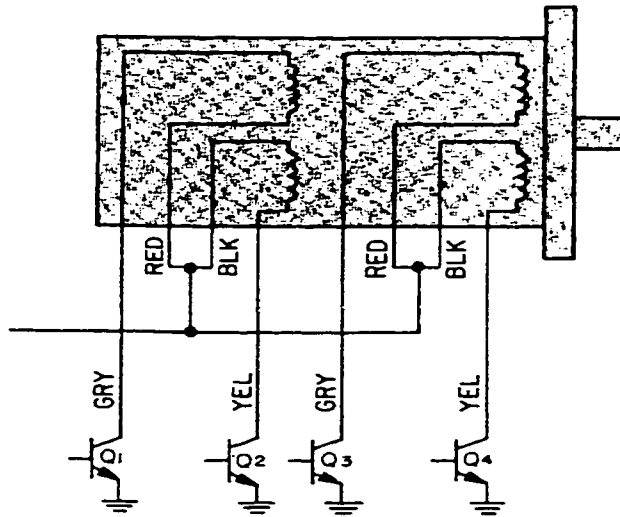
K82-31

Gear Ratio	30 : 1
Output Step Angle	0.25°



AIRPAC®
NORTH AMERICAN PHILIPS CONTROLS CORP.

Cheshire Division
Cheshire Industrial Park
Cheshire, CT 06410
(203)-272-0301



UNIPOLAR

**Normal
4 Step Sequence**

	Step	Q ₁	Q ₂	Q ₃	Q ₄	
CW ROTATION	1	ON	OFF	ON	OFF	CCW ROTATION
	2	ON	OFF	OFF	ON	
	3	OFF	ON	OFF	ON	
	4	OFF	ON	ON	OFF	
	1	ON	OFF	ON	OFF	

**1/2 Step
8 Step Sequence**

CW ROTATION	1	ON	OFF	ON	OFF	CCW ROTATION
	2	ON	OFF	OFF	OFF	
	3	ON	OFF	OFF	ON	
	4	OFF	OFF	OFF	ON	
	5	OFF	ON	OFF	ON	
	6	OFF	ON	OFF	OFF	
	7	OFF	ON	ON	OFF	
	8	OFF	OFF	ON	OFF	
	1	ON	OFF	ON	OFF	

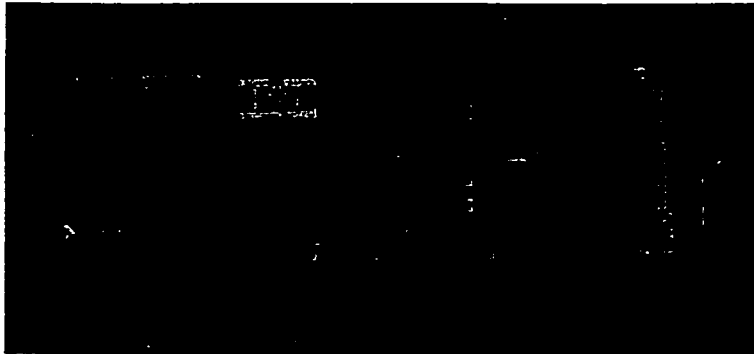
**Wave Drive
4 Step Sequence**

CW ROTATION	1	ON	OFF	OFF	OFF	CCW ROTATION
	2	OFF	OFF	OFF	ON	
	3	OFF	ON	OFF	OFF	
	4	OFF	OFF	ON	OFF	
	1	ON	OFF	OFF	OFF	

APPENDIX 13
SELECTED AT-MIO-64E-5 SPECIFICATIONS

High-Speed Multifunction I/O Boards for the IBM PC AT

AT-MIO-64F-5

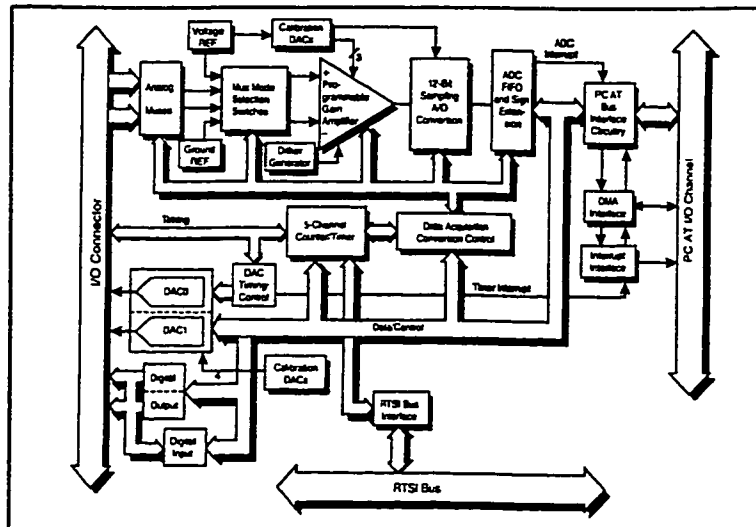


Overview

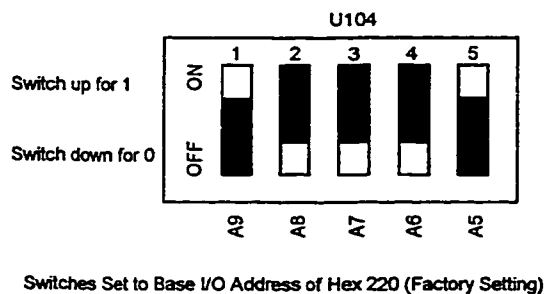
The AT-MIO-64F-5 is a high-performance, multifunction analog, digital, and timing input / output board for the PC AT compatible computers. It contains a 12 bit sampling ADC with 64 analog inputs, two 12 bit DACs with voltage outputs, eight lines of TTL-compatible digital I/O, and three 16-bit counter / timer channels for timing I/O.

Features

- Fast, 12-bit ADC
 - 200 kS/s sustained sampling rate
 - 64 single-ended or 32 differential channels
 - NI-PGIA amplifier guarantees settling at all gains
 - Software-selectable range : 0-10V, $\pm 5V$, $\pm 10V$
 - Channel-independent programmable gains : 1, 2, 5, 10, 20, 50, 100
 - 512 sample FIFO buffer
 - Internal or external ADC timing and triggering
- 2 independent double-buffered, multiplying 12-bit DACs
 - 2 Kword FIFO DAC buffer
- 8 digital I/O lines capable of sinking up to 24 mA each
- 3 independent 16-bit counter / timers
- Software-controlled self-calibration
- RTSI bus for system-level timing and multiboard synchronization
- 16-bit DMA with single and dual DMA channel modes
- Programmed with LabVIEW, LabWindows/CVI, and LabWindows



Base Address Selection



Switch Setting					Address Space
A9	A8	A7	A6	A5	Used (Hex)
0	0	X	X	X	Reserved
0	1	0	0	0	100 - 11F
0	1	0	0	1	120 - 13F
0	1	0	1	0	140 - 15F
0	1	0	1	1	160 - 17F
0	1	1	0	0	180 - 19F
0	1	1	0	1	1A0 - 1BF
0	1	1	1	0	1C0 - 1DF
0	1	1	1	1	1E0 - 1FF
1	0	0	0	0	200 - 21F
1	0	0	0	1	220 - 23F
1	0	0	1	0	240 - 25F
1	0	0	1	1	260 - 27F
1	0	1	0	0	280 - 29F
1	0	1	0	1	2A0 - 2BF
1	0	1	1	0	2C0 - 2DF
1	0	1	1	1	2E0 - 2FF
1	1	0	0	0	300 - 31F
1	1	0	0	1	320 - 33F
1	1	0	1	0	340 - 35F
1	1	0	1	1	360 - 37F
1	1	1	0	0	380 - 39F
1	1	1	0	1	3A0 - 3BF
1	1	1	1	0	3C0 - 3DF
1	1	1	1	1	3E0 - 3FF

APPENDIX 14
SELECTED PCDIO 216 SPECIFICATIONS

PCDIO 216 Digital I/O Board

Overview

The PCDIO family consists of generalized multiple channel digital I/O boards providing buffered, user selectable inputs and outputs based on the 8255 PIO by Intel. The I/O is software selectable in groups of 8 (groups of 4 for Port C of each 8255) for either input or output.

The boards are pin compatible with popular industrial solid state I/O racks and modules such as the PB8, PB 16A, PB24 and those manufactured by Opto 22, Grayhill, Gordos, and others. The PCDIO's provide a 50-pin male IDC connector for each 8255, which will interface to 8, 16 and 24 position OPTO racks. Because the I/O are programmable, both input and output modules can be mixed on one rack in 8 (or 4) channel groups. Five volts to power a single 8-channel rack is available on the board and is fused. If a large relay rack is required, an external +5VDC power source is required. Most computers have sufficient power available for multiple racks, however, the power must be taken directly from the power supply, not through the PCDIO the board.

The boards use LS245 buffers which will provide 15mA of source current and 24mA of sink current (48mA with specially requested buffers.)

On the PCDIO24-P and PCDIO216-P cards, two of the VO lines of each 8255 may be used to generate a hardware interrupt. The inputs are buffered and can be connected to IRQ 2, thru 7 on the PCDIO24-P, and the same plus 10 thru 12, 14, and 15 on the PCDIO216-P. Only one hardware interrupt connection per board is allowed although multiple 8255s may share an interrupt line.



Industrial Computer Source
9950 Barnes Canyon Road
San Diego, CA 92121-2720
(610)-677-0877

Features

- 24 to 216 Channels of Digital Input/Output
- Four and Eight Bit Groups Independently Selectable for I/O
- All I/O Lines Buffered on Board
- Hysteresis on I/O Lines
- All lines pulled up to +5Volts
- OPTO-22 Compatible 50 Pin Connector
- Positive True Logic
- Interrupt and Interrupt Disable Capability (PCDIO24-P, and PCDIO216-P)
- Fused on Board +5V Supply Available for User (PCDIO24-P only)

Specifications

- Size
 - 13.25" L
- Environment
 - Operating Temp 0 to 60 °C
 - Storage Temp -50 to 120 °C
- Humidity
 - 0 to 90 % non-condensing
- Agency Approvals
 - EC Compliance 89/336/EEC,EU

Inputs and Outputs

- PCDIO126
 - 216 Input / Output lines
- 15 mA Output Source Current (output high)
- 24 mA Output sink Current (output low)
- Address
 - 36 Bytes mapped within 200-3FF range of I/O address space
- Power Output
 - +5V, 1 A fuse
- I/O connector
 - Asingle 50 Pin header for each 8255 (24 channels)

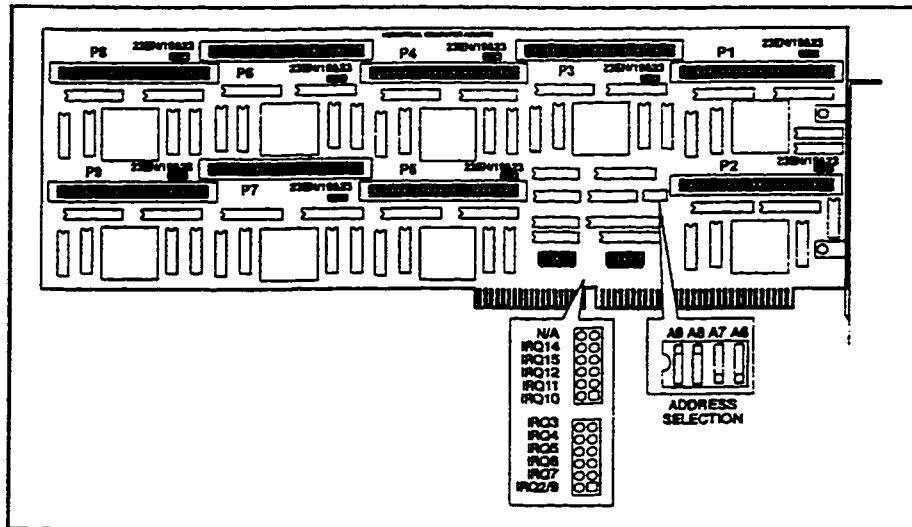


Figure 9 PCDIO 216 board layout

Base Address

	A9	A8	A7	A6
200H	OFF	ON	ON	ON
300H	OFF	OFF	ON	ON
350H	OFF	OFF	ON	OFF

Connector Pin Assignments

A 50 pin ribbon cable header is used for interfacing to I/O. Connector pin assignments are as follows:

<u>Assignemnt</u>	<u>Channel</u>	<u>I/O</u>	<u>Pin</u>
Port C HIGH	PC7	23	1
Port C HIGH	PC6	22	3
Port C HIGH	PC5	21	5
Port C HIGH	PC4	20	7
Port C LOW	PC3	19	9
Port C LOW	PC2	18	11
Port C LOW	PC1	17	13
Port C LOW	PC0	16	15
Port B	PB7	15	17
Port B	PB6	14	19
Port B	PB5	13	21
Port B	PB4	12	23
Port B	PB3	11	25
Port B	PB2	10	27
Port B	PB1	9	29
Port B	PB0	8	31
Port A	PA7	7	33
Port A	PA6	6	35
Port A	PA5	5	37
Port A	PA4	4	39
Port A	PA3	3	41
Port A	PA2	2	43
Port A	PA1	1	45
Port A	PA0	0	
47+5VDC	(1A Fuse)		49

All even numbered pins are connected to a common ground.

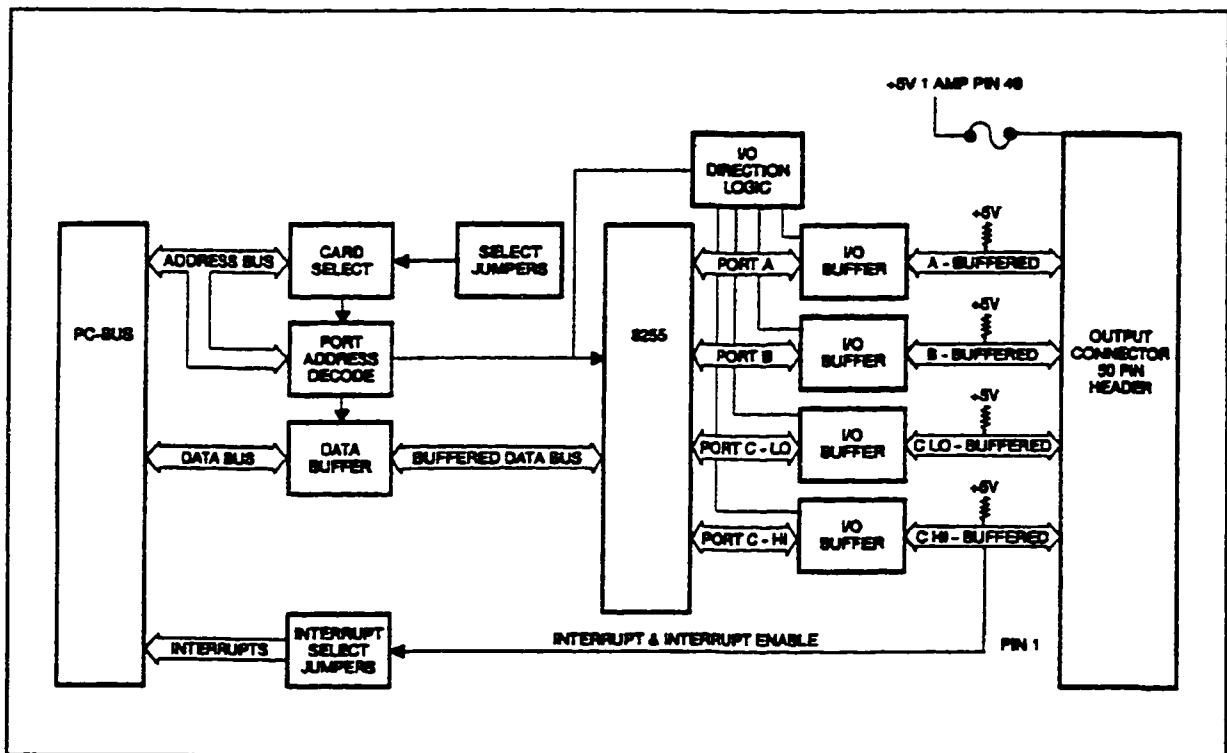


Figure 10 Block diagram

Programming

The PCDIOs use the 8255 PIO to provide the 24 bits of Input and Output. These 24 bits are divided into three 8-bit ports: A, B and C. Port C is further divided into two 4-bit ports:

C-Lower and C-Upper.

The outputs of the PCDIOs are positive true, i.e. a 1 written to a bit will cause it go high. This high output will turn OFF a solid state relay attached to that channel. To turn a relay ON, write a 0 to the appropriate bit.

The 8255 provides a CONTROL REGISTER at the card's BASE ADDRESS+3. This is a WRITE ONLY, 8-bit register and is used to set the MODE and DIRECTION of these 4 ports (A, B, C-Upper, C-Lower). On Power-Up or Reset, all 24 VO lines are set as inputs. The 8255 should be configured by writing to the CONTROL REGISTER before the chip's ports are accessed, even as inputs.

CAUTION:

Setting a port as an output port by writing to the control register initializes the outputs as all 0's or LOW. If attached to a relay rack such as the PB24, relays attached to these ports will be turned ON until 1's are written to the appropriate output bits to turn them off.

Register Access

Each 8255 section is mapped into 4 bytes in the IBM yO space. The address definitions are:

Base Address +	0	Port A	Read/Write
1	Port B		Read/Write
2	Port B		Read/Write
3	Control Register		Write Only

For the additional ports on the PCDIO48-P, PCDIO72-P, PCDIO120-P, and PCDIO216-P add 4 to the Base Address for each additional 8255. For example, the third port set on the PCDIO120-P would be channels 49 through 72, the 24 bits of the third 8255. The ports of that chip will be Base Address + 8 through Base Address + 11 for Port A through the Control Register respectively.

To access the board from BASIC, for example:

1. Determine which ports are to be for input and ~ to write to the CONTROL REGISTER as discussed in the following section.

~ example, for a configuration of:

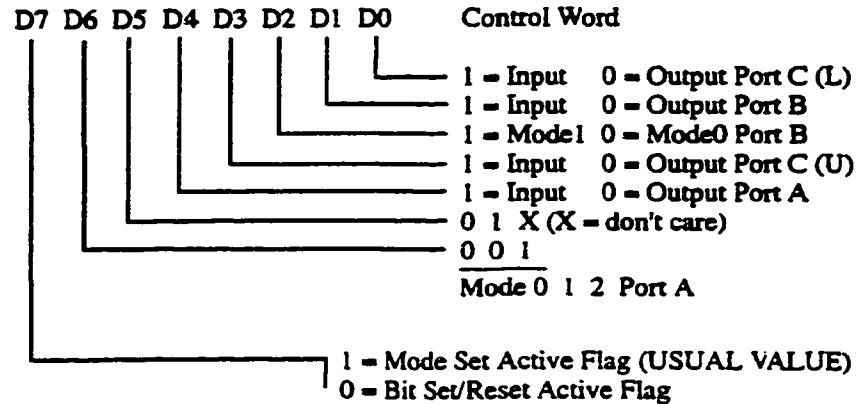
Port A	Input
Port B	Output
Port C-Upper	Input
Port C-Lower	Output

The bit pattern would be 1001 1000 (Hex 98). Use the BASIC OUT statement to write to the CONTROL REGISTER (referto the next section for more detail on the CONTROL REGISTER):

```
10 BASEADDR ~ &H300
20 OUT BASEADDR+3, &H98
30 X=INP(BASEADDR): Y=INP(BASEADDR+2)      'READ PORTS A,C
40 PRINT X;Y                                'PRINT THE VALUES
50 OUT BASEADDR+1,255                        'TURN OFF PORT B RELAYS
60 OUT BASEADDR+Z,15                         'TURN ON 4 BITS OF C
```

Before the 8255 can be used, it must be configured. Each of the 8-bit ports A, B, and C can be selected as input or output ports (port C can also be selected as a control port if the buffers have been replaced with wire jumpers). Remember, the 8255 power-up default mode is with all ports configured for inputs and the CONTROL REGISTER should be configured even if this is the mode you require.

8255 Mode Definition Format



Modes of Operation

The 8255 offers 3 MODES of operation as set by bits 3, 6 and 7. Bit 3 sets the mode for port B and bits 6 or 7 for port A. Port C has no independent modes. The 8255 has several input/output modes and the user is directed to the 8255 data sheet for complete programming details. The following information will provide sufficient information for the majority of users.

Modes 1 and 2 require bi-directional I/O which cannot be supported with the buffered I/O of the PCDIOs. This information is given only as a point of reference to help the user understand the 8255 datasheet. If Modes 1 or 2 are required for an application, Industrial Computer Source offers an un-buffered, 24 channel card called the DI024-P. All three modes are supported on the DI024P as all I/O connects directly to the 8255. These modes can also be implemented on the PCDIO's by removing the 74LS243 I/O buffers from the board. Replace the buffers with wired headers to carry the 8255 signals through the sockets to the edge connector. See the schematics located at the end of this manual for wiring details. The modes are described below (Mode 1 and 2 descriptions are provided for information only):

MODE 0 Basic Input/Output - This is the standard PCDIO mode

MODE 1 Strobed Input/Output

MODE 2 Bi-directional Bus

Note: If you choose to remove the buffers from the board, your warranty will not be voided as long as proper care is taken in removing the ICs. If the board is physically damaged in the process, the warranty will be voided.

PCDIO Series Manual

MODE 0

This functional configuration provides simple input and output operations for each of the three ports. No "handshaking" is required, data is simply written to or read from a specified port.

Mode 0 Basic Functional Definitions:

- Two 8 bit ports and two 4-bit ports
- Any port can be input or output
- Outputs are latched
- Inputs are not latched
- 16 different Input/Output combinations are possible in this mode

MODE 1

This functional configuration provides a means for transferring y0 data to or from a specified port in conjunction with strobes or "handshaking" signals. In Mode 1, Port A and Port B use the lines on Port C to generate or accept these "handshaking" signals.

Mode 1 Basic Functional Definitions

- Two Groups (Group A and Group B)
- Each group contains one 8-bit data port and one 4-bit control/data port
- The 8-bit data port can be either input or output. Both inputs and outputs are latched
- The 4-bit port is used for control and status of the 8-bit data port

MODE 2

This functional configuration provides a means for communicating with a peripheral device or structure on a single 8-bit bus for both transmitting and receiving data (bi-directional bus I/O). "Handshaking" signals are provided to maintain proper bus flow discipline in a similar manner to MODE 1. Interrupt generation and enable/disable functions are also available.

Mode 2 Basic Functional Definitions

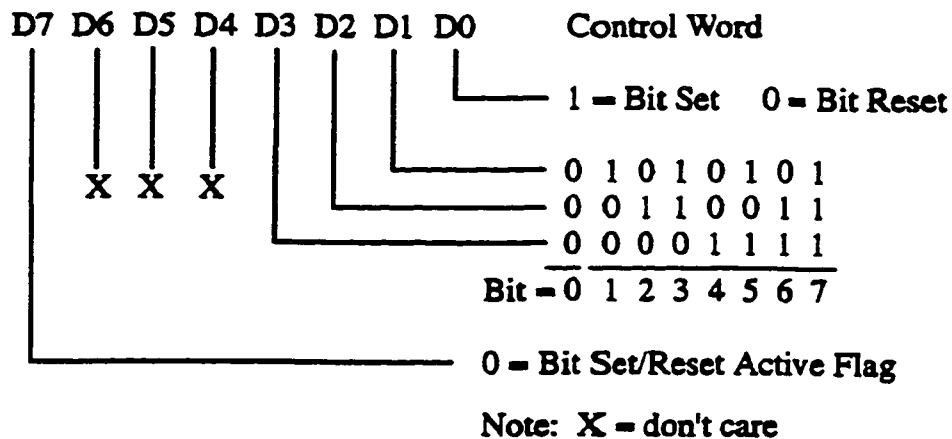
- Used in Group A only
- One 8-bit, bidirectional bus port (Port A) and a 5-bit control port (Port C)
- Both inputs and outputs are latched
- The 5-bit control port (Port C) is used for control and status for the 8-bit, bi-directional bus port (Port A)

Any of the eight bits of Port C can be Set or Reset using a single OUTput instruction. This feature reduces software requirements in control-based applications. When Port C is being used as status/ control for Port A or B, these bits can be set or reset by using the Bit Set/Reset operation just as if they were data output ports.

Note: This mode required disabling the buffers by replacing the buffers with wire jumpers.

Bit 7 controls the Bit Set/Reset function. When bit 7 is 1 the port is a control port. When set to 0 the Bit Set/Reset function is used.

Bit Set/Reset Format

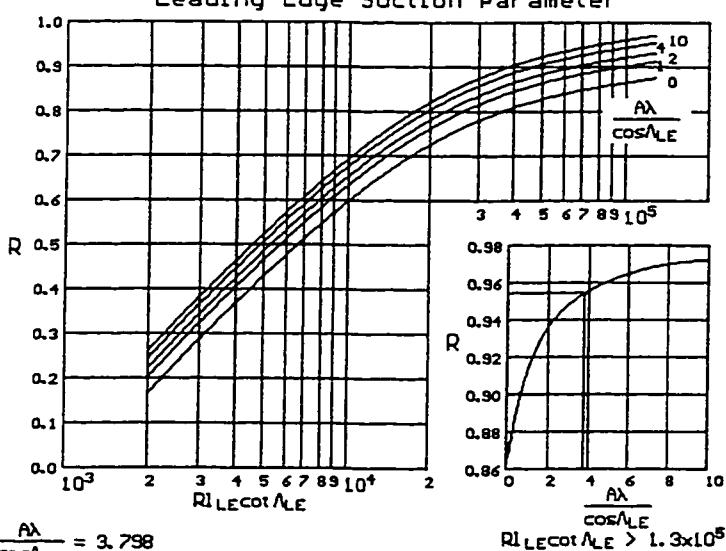


APPENDIX 15
STEP BY STEP SAMPLE COEFFICIENT SOLUTION

AIRCRAFT STABILITY AND CONTROL ANALYSIS V2.00	WING AERO
<p>The induced drag due to lift at aircraft $\alpha=0$ C_{DLU} (part 1)</p> <p>Given : air density $\rho = 1.225 \text{ kg/m}^3$ air absolute viscosity $\mu = 1.789\text{e-}05 \text{ kg/m s}$ aircraft velocity $u = 100.0 \text{ m/s}$ wing leading edge radius $l_{LEu} = 1.00 \text{ } \sqrt{C_u}$ wing mean aerodynamic chord $C_u = 1.56 \text{ m}$ wing leading edge sweep angle $\Lambda_{LEu} = 28.65^\circ$ wing aspect ratio $A_u = 6.67$ wing taper ratio $\lambda_u = 0.50$ wing lift curve slope $C_{L\alpha u} = 4.239 \text{ /rad}$ wing lift coef at aircraft $\alpha=0$ $C_L = 0.501$</p> <p>The Reynolds based on the leading edge radius</p> $R_{LEu} = \frac{\rho u l_{LEu} C_u}{\mu}$ $R_{LEu} = \frac{(1.225 \text{ kg/m}^3)(100.0 \text{ m/s})(0.01 \times 1.56 \text{ m})}{1.789\text{e-}05 \text{ kg/m s}}$ $R_{LEu} = 1.065\text{e+}05$	<p>ZERO LIFT αu LIFT SLOPE LIFT AT $\alpha u=0$ LIFT AT $\alpha_{ac}=0$ DRAG AT 0 LIFT 1 DRAG AT 0 LIFT 2 DRAG AT 0 LIFT 3 INDUCED DRAG 1 INDUCED DRAG 2 INDUCED DRAG 3 DRAG DUE TO α PITCH AT 0 LIFT GRAPH R_u GRAPH R_{uf} GRAPH R_{LS} GRAPH C_f</p>
	PREVIOUS

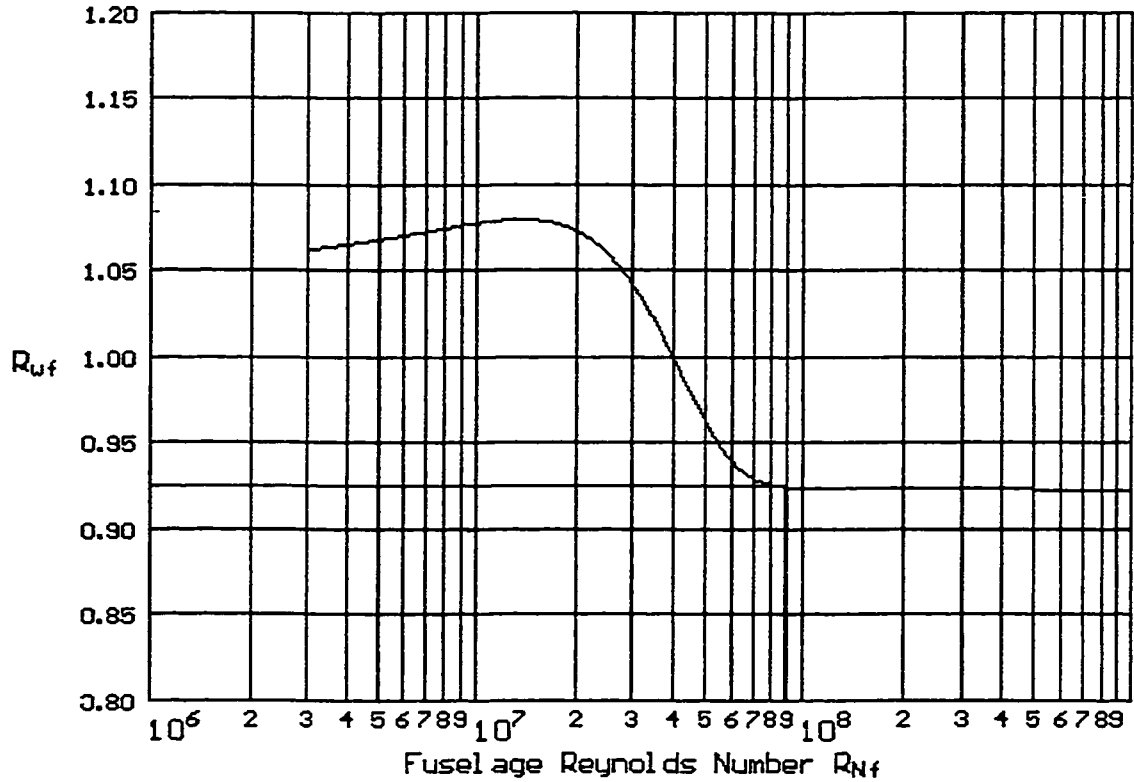
AIRCRAFT STABILITY AND CONTROL ANALYSIS V2.00	WING AERO
<p>The induced drag due to lift at aircraft $\alpha=0$ C_{DLU} (part 2)</p> <p>The first parameter used to determine the leading edge suction parameter</p> $P_1 = R_{LEu} \cot \Lambda_{LEu}$ $P_1 = (1.065\text{e+}05)(\cot 28.65^\circ)$ $P_1 = 1.949\text{e+}05$ <p>The second parameter used to determine the leading edge suction parameter</p> $P_2 = \frac{A_u \lambda_u}{\cos \Lambda_{LEu}}$ $P_2 = \frac{(6.67)(0.50)}{\cos 28.65^\circ}$ $P_2 = 3.80$ <p>Refer to GRAPH R_u to determine the leading edge suction parameter R_u</p> $R_u = 0.95$	<p>ZERO LIFT αu LIFT SLOPE LIFT AT $\alpha u=0$ LIFT AT $\alpha_{ac}=0$ DRAG AT 0 LIFT 1 DRAG AT 0 LIFT 2 DRAG AT 0 LIFT 3 INDUCED DRAG 1 INDUCED DRAG 2 INDUCED DRAG 3 DRAG DUE TO α PITCH AT 0 LIFT GRAPH R_u GRAPH R_{uf} GRAPH R_{LS} GRAPH C_f</p>
	PREVIOUS

AIRCRAFT STABILITY AND CONTROL ANALYSIS V2.00	
<p>The induced drag due to lift at aircraft $\alpha=0$ C_{DLu} (part 3)</p> <p>The wing span efficiency factor is</p> $e_u = \frac{1.1 C_{L_{\alpha u}} / A_u}{R_u (C_{L_{\alpha u}} / A_u) + (1 - R_u) \pi}$ $e_u = \frac{1.1 (4.239 / 6.67)}{0.95 (4.239 / 6.67) + (1 - 0.95) \pi}$ $e_u = 0.93$ <p>Finally the induced drag due to lift at aircraft $\alpha=0$ C_{DLu}</p> $C_{DLu} = \frac{C_{L_u}^2}{\pi R_u e_u}$ $C_{DLu} = \frac{0.501^2}{\pi (6.67) (0.93)}$ $C_{DLu} = 1.289e-02$	<p>WING AERO</p> <p>ZERO LIFT α_u</p> <p>LIFT SLOPE</p> <p>LIFT AT $\alpha_u=0$</p> <p>LIFT AT $\alpha_{ac}=0$</p> <p>DRAG AT 0 LIFT 1</p> <p>DRAG AT 0 LIFT 2</p> <p>DRAG AT 0 LIFT 3</p> <p>INDUCED DRAG 1</p> <p>INDUCED DRAG 2</p> <p>INDUCED DRAG 3</p> <p>DRAG DUE TO α</p> <p>PITCH AT 0 LIFT</p> <p>GRAPH R_u</p> <p>GRAPH R_{uf}</p> <p>GRAPH R_{LS}</p> <p>GRAPH C_f</p> <p>PREVIOUS</p>

AIRCRAFT STABILITY AND CONTROL ANALYSIS V2.00	
<p>Leading Edge Suction Parameter</p>  <p>$\frac{\Lambda}{\cos \Lambda / L E} = 3.798$</p> <p>$R_{1 L E \cot \Lambda / L E} = 1.949e+05$</p> <p>$R = 0.954$</p>	<p>WING AERO</p> <p>ZERO LIFT α_u</p> <p>LIFT SLOPE</p> <p>LIFT AT $\alpha_u=0$</p> <p>LIFT AT $\alpha_{ac}=0$</p> <p>DRAG AT 0 LIFT 1</p> <p>DRAG AT 0 LIFT 2</p> <p>DRAG AT 0 LIFT 3</p> <p>INDUCED DRAG 1</p> <p>INDUCED DRAG 2</p> <p>INDUCED DRAG 3</p> <p>DRAG DUE TO α</p> <p>PITCH AT 0 LIFT</p> <p>GRAPH R_u</p> <p>GRAPH R_{uf}</p> <p>GRAPH R_{LS}</p> <p>GRAPH C_f</p> <p>PREVIOUS</p>

APPENDIX 16
NUMERICAL SOLUTION OF DATCOM GRAPHICAL DATA

Wing Fuselage Interference Factor



$$R_{Nf} = 8.900e+07$$

$$R_{wf} = 0.924$$

```

/*****
/*****
/**
/**      Wing Fuselage Interference Factor      **
/**      Ref : Roskam, Airplane Design Part VI, Fig 4.1      **
/**      *****/
double GR_Rwf(double x)
/*-----
    x = Fuselage Reynolds Number
    y = Wing Fuselage Interference Factor

    n = (x-c)/d
    y = a+b*exp(-0.5n^2)

a= 0.9237054300705015

```

```

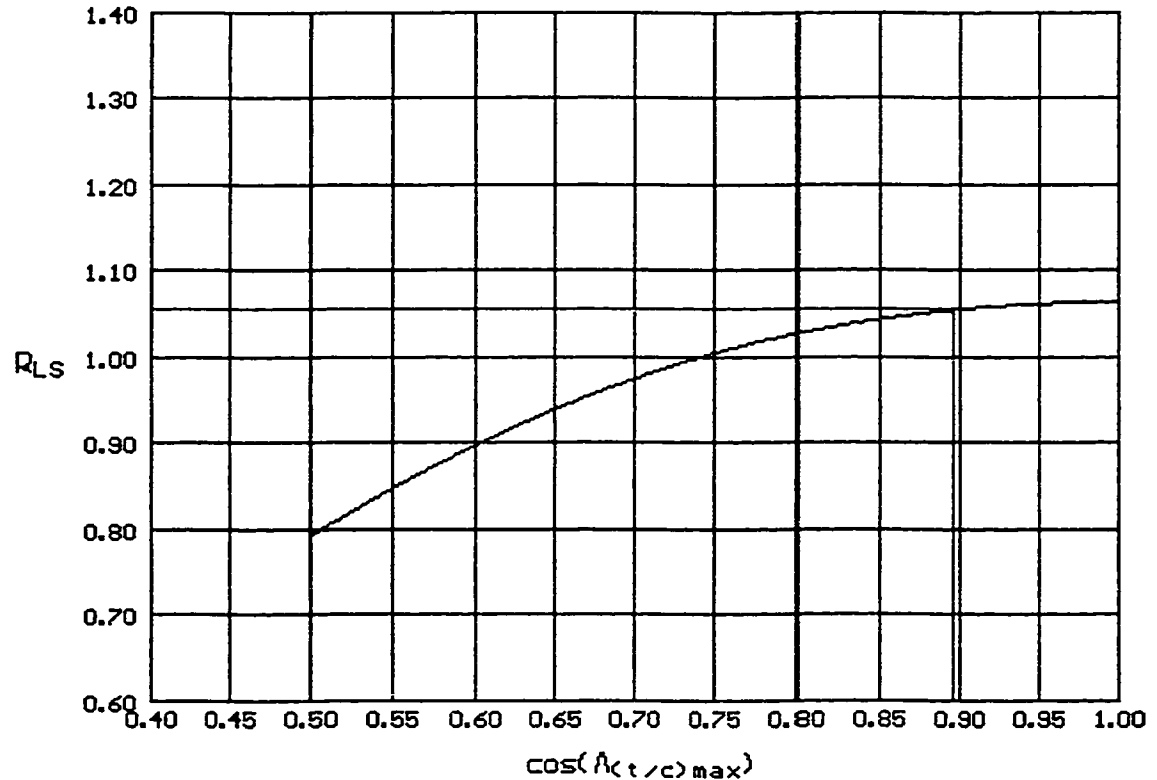
    b= 0.1554171549944683
    c= 13437286.29431236
    d= 21527910.20842851
*-----*/
{
    double y;
    double n;

    n=(x-13437286.29431236)/21527910.20842851;
    y=0.9237054300705015+0.1554171549944683*exp(-0.5*n*n);
    if (x>=5.0e8) y=0.92238005;

    return(y);
}

```

Swept Wing Lift Correction Factor



$$\cos(\Lambda_{t/c})_{\max} = 0.896$$

$$R_{LS} = 1.054$$

```

/*****/
/*****/
/**
/**      Swept Wing Lift Correction Factor      **/
/**
/**      Ref : Roskam, Airplane Design Part VI, Fig 4.2      **/
/**
/*****/
/*****/
/*-----*/
double GR_Rls(double x)
/*-----*/
    x= cosine of sweep angle at maximum wing thickness
    y= Swept Wing Lift Correction Factor

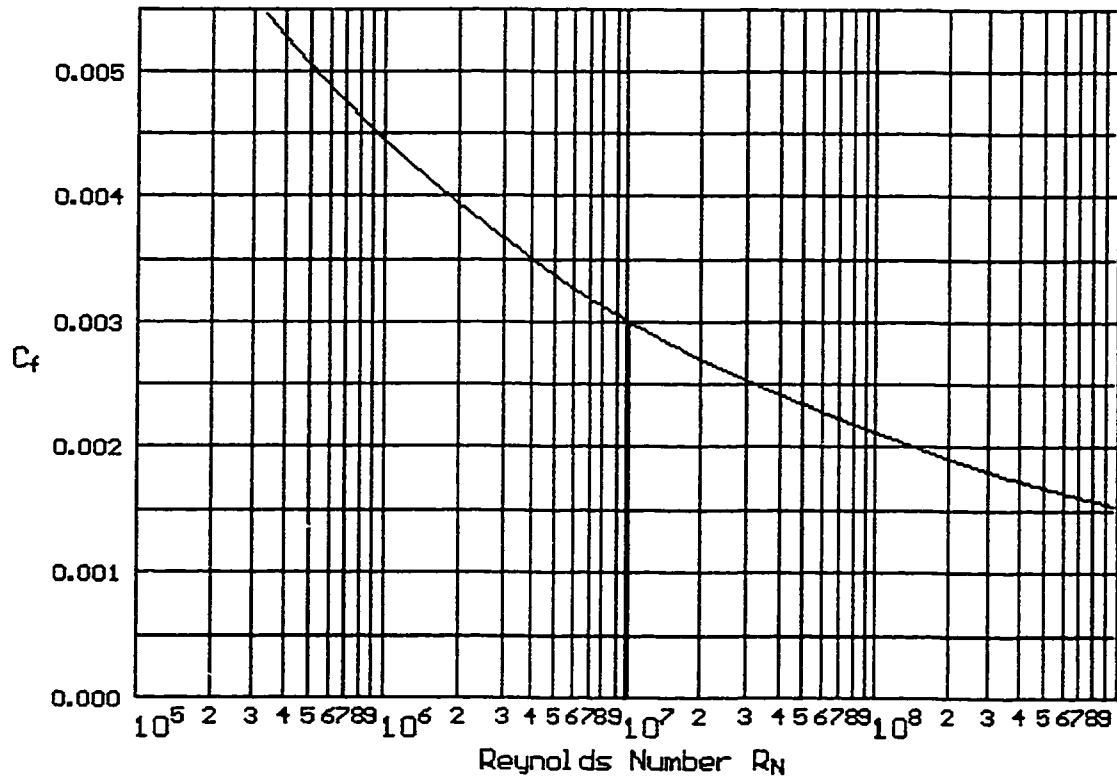
    y = (a + b*(log(x))^2)^-1

    a= 0.9409364970775835
    b= 0.666566816982024
/*-----*/

```

```
{  
  double y;  
  double x1;  
  
  x1=log(x)*log(x);  
  y=0.9409364970775835+0.6665668169820240*x1;  
  return(1/y);  
}
```

Skin Friction Coefficient



$R_N = 1.025e+07$

$C_f = 0.003$

```

/*****
/*****
/**                               **/
/**      Skin Friction Coefficient      **/
/**                               **/
/**      Ref : Roskam, Airplane Design Part VI, Fig 4.3      **/
/**                               **/
/*****
/*****
/*-----*/
double GR_Cf(double x)
/*-----*
    x= Reynolds Number
    y= Skin Friction Coefficient

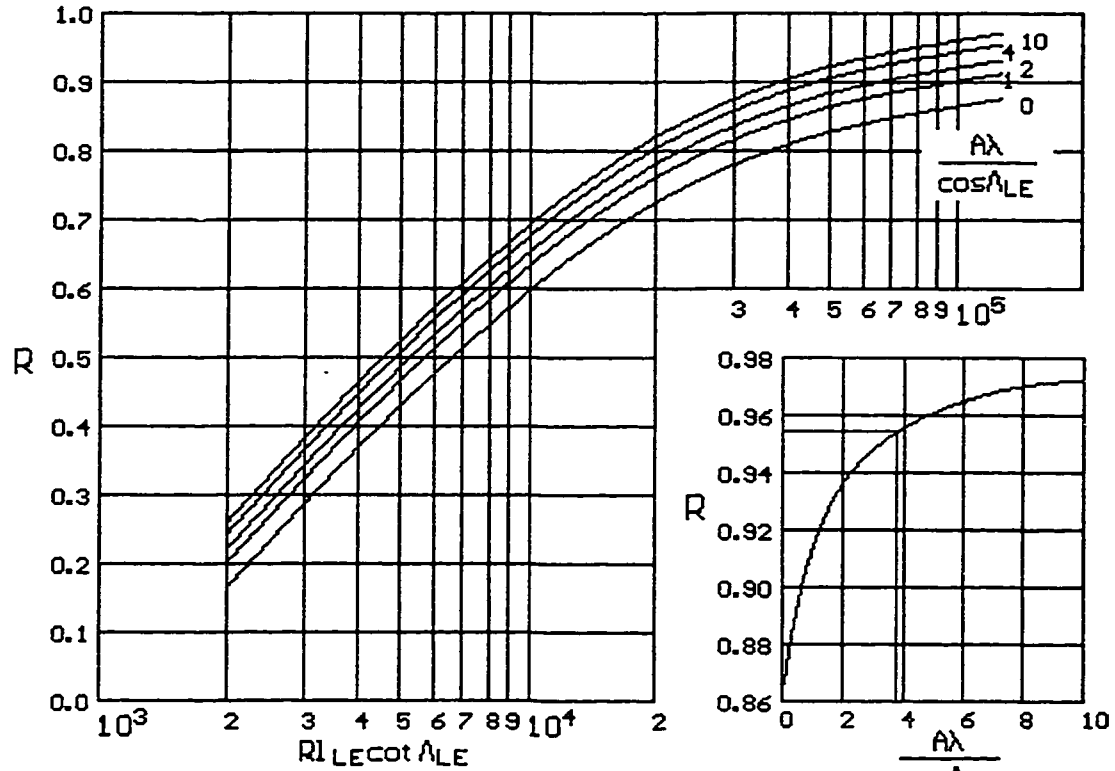
    y = (a + b/log(x))^2

    a= -0.01601752613511262
    b= 1.142517531148012
*-----*/

```

```
{  
    double y;  
    double x1;  
  
    x1=1.0/log(x);  
    y=-0.01601752613511262+1.142517531148012*x1;  
    return(y*y);  
}
```


Leading Edge Suction Parameter



$$\frac{A\lambda}{\cos \Lambda_LE} = 3.798$$

$$R1_LECot \Lambda_LE = 1.949e+05$$

$$R = 0.954$$

$$R1_LECot \Lambda_LE > 1.3 \times 10^5$$

```

/*****
/*
/*      Leading Edge Suction Parameter
/*      (Reynolds(Leading edge radius) x cos(Leading sweep)
/*      <= 1.3e5
/*
/*      Ref : Roskam, Airplane Design Part VI, Fig 4.7
/*
/*
/*****
/*-----*/
double GR_R_LER_A(double x, double y)
/*-----*/
    x = (Reynolds(Leading edge radius) x cos(Leading sweep)
    y = (Aspect ratio)(taper ratio)/cos(Leading sweep)
    z = Leading Edge Suction Parameter

    z = a+b(lnx)^2+c*lnx+d/lnx+e/x^(0.5)+f*lnx/x+gy+hy^(0.5) lny+iy^(0.5)

    a= 407.2257114331721
    b= 0.7009214773562579

```

```

c= -28.72407811435704
d= -1997.399793583934
e= 1572.198729250785
f= -419.902200634983
g= -0.02217983956494627
h= 0.01825011273664422
i= 0.05806201770954975
*-----*/
{
double z;
double f1,f2,f3,f4,f5,f6,f7,f8;

f1=log(x)*log(x);
f2=log(x);
f3=1.0/log(x);
f4=1.0/sqrt(x);
f5=log(x)/x;
f6=y;
f7=sqrt(y)*log(y);
f8=sqrt(y);
z=407.2257114331721+0.7009214773562579*f1
-28.72407811435704*f2-1997.399793583934*f3
+1572.198729250785*f4-419.9022006349830*f5
-0.02217983956494627*f6+0.01825011273664422*f7
+0.05806201770954975*f8;

return z;
}
/*****
/*
/*      Leading Edge Suction Parameter
/*      (Reynolds(Leading edge radius) x cos(Leading sweep)
/*      > 1.3e5
/*
/*
/*      Ref : Roskam, Airplane Design Part VI, Fig 4.7
/*
/*
/*****
/*-----*/
double GR_R_LER_B(double x)
/*-----*
x = (Aspect ratio) (taper ratio)/cos(Leading sweep)
y = Leading Edge Suction Parameter

x1 = x;
x2 = x*sqrt(x);
x3 = exp(x);
x4 = exp(-x);

y = a+b*x1+c*x2+d*d3+e&x4

a= 0.9277020635020378
b= 0.01186689887562091
c= -0.002328078375015301
d= -7.126511884153572E-08
e= -0.06770273541114157
*-----*/
{
double y;

```

```

double x1,x2,x3,x4;

x1=x;
x2=x*sqrt(x);
x3=exp(x);
x4=exp(-x);
y=0.9277020635020378+0.01186689887562091*x1
  -0.002328078375015301*x2-7.126511884153572E-08*x3
  -0.06770273541114157*x4;

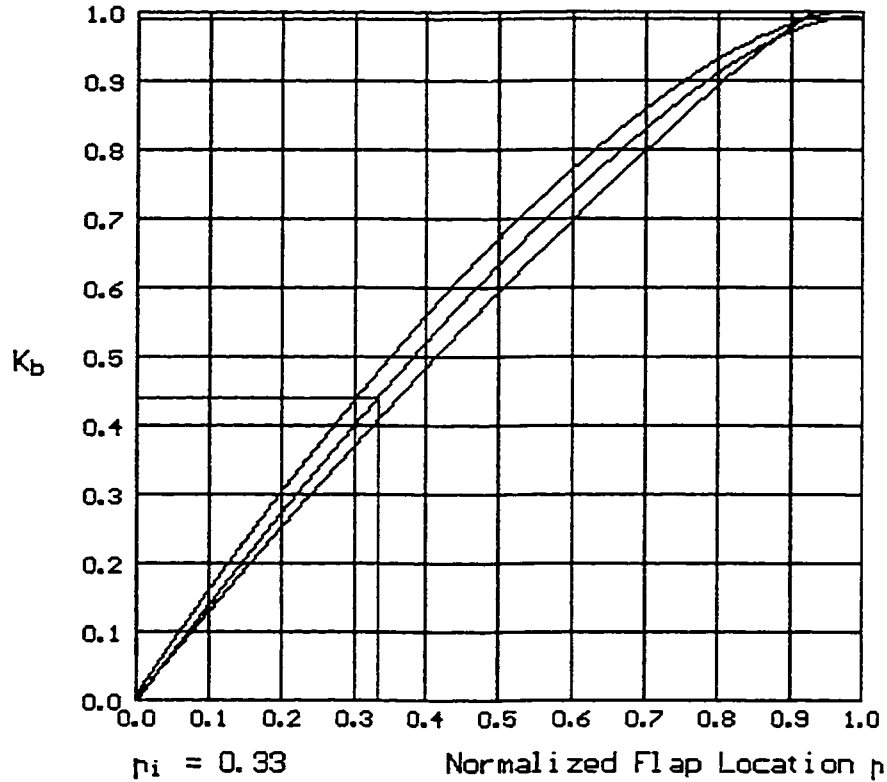
return(y);
}
/*-----*/
/*****
/*****
/*****
/**
/**      Leading Edge Suction Parameter      **
/**
/**      Ref : Roskam, Airplane Design Part VI, Fig 4.7      **
/**
/*****
/*****
/*-----*/
double GR_R_LER(double x, double y)
/*-----*
    x = (Reynolds(Leading edge radius) x cos(Leading sweep)
    y = (Aspect ratio)(taper ratio)/cos(Leading sweep)
    z = Leading Edge Suction Parameter
/*-----*/
{
    double z;

    if(x>1.3e5)
        z = GR_R_LER_B(y);
    else
        z = GR_R_LER_A(x,y);

    return z;
}

```

Taper Ratio and Flap Span Effect



$$K_b = K_{bo} - K_{bi} = 0.99 - 0.44 = 0.550$$

```

/*****
/*****
/**
/**      Taper Ratio and Flap Span Effect
/**
/**      Ref : Roskam, Airplane Design Part VI, Fig 8.52
/**
/*****
/*****
/*****
/*****
double GR_Kb(double x, double y)
/*****
    x = Flap Location
    y = Taper Ratio
    z = Taper Ratio and Flap Span Effect

    z = (a+cx+ey+gx^2+iy^2+kxy)/(1+bx+dy+fx^2+hy^2+jxy)

a= 0.005923755181552471
b= -0.3994372200805626

```

```

c= 1.581396468779569
d= -0.09572156240408393
e= -0.03302182924458352
f= -0.1734246620068174
g= -1.15876451847447
h= 0.01513970643137449
i= 0.02763852817635231
j= -0.3091625031858922
k= -0.3858408180781654
*-----*/
{
double z;

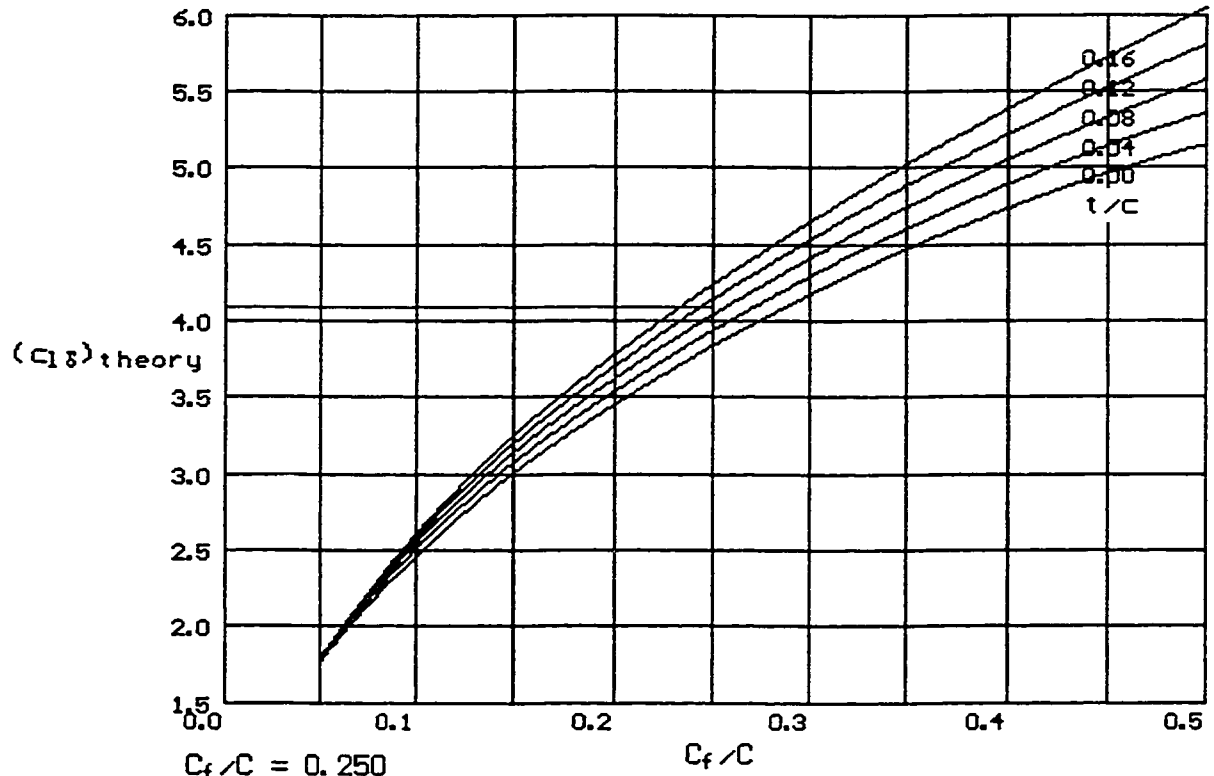
z=(0.005923755181552471+x*(1.581396468779569+x*(-1.158764518474470))+
y*(-0.03302182924458352+y*(0.02763852817635231))+
x*y*(-0.3858408180781654))/
(1+x*(-0.3994372200805626+x*(-0.1734246620068174))+
y*(-0.09572156240408393+y*(0.01513970643137449))+
x*y*(-0.3091625031858922)));

if(z<0.0) z = 0.0;
if(z>1.0) z = 1.0;

return z;
}

```

Lift Effectiveness of a Plain Flap



```

/*****
/*****
/**
/**      Plain Flap Lift Effectiveness
/**
/**      Ref : Roskam, Airplane Design Part VI, Fig 8.14
/**
/*****
/*****
/*-----*/
double GR_Cld_th(double x, double y)
/*-----*/
    x = Flap Chord Ratio
    y = Thickness Ratio
    z = Plain Flap Lift Effectiveness

    z = (a+bx+cx^2+dy+ey^2+fy^3)/(1+gx+hx^2+ix^3+jy)

    a= 0.8175957257666048
    b= 25.13880606159232

```

```

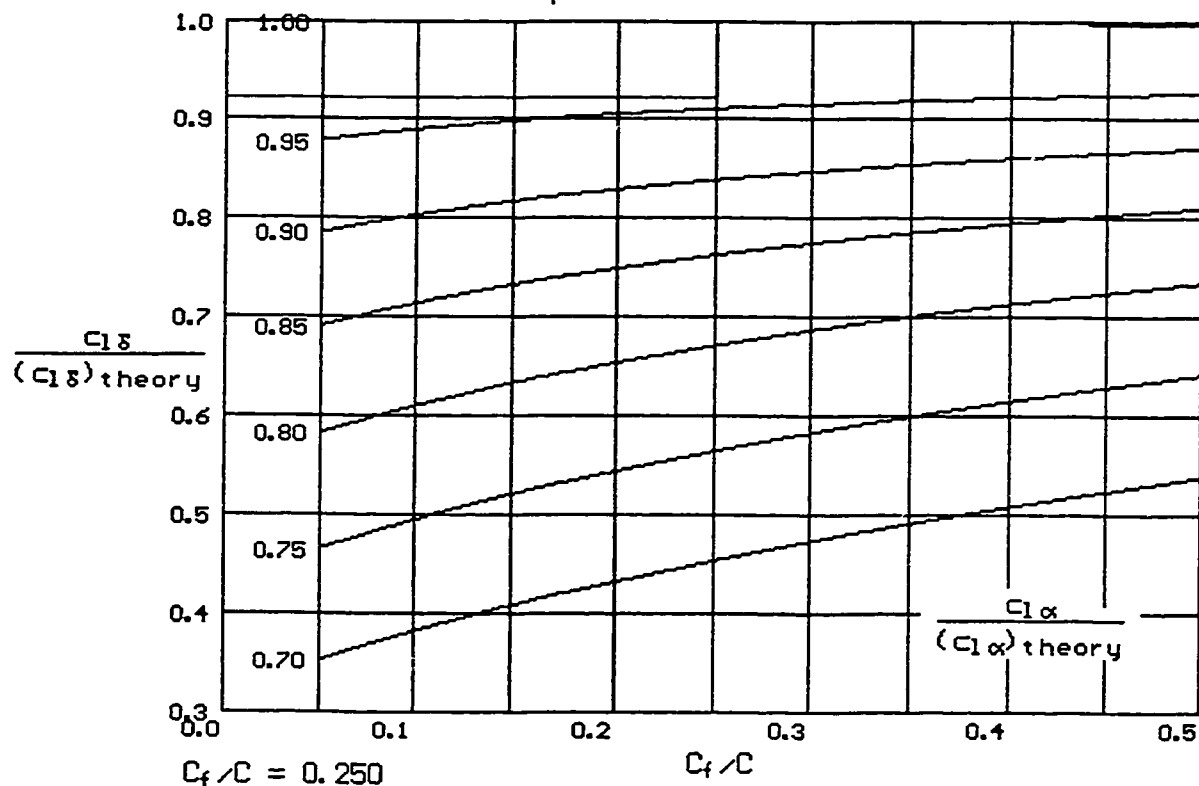
c= -32.06238550543281
d= -1.904824467363515
e= -2.915845099724644
f= -5.479167936707572
g= 2.932255619531504
h= -7.221576426701003
i= 3.059776332815722
j= -1.379032141446571
*-----*/
{
double z;
double z1,z2,z3,z4;

z1=0.8175957257666048+x*(25.13880606159232+
x*-32.06238550543281);
z2=y*(-1.904824467363515+
y*(-2.915845099724644+y*-5.479167936707572));
z3=1.000000000000000+x*(2.932255619531504+
x*(-7.221576426701003+x*3.059776332815722));
z4=y*-1.379032141446571;
z=(z1+z2)/(z3+z4);

return z;
}

```

Plain Flap Correction Factor



```

/*****
/*****
/**
/**      Plain Flap Lift Corection Factor
/**
/**      Ref : Roskam, Airplane Design Part VI, Fig 8.15
/**
/**
/*****
/*****
/*-----*/
double GR_CldCld_th(double x, double y)
/*-----*/
    x = Flap Chord Ratio
    y = (actulal airfoil lift slope)/(theoretical airfoil lift slope)
    z = Plain Flap Lift Corection Factor

    z = (a+bx+cx^2+dx^3+ey)/(1+fx+gy+hy^2+iy^3)

    a= -0.002302155067196882
    b= 0.04426181923149298

```



```

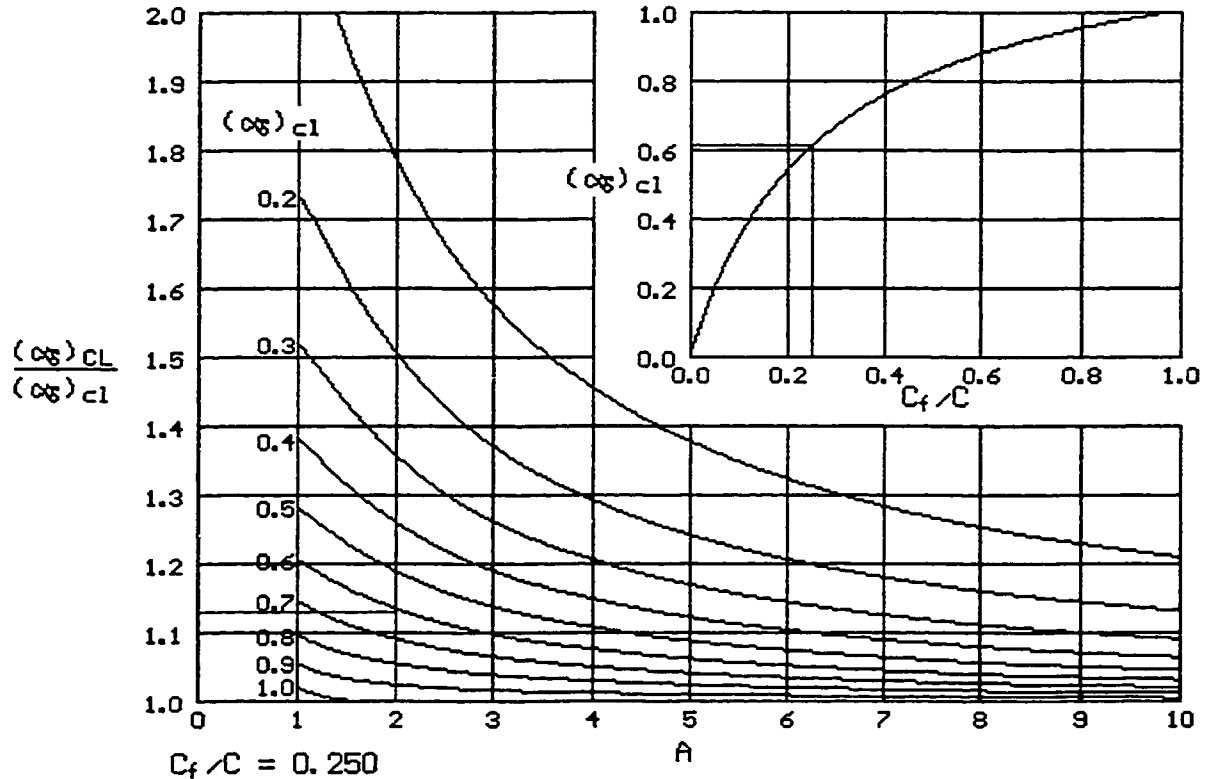
c= -0.006174257785062725
d= 0.004219675648187597
e= 0.02356844626850709
f= 0.04244054495267236
g= -3.101690684697937
h= 3.315896942108051
i= -1.192885218026223
*-----*/
{
double z;
double z1,z2,z3,z4;

if(y>1.0) y = 1.0;

z1=-0.002302155067196882+x*(0.04426181923149298+
    x*(-0.006174257785062725+x*0.004219675648187597));
z2=y*0.02356844626850709;
z3=1.000000000000000+x*0.04244054495267236;
z4=y*(-3.101690684697937+
    y*(3.315896942108051+y*-1.192885218026223));
z=(z1+z2)/(z3+z4);
if(z>1.0) z =1.0;
return z;
}

```

Flap & Chord Ratio Effect on 3D Flap Effectiveness



$C_f/C = 0.250$

$A = 2.00$

$(\alpha_{eff})_{C1} = 0.611$

$\frac{(\alpha_{eff})_{CL}}{(\alpha_{eff})_{C1}} = 1.13$

```

/*****
/*****
/**
/**      Chord Ration Effect on 3D Flap Effectivemness (part A) **/
/**
/**      Ref : Roskam, Airplane Design Part VI, Fig 8.53      **/
/**
/*****
/*****
/*-----*/
double GR_adCL_adclA(double x)
/*-----*
    x = Flap Chord Ratio
    y = Effective Angle of Attack

    y = (a+cx)/(1+bx)

    a= 0.004325812202535138
    b= 3.638247859522454
    c= 4.650557358561155
    
```

```

*-----*/
{
    double y;

    y=(0.004325812202535138+x*4.650557358561155)/
        (1.0+x*(3.638247859522454));

    if(y>1.0) y = 1.0;
    if(y<0.0) y = 0.0;

    return(y);
}

/*****
/*****
/**
/**      Chord Ration Effect on 3D Flap Effectivemness (part B) **/
/**
/**      Ref : Roskam, Airplane Design Part VI, Fig 8.53      **/
/**
/*****
/*****
/*-----*/
double GR_adCL_adclB(double x, double y)
/*-----*/
    x = Aspect Ratio
    y = Effective Angle of Attack
    z = Chord Ration Effect on 3D Flap Effectivemness

    z =
a+b/x+c*lny+d/x^2+e(lny)^2+f(lny)/x+g/x^3+h(lny)^3+i(lny)^2/x+j(lny)/x^2

    a= 0.9989225862060573
    b= -0.01469964435393975
    c= 0.01417225362651576
    d= 0.002160791029572189
    e= 0.01702589839847724
    f= -0.6988712914748292
    g= 0.03413100241945523
    h= 0.002645213332458325
    i= 0.06245656144039343
    j= 0.3631691899572078
*-----*/
{
    double z;

    x=1.0/x;
    y=log(y);
    z=0.9989225862060573+

x*(-0.01469964435393975+x*(0.002160791029572189+x*(0.03413100241945523))
)+

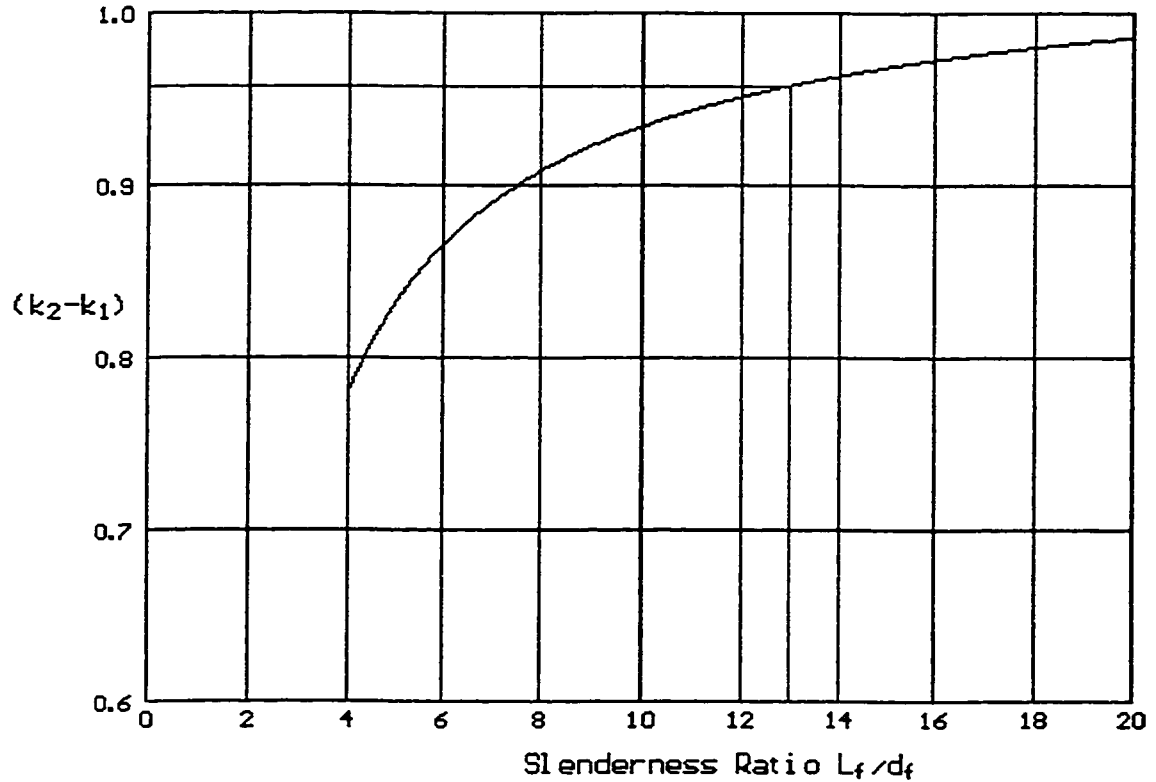
y*(0.01417225362651576+y*(0.01702589839847724+y*(0.002645213332458325)))
+

x*y*(-0.6988712914748292+y*(0.06245656144039343)+x*(0.3631691899572078))

```

```
;
    if(z<1.0) z = 1.0;
    return z;
}
```

Effect of FusSlenderness on the App Mass Factor



$$L_f/d_f = 13.000$$

$$(k_2 - k_1) = 0.957$$

```

/*****
/*****
/**
/**      Effect of Fuselage Slenderness on Apparent Mass Factor **/
/**
/**      Ref : Roskam, Airplane Design Part VI, Fig 8.111      **/
/**
/*****
/*****
/*-----*/
double GR_K2_K1(double x)
/*-----*
    x = Fuselage Slenderness Ratio
    y = K2-K1

    y = a+b/x

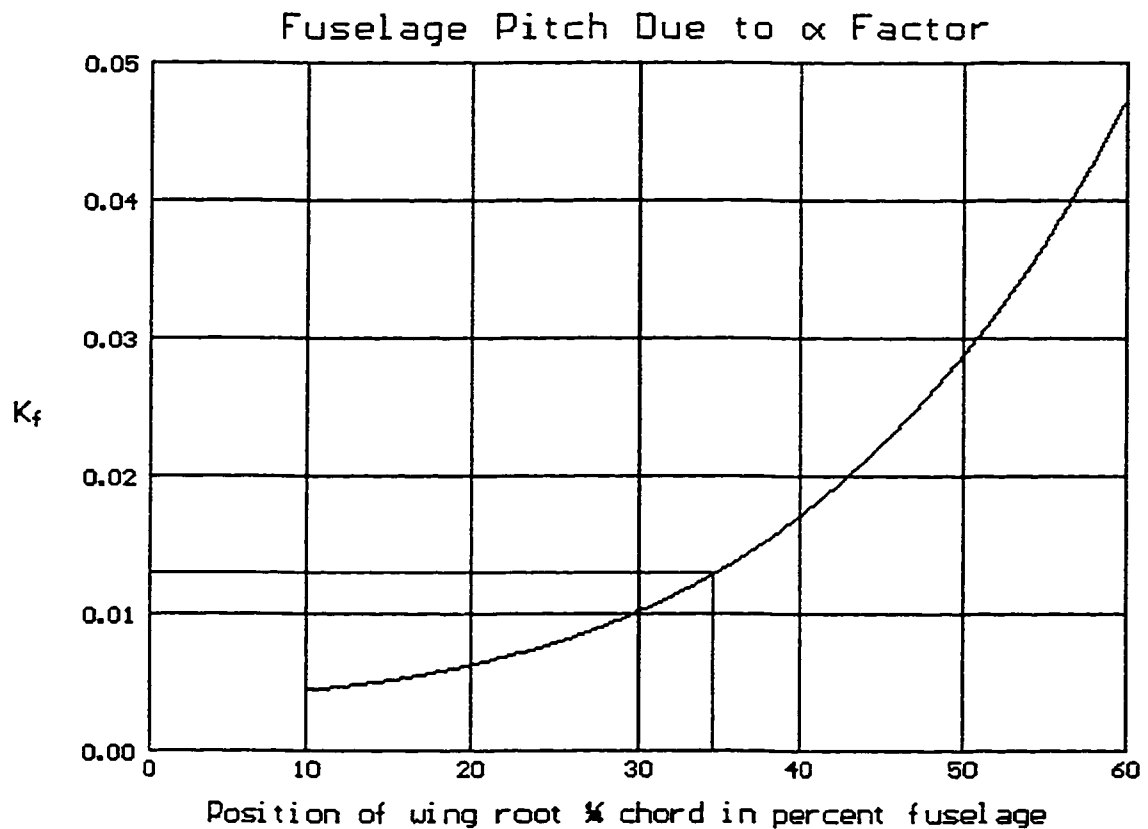
    a= 1.036443632680053
    b= -1.029420224251813
*-----*/

```

```
{
  double y;
  double x1;

  x1=1.0/x;
  y=1.036443632680053-1.029420224251813*x1;
  if(y>1.0) y=1.0;

  return(y);
}
```



$$K_f = 1.285e-02$$

```

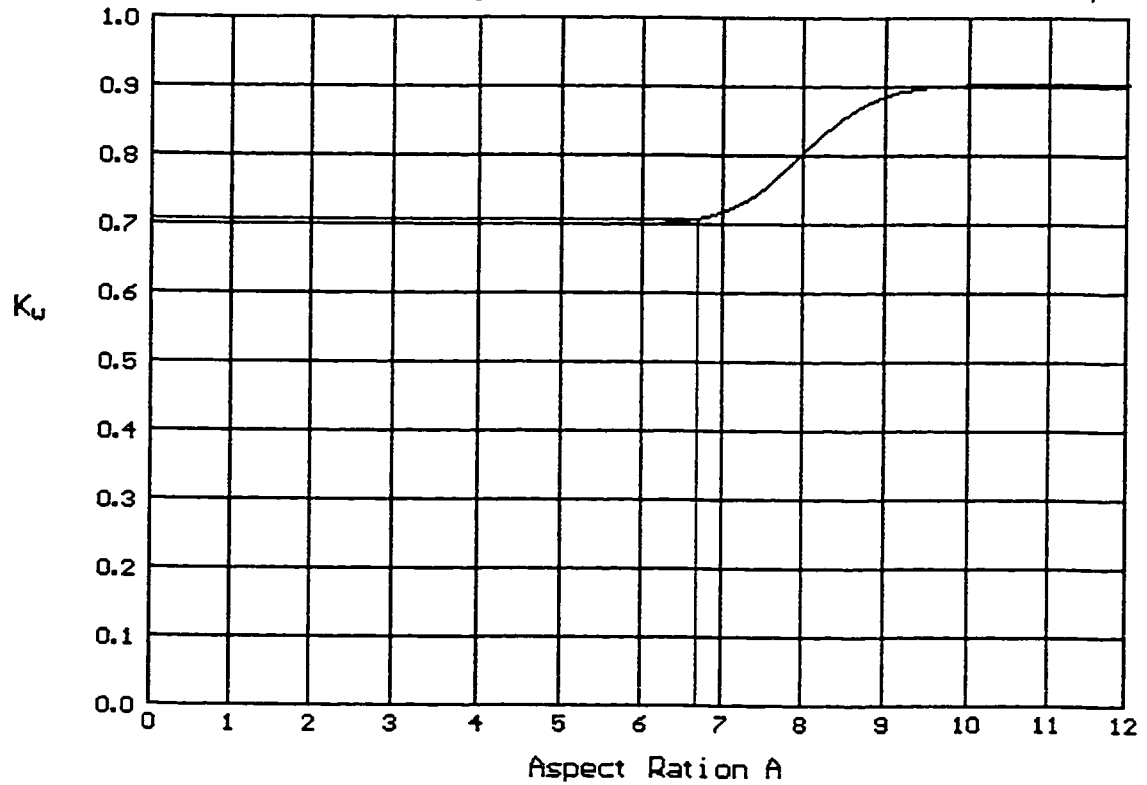
/*****
/*****
/**
/**      Fuselage Pitch Contribution Due to Angle of Attack      **
/**      PERKINS & HAGE, Airplane Performance, Stability and    **
/**      Control, fig 5-16                                       **
/**                                                              **
/*****
/*****
/*-----*/
double GR_Kf(double x)
/*-----*/
    x = Position of wing root quarter chord % position of fuselage
    y = Fuselage Pitch Contribution Due to Angle of Attack
/*-----*/
{

    double a = 0.061504401;

```

```
double b = 4.3267859e-5;  
double z;  
  
z = a+b*x*x;  
z = z*z;  
  
return z;  
}
```


Correction to Wing Contribution to Pitch Damping



$$A = 6.67$$

$$K_u = 0.71$$

```

/*****
/*****
/**
/**      Correction to Wing Contribution to Pitch Damping      **
/**      Ref : Roskam, Airplane Design Part VI, Fig 10.40      **
/**      ****
/*****
/*****
/*-----*/
double GR_Kw(double x)
/*-----*
    x = Aspect Ratio
    y = Correction to Wing Contribution to Pitch Damping

    y=(a+cx+ex^2+gx^3+ix^4+kx^5)/
      (1+bx+dx^2+fx^3+hx^4+jx^5)

    a= 0.7006441008676853
    b= -0.0003783497422434312

```

```

c= -0.002737357559235027
d= -0.005256659310493493
e= -0.00240401470653993
f= -0.005746944482679346
g= -0.004066234135706156
h= 0.0006378350835078342
i= 0.0003911797733949342
j= -4.254736192369186E-06
k= 3.304914785271525E-06
*-----*/
{
    double y;

    if(x<6.0)  x = 6.0;
    if(x>10.0) x = 10.0;

    y=(0.7006441008676853+x*(-0.002737357559235027+
        x*(-0.002404014706539930+x*(-0.004066234135706156+
        x*(0.0003911797733949342+x*3.304914785271525E-06))))/
        (1.0+x*(-0.0003783497422434312+x*(-0.005256659310493493+
        x*(-0.005746944482679346+x*(0.0006378350835078342+
        x*-4.254736192369186E-06)))));

    return(y);
}

```